Making Algorithmic Stablecoins More Stable: The Terra-Luna Case Study

Federico Calandra[1,*], Francesco P. Rossi[1], Francesco Fabris[2] and Marco Bernardo[1]

[1]Dipartimento di Scienze Pure e Applicate, Università degli Studi di Urbino Carlo Bo, Urbino, Italy
[2]Dipartimento di Matematica, Informatica e Geoscienze, Università degli Studi di Trieste, Trieste, Italy

Abstract
We study the dynamics of the Terra-Luna collapse occurred in May 2022 by creating a simulation environment that embodies both the free market buying-selling transactions and the Terra-Luna protocol exchange features. The parameters used during the simulation generate the conditions necessary for triggering the deviation from the peg of the stablecoin UST, along with the subsequent collapse of its value to almost zero. Then we present three proposals to increase the stability of algorithmic stablecoins and we employ the simulation environment to show how they could help stabilize the algorithmic stablecoin's peg.

Keywords
Decentralized finance, algorithmic stablecoins, Terra-Luna ecosystem

1. Introduction

The *decentralized finance* (*DeFi*) environment introduces a new paradigm in the finance sector. It enables unbanked users with smartphones and Internet access to engage in financial activities like money transfer, lending, borrowing, or speculating on synthetic stocks or commodities, on a 24/7/365 basis, without the need for a bank or other financial institution's involvement [1]. DeFi platforms utilize cryptocurrencies as the primary medium of exchange within their ecosystems.

The volatility of these digital assets has forced the introduction of *collateralized stablecoins* (CS). These are cryptocurrencies backed by a fiat currency or a commodity – typically USD, but also EUR, CHF, JPY, RMB, KRW, or gold. These digital assets maintain dynamically the peg with the corresponding fiat currency through a seigniorage process, along with a certification that for each collateralized token there exists a corresponding amount of value of the fiat currency, deposited in a bank, that can be redeemed.

Nowadays, we count dozens of fiat-backed stablecoins traded on the main cryptocurrency exchanges, such as USDT, USDC, TUSD, and BUSD. Their capitalization and success are increasing over time. They are used as a safe haven of stability when traders want to exit the volatility turmoil of the market and control the keys of the owned tokens, avoiding the use of a centralized exchange, the only able to swap a cryptocurrency with a fiat currency.

Proceedings of the 6th Distributed Ledger Technology Workshop (DLT 2024), May 14 - 15, 2024, Torino, Italy
*Corresponding author.
federico.calandra@unicam.it (F. Calandra); f.rossi51@campus.uniurb.it (F. P. Rossi); ffabris@units.it (F. Fabris); marco.bernardo@uniurb.it (M. Bernardo)

The spirit of the DeFi philosophy is that of creating a new paradigm for a decentralized financial system, totally detached from (central) banks, financial institutions, and fiat currencies. Unfortunately, collateralized stablecoins do not comply with this philosophy, since by using them the DeFi sector remains bonded with the traditional financial market via the fiat currency used as collateral. It is at this point that *algorithmic stablecoins* (AS) come into play. They are coins or tokens whose price is anchored to fiat currency solely by using an algorithmic protocol. More precisely, there are two different kinds of pure algorithmic stablecoin pegging mechanisms: *rebasing* and *seigniorage*.

In the rebasing model, stablecoin's total supply is not fixed and is modified adaptively on a regular basis, directly on the wallets of all users. The general idea is that when the price of the AS is above the parity with the adopted fiat currency – say the USD – it is necessary to increase the supply, since the demand is high and the token is too scarce. On the contrary, when the price is below the parity, it is necessary to decrease the supply, since the demand is low and the token is too abundant. This implies that a user with, say, 1000 tokens today, could find a greater amount in her/his wallet the day after, say 1100, if AS price $> \$1$, without any action by the user. On the contrary, if AS price $< \$1$, then the wallet content could be smaller, say 900 tokens. The *Ampleforth* protocol (AMPL) [2] is an example of this model.

The seigniorage model typically has two tokens: the AS token and the *governance token* (GT). The AS token is the algorithmic stablecoin, while the GT token is used to absorb the volatility of the AS token. This means that when AS price $< \$1$, we can profitably burn 1 AS for $1 worth of GT inside the protocol, which we can sell at market, yielding the difference as a profit. This reduces the total supply of AS and stabilizes the price. On the contrary, when AS price $> \$1$, burning $1 worth of GT can mint 1 AS, which we can sell at market, yielding the difference as a profit. This increases the total supply of AS and stabilizes the price.

The *Terra-Luna* protocol [3], based on the AS TerraUSD (UST) and the GT LUNA, is of this kind. It has been, at the same time, the most successful and the worst example of how to build a decentralized AS. The most successful because it globally collected almost $60B of capitalization and $20B of Total Value Locked[1] (TVL) in no more than 15 months between January 2021 and May 2022, in an unprecedented rash of money induced by the *Anchor protocol* platform, which ensured 20% of Annual Percentage Yield (APY) for users who were lending UST. The worst example because more than 90% of the entire market cap was lost in 7 days between May 9 and May 15, 2022, as a consequence of a disastrous collapse induced by an irreversible depeg of UST, which crashed its value to almost zero.

After recalling the basics of Terra-Luna (Section 2), we present the first contribution of this paper, which is a simulation environment for Terra-Luna based on Matlab® (Section 3). This environment mirrors the dynamics of the free market, allowing users to buy and sell UST and LUNA. Moreover, by changing some parameters, we can induce the market conditions that led to the collapse of the UST peg.

The second contribution of this work is to illustrate three proposals that aim to improve the stability of the original protocol (Section 4). Firstly we propose a redesign of the original algorithmic stabilization mechanism used in the Terra protocol, whose failure caused the collapse of the system. The second proposal involves creating a USDT reserve pool that the protocol

[1]It is the amount of USD locked in smart contracts of the DeFi's protocols.

can utilize to automatically restore the AS price. Finally, we propose a solution that aims to prevent hyperinflation of GT by introducing an automatic BTC pool. The paper concludes with all simulation results (Section 5) and a final discussion (Section 6).

2. Background

2.1. The Terra Stabilization Mechanism

The *Terra algorithmic market module* (TMM) played a central role in maintaining the price stability of UST. This is the module that provides incentives for arbitrageurs[2] to mint or burn UST in response to price deviations from the peg [4].

When UST's market price falls below the peg, e.g., \$0.98, arbitrageurs can profitably burn 1 UST obtaining automatically \$1 worth of LUNA from the protocol, making a \$0.02 profit per UST burnt. Conversely, if UST's price exceeds the peg, e.g., \$1.02, they can burn \$1 worth of LUNA and mint 1 UST, again yielding a \$0.02 profit. This mechanism operates via the protocol's algorithmic market-maker, the *virtual liquidity pool* (VLP), with LUNA's price sourced from validator oracles, see Figure 1.

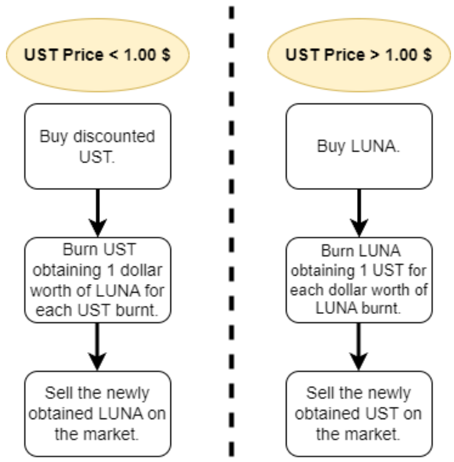


Figure 1: Strategy used by arbitrageurs in the Terra stabilization protocol

The VLP is implemented through a variant of the classical constant-product market-making algorithm [5]. In this case the constant-product formula CP is defined as:

$$CP = Pool_{Base}^2 \cdot \frac{1}{Price_{LUNA}} \tag{1}$$

where $Pool_{Base}$ is the initial quantity of UST in the pool, while the fraction $1/Price_{LUNA}$ expresses the price of LUNA in USD as observed in external markets [4]. $Price_{LUNA}$ is repeatedly updated by oracles, implying that the pool actively adapts to market fluctuations.

[2] An arbitrageur is an individual or entity that engages in the practice of exploiting price discrepancies in different markets to make profits.

The TMM integrates the $TerraPool_\delta$ stabilization mechanism, with the parameter δ indicating the deviation of the UST amount into the VLP compared to its base size $Pool_{Base}$:

$$Pool_{UST} = Pool_{Base} + \delta, \quad Pool_{LUNA} = \frac{CP}{Pool_{UST}} \tag{2}$$

The dynamics of δ play a crucial role in adjusting the liquidity pool sizes in response to market activities. As swaps happen and the balance between UST and LUNA quantities shifts, δ changes to ensure that CP stays constant. A key aspect of the market module's functionality is its ability to replenish the VLP, progressively bringing δ back towards zero. The rate of this replenishment is determined by the $PoolRecoveryPeriod$ parameter. At the end of each block – with one block being produced approximately every 6 seconds – δ is updated using the following formula:

$$\delta := \delta \cdot \left(1 - \frac{1}{PoolRecoveryPeriod}\right) \tag{3}$$

This formula governs the adjustment of δ, with $PoolRecoveryPeriod$ influencing the pace of the adjustment. This parameter was determined by the Terra community, and at the time of the de-pegging event, its value was 36, meaning that a partial replenishment of the VLP occurs every $36 \cdot 6 = 216$ seconds if no transactions take place during this period [4]. Note, as a consequence, that a full replenishment can be obtained only when the number of blocks tends to infinity.

2.2. The Terra-Luna Collapse

The collapse of the Terra protocol was triggered by a complex series of events [6]:

1. On May 5, 2022, there was evident selling pressure on UST and LUNA, indicated by negative hourly log returns. This selling pressure persisted, indicating a loss of confidence in both cryptocurrencies.
2. On May 7, 2022, the stablecoin UST lost its peg with USD due to a liquidity pool attack on *Curve-3pool*[3]. This event triggered intervention from the *Luna Foundation Guard* (LFG), which defended the UST peg and recovered the price temporarily.
3. Despite this intervention, on May 9, 2022, UST lost its peg for the second and final time, leading to a significant decrease in both LUNA and UST prices. This event marked a critical blow to the stability of the Terra-Luna ecosystem.
4. Finally, on May 11, 2022, an announcement by Do Kwon (co-founder and CEO of *Terraform Labs*), presumably indicating the last attempt to defend the peg by endorsing community proposal 1164, was interpreted by the market as a signal of the impending demise of the Terra-Luna ecosystem. This interpretation caused a final crash in the market, indicating the collapse of the Terra protocol.

[3]Curve-3pool is a DeFi liquidity pool protocol designed to facilitate efficient stablecoin trading. It is a component of the Curve Finance platform, which specializes in providing low-slippage swaps between similar assets, particularly stablecoins. Curve-3pool specifically focuses on stablecoins like USDT, USDC, and DAI, among others.

These events highlight a cascade of failures within the Terra-Luna protocol, including an irreversible depeg of the UST stablecoin, the lost of about $54B of capitalization, and the collapse of the other two DeFi platforms involved in the Terra-Luna ecosystem. They are the lending platform *Anchor* and the exchange *Mirror protocol*, which allowed users to create and trade *mirrored assets* (mAssets) that mirror the price of stocks and/or commodities of the real world.

2.3. Literature Review

Stablecoins have gained interest from researchers across disciplines and from financial institutions. The literature delves into their design, mechanisms, and impacts on financial stability, monetary policy, and regulation.

The ECB *Crypto-Assets Task Force* has tackled the issue of stablecoins in their report n. 247 [7]. Calcaterra et al. [8] study the first-order design principles for stablecoins by illustrating the core design features and their interoperative feedback, while the recent BIS Paper n. 141 [9] provides an overview of the evolution of the stablecoin market over the past decade and examines whether stablecoins have stayed true to their name in terms of being “stable”. Ante et al. [10] review 22 peer-reviewed articles, offering insights into stablecoin types, benefits, risks, and regulatory challenges. They identify research gaps, notably the lack of robust data and frameworks for analyzing the stablecoin ecosystem’s complexities. Furthermore, recent research by Choi and Kim [11] examines the challenges and opportunities associated with stablecoins and central bank digital currency (CBDC). They explore the financial stability implications of stablecoins and the potential for CBDCs to revolutionize payment infrastructures and monetary policies.

As for the algorithmic stablecoins, Clements [12] examine their fragility, highlighting risks like self-fulfilling runs and coordination failures. Recent failures, like the Terra-Luna collapse, bolster Clements’ argument, prompting questions about designing stablecoins resilient to high volatility.

The Terra-Luna ecosystem has been the subject of several studies that explore its unique features and challenges, as well as the failure that occurred in May 2022.

Cho [13] explains the collapse of the Terra project by analyzing the impact of the Anchor protocol on the system’s stability. Anchor was a great catalyzer for the demand and supply of UST, but – at the same time – was one of the responsible for the collapse of the system. Cho shows that, during the de-pegging event, the UST supply increased rapidly from 2.3B to 3.4B, contrary to the expected contraction mechanism that should have reduced the UST supply when the UST price was below the peg. The article attributes this anomaly to the run on the Anchor protocol, which allowed users to borrow UST at a low interest rate and sell it on the market, creating a downward pressure on the UST price and an upward pressure on the UST supply. The article of Cho showed that the UST price on exchanges followed the redeemed value of UST that users could obtain by swapping UST for LUNA and selling it on the market. Cho finds that the redeemed value of UST was consistently lower than the UST price on exchanges, indicating that users were under-compensated when they redeemed UST for LUNA. Cho also finds that the UST price on exchanges followed the redeemed value of UST closely, suggesting that users were arbitraging the price difference by selling UST on the market and buying UST on the Terra blockchain. This mechanism played a significant role in the de-pegging event, as it created a disincentive for users to hold UST and a downward pressure on the UST price.

Briola et al. [6] quantitatively describe the main events that led to the Terra project's failure by reviewing, in a systematic way, news from heterogeneous social media sources and by discussing the fragility of the Terra project and its vicious dependence on the Anchor protocol. They also identify the crash's trigger events, analyzing hourly and transaction data for BTC, LUNA, and UST.

Liu et al. [14] use data from the Terra blockchain and trading data from exchanges to study the dynamics and interactions of the system. They showed that it was a complex phenomenon that happened across multiple chains and assets and that the run on Terra was not due to market manipulation but rather to growing concerns about the sustainability of the system.

Kurovskiy and Rostova [15] investigate the collapse of the Terra-Luna ecosystem by using transaction-level data from the Terra blockchain and cryptocurrency exchanges, illustrating the several flaws in the design of UST that impeded its price stabilization.

Uhlig [16] develops a novel theory to account for the Terra crash and uses it to shed light on the data. He introduces a new methodology to show how crashes unfold gradually, by introducing the method of quantitative interpretation.

Ferretti and Furini [17] investigate the collapse of UST through Twitter as a passive sensor, analyzing sentiment in tweets to explore correlations with market value, highlighting the challenge of foreseeing sudden catastrophic events solely through sentiment analysis.

3. The Simulation Environment

We present two distinct and independent simulations of the Terra stability mechanism. The first one investigates the role of the mechanism in the collapse of Terra-Luna and is rooted in Cho's study [13], which highlights the system's limited redemption capacity. The second model delves into the repercussions of a sudden and substantial increase in LUNA supply during a collapse, resulting in its complete devaluation over a short period.

The same simulation environment will be used in Section 4 to propose three enhancements to the Terra protocol. The first one consists of a redesign of the stability mechanism, while the other two have to do with limiting the LUNA supply growth.

3.1. Price Dynamics through an AMM

To achieve our goals effectively, we need an environment that can accurately reflect the price changes in the free markets of the exchanges, for both the stablecoin and the governance coin, and the dynamics of the Terra stabilizing protocol. This can be done by simulating an *Automated Market Maker* (AMM), which operates within discrete time intervals called *iterations* or *samples*. AMMs serve as foundational components in *decentralized exchanges* (DEX) [18], allowing the two tokens associated with the AMM itself to be swapped; this approach revolutionizes assets trading through automated decentralized processes. Unlike traditional order-book-based exchanges, AMMs rely on *liquidity pools* (LP), constituted by the reserve of the two tokens, that algorithmically pair and maintain the two assets. The LP operates through a *constant-product* formula mechanism [5], which ensures that the token balance within its pools remains stable: $k = x \cdot y$, where x and y are the token reserve balances of the two tokens, while k is a constant called the *invariant* of the pool.

In our simulations, two AMMs (or markets) are implemented: one that governs the buying and selling of the stablecoin, and the other that governs the buying and selling of the governance token. During each time interval, a swap occurs within each AMM, influencing simulated token prices. The impact of these swaps on prices depends on the volume of tokens involved: larger volumes have a greater effect on simulated prices, so inducing a slippage on it.

Let us explore in detail how AMMs are implemented within the simulations. Each AMM is described by a LP Π_{T_a,T_b} composed of two tokens T_a and T_b. At the discrete time instant n, the state of a LP is defined by:

- $Q_a(n)$ and $Q_b(n)$, which represent the supplies of T_a and T_b, respectively, at iteration n.
- $k = Q_a(n) \cdot Q_b(n)$, which is the invariant at time n.

In our simulations, we assume zero transaction fees and constant pool liquidity for simplicity, as these factors are deemed unimportant for our analysis goals. The liquidity pool state at a given time n can be written as:

$$\Pi_{T_a,T_b,k}(Q_a(n), Q_b(n))$$

We can define a swap as a function that operates on a liquidity pool state. The swap function takes a token T_i and a swapped quantity q_i as input and produces a token T_o and the related quantity q_o as output. The token T_i can either be T_a or T_b, and the token T_o is consequently T_b or T_a. If the swap is performed at time $n+1$, the difference in supply at the output is equal to:

$$Q_o(n) - Q_o(n+1) = \frac{k}{Q_i(n)} - \frac{k}{Q_i(n) + q_i} \tag{4}$$

where $Q_i(n)$ and $Q_o(n)$ are the pool quantities of tokens T_i and T_o respectively.

In an AMM, the token price is measured in terms of the other token in the liquidity pool. Therefore, it is crucial to establish a fixed reference for token pricing. Here, the reference is USD, serving as a stable benchmark for analyzing price fluctuations of the algorithmic stablecoin. As fiat currency cannot be used in DeFi, we need to make an assumption. Let's designate the other token as T_U, a fully collateralized stablecoin pegged to USD; for example, USDT or USDC. This allows token values to be expressed in USD, assuming T_U remains at a constant external value of $1.

The concept of algorithmic stablecoin involves a stabilization mechanism that is usually based on two tokens: the stablecoin T_s (e.g. UST) and the governance token T_v, which is volatile (e.g. LUNA). Both simulations instantiate the AMM model above by employing two liquidity pools to emulate the price of T_s and T_v. The first pool, denoted by Π^S, is designed to simulate the market operations on T_s and consists of T_s and T_U. The second pool, denoted by Π^V, replicates the market dynamics of T_v and consists of T_v and T_U. At each time instant n, a swap occurs within Π^S and Π^V altering the price of T_s and T_v, the state of the liquidity pool is given by $\Pi_{T_a,T_U,k}(Q_a(n), Q_U(n))$ where $a = s$ or $a = v$, and the price $P_a(n)$ of the token T_a is determined by the ratio of the quantities of the two tokens inside the pool at that specific moment:

$$P_a(n) = \frac{Q_U(n)}{Q_a(n)} \tag{5}$$

This is what Uniswap, the most used DEX with the highest TVL, refers to as the "mid price" [19]; it can be viewed as the price at which one could theoretically trade an infinitesimally small amount of one token for the other in the pool, without slippage of the price.

3.2. Wallet Distribution

The simulation environment imposes the crucial choice of setting the effective quantities of each token we have to swap to simulate an ordinary session of the free market or to use in the simulated Terra-Luna redemption protocol. One possible approach could be that of using information about the amount of tokens inside the wallet of each user. Since we have no data about the original distribution of UST and LUNA, we estimated such data from the distribution of BTC and ETH [20] among addresses within their respective blockchains, under the hypothesis that all the cryptocurrencies show a similar wallet balance distribution. We found that BTC and ETH balances can be approximated with an exponential distribution with parameter λ, whose probability density function is defined as follows:

$$f(B,\lambda) = \begin{cases} \lambda e^{-\lambda B} & \text{if } B > 0 \\ 0 & \text{if } B \leq 0 \end{cases}$$

where B represents a given wallet balance and parameter λ was obtained using MATLAB's `fitdist()` function, which returns the value of λ that best fits the provided input data – in our case, the balances of BTC wallets. The analysis reveals that the majority of the wealth is held by a few "whales": for example, at the time of writing over 90% of all Bitcoin are held in wallets with a balance of more than 1 BTC, and only 4 addresses hold more than 100 000 BTC [21].

3.3. Stochastic Swaps

The probability of buying or selling a token within a liquidity pool is set by using the stochastic process of a *random walk*, which fully reflects the intrinsic uncertainty of the market. A random walk represents the cumulative effect of a sequence of random steps or movements, taken at discrete time intervals, starting from an initial position. Mathematically, a simple one-dimensional random walk can be defined as follows. Let $\{X_i\}$ be a sequence of independent and identically distributed random variables, with a probability distribution function $\mathcal{P}(X_i) = \mathcal{P}(X_0)$, representing the successive steps taken at discrete time points $t = 0, 1, 2, \ldots$ on the left or the right. The position of the walker at time t is given by the sum of all the random steps up to that point:

$$S_t = X_0 + X_1 + X_2 + \ldots + X_t$$

In the stochastic swap model presented in this study, a random walk is employed to determine the probability of a token T to be swapped within the pool at a given iteration. Let's consider the example of liquidity pool $\Pi_{T_a,T_b,k}$. At iteration n, $T = T_a$ with probability $p(n)$, and $T = T_b$ with probability $1 - p(n)$. So, there are three possible scenarios:

$$\begin{aligned} p(n) = \tfrac{1}{2} &\implies \text{equilibrium condition} \\ p(n) < \tfrac{1}{2} &\implies T_a \text{ experiences buying pressure} \\ p(n) > \tfrac{1}{2} &\implies T_a \text{ experiences selling pressure} \end{aligned} \tag{6}$$

At the beginning of each iteration, $p(n)$ is varied from its current value by an amount Δ:

$$p(n+1) = p(n) + \Delta \tag{7}$$

where Δ is a random variable with a normal distribution $\Phi(\mu_\Delta, \sigma^2_\Delta)$ such that its mean value μ_Δ is equal to zero, while its variance σ^2_Δ is a simulation parameter initialized during the simulation setup that expresses the market volatility. At the beginning of the simulations, we suppose that the market is in a state of equilibrium, hence $p(0) = \frac{1}{2}$. The value of $p(n)$ is subsequently updated based on equation (7), allowing for both positive or negative steps, that increases or reduces the probability of sale.

3.4. Inducing the Collapse

We have now the problem of representing the conditions of a panic sell (FUD – Fear, Uncertainty, Doubt) or of an irrational buy (FOMO – Fear of Missing Out) that can afflict the free market. During the FUD phase, users who are in a state of panic tend to sell their cryptocurrencies with great intensity, triggering what is known as a "bank run". Widespread panic can be triggered by any external event. In the case of the Terra-Luna ecosystem, it was the withdrawal of large amounts of UST from the Anchor protocol and the partial loss of the peg.

We can model this by introducing the concept of *panic zone* for the stablecoin. When the market enters the panic zone, a mechanism is triggered to represent the irrational behavior of users, whose decisions are driven more by emotions than rationality. Otherwise, we are in the *healthy zone*, where the users act as usual, in a normal condition of the market. It is understood that the stablecoin tends to maintain the peg inside the healthy zone, while tends to lose it outside, when entering the panic zone.

Concerning the pool Π^S, we have implemented two different definitions of panic zones, used in the simulations to represent the behavior of the buying/selling probability, when hovering over the boundaries separating the healthy zone from the panic zone of the market. The two approaches respectively use the price of the stablecoin and its probability of sale to trigger the risk of a collapse when entering the panic zone.

3.4.1. The Variable Mean Approach Based on Price

Suppose that at iteration $n = 0$ the price is set to $P_s(0) = 1$. As the simulation progresses, various buy and sell transactions of T_s will occur, leading to fluctuations in its probability and price on the basis of equations (6) and (7). At iteration n, T_s is in the healthy zone if:

$$|1 - P_s(n)| < \rho \qquad \text{with } \rho < 1 \tag{8}$$

In our simulation we have set $\rho = 0.05$; this implies that the healthy zone of T_s covers the interval \$$0.95 <$ $P_s(n) <$ \$$1.05$. When the price of the algorithmic stablecoin exits this healthy zone, a price collapse mechanism for T_s is triggered. In this scenario, the goal is to create selling or buying pressure that acts proportionally to the deviation of the price from \$1. To implement this selling or buying pressure, we adjust the mean value μ_Δ, initially set to 0, on the basis of the following formula:

$$\mu_\Delta = (1 - P_s(n)) \cdot \sigma_\Delta \tag{9}$$

So, when $|1 - P_s(n)| \geq \rho$ we are in the panic zone, and the following market conditions hold:

$$1-P_s(n) > 0 \implies \mu_\Delta > 0 \implies \text{selling pressure}$$
$$1-P_s(n) < 0 \implies \mu_\Delta < 0 \implies \text{buying pressure}$$

Note that the slippage of the mean is proportional to the slippage of the price.

3.4.2. The Variable Probability Approach

In the second approach the sell probability $p(n)$ freely fluctuates as described in equations (6) and (7), when we are in a healthy zone. The only difference is that now the healthy zone itself is defined in terms of probabilities, more precisely as the interval $0.3 \leq p(n) \leq 0.7$. Outside this interval, we are in the panic zone, and the user behavior is modeled through a completely different function representing her/his irrational mindset, which is an exponential-type function.

Define now the "$\mathcal{B}$ zone", or *buy zone*, as the market condition in which $p(n) < 0.3$: in this scenario, the demand for buying cryptocurrency is particularly high for various reasons. On the contrary, the "$\mathcal{S}$ zone", or *sell zone*, represents the most critical phase of the market, in which $p(n) > 0.7$ and the stablecoin risks the collapse. The functions $\mathcal{B}(d)$ and $\mathcal{S}(d)$ compute the sell probability within the panic zone:

$$\mathcal{B}(d) = \frac{1}{1+\mathrm{e}^{\frac{d+a_{\mathcal{B}}}{\eta}}} \qquad \mathcal{S}(d) = \frac{1}{1+\mathrm{e}^{-\frac{d+a_{\mathcal{S}}}{\eta}}} \tag{10}$$

Here d is the distance from the healthy zone, η is a parameter determining the slope of the function, and $a_{\mathcal{B}}$ and $a_{\mathcal{S}}$ need to be computed based on the parameters $\texttt{lowerBound} = 0.3$ and $\texttt{upperBound} = 0.7$ fixing the limits of the healthy zone, so as to align the values of d inside and outside the interval. More precisely we have:

$$a_{\mathcal{B}} = d \quad \text{if} \quad \frac{1}{1+\mathrm{e}^{\frac{d}{\eta}}} = \texttt{lowerBound} \qquad\qquad a_{\mathcal{S}} = d \quad \text{if} \quad \frac{1}{1+\mathrm{e}^{-\frac{d}{\eta}}} = \texttt{upperBound}$$

In our case $a_{\mathcal{B}} = a_{\mathcal{S}}$ because $0.5 - \texttt{lowerBound} = \texttt{upperBound} - 0.5$.

Inside the Terra-Luna protocol, the simulation keeps track of the number of tokens in circulation for both cryptocurrencies and models the principle of scarcity in the following way. When a new quantity of a token is minted, a positive step is taken during the random walk of the market in which it is traded, increasing its probability of sale. Conversely, when a certain quantity of a token is burnt, a negative step is taken during the random walk of the market in which it is traded, increasing its probability of purchase. The length of the step taken is proportional to the quantity minted or burnt.

The Terra-Luna protocol is not used during each time unit, but its probability of use $P_{Terra}(\xi)$ increases with the deviation ξ from the peg value. This is because a larger variation in the price of UST offers greater profit opportunities for arbitrageurs, who tend to use the protocol more frequently. The function $P_{Terra}(\xi)$ is defined as follows:

$$P_{Terra}(\xi) = \mathrm{e}^{-\frac{1}{(\xi+a)^t}} \cdot 0.5 + 0.5 \tag{11}$$

with a and t suitable constants. In our case $a = 0.55$, $t = 5$, and $\xi = |1 - \text{UST}.\texttt{price}| \cdot 10$. Equation (11) assumes different interpretations depending on whether the price of UST is above or below the parity. In the former case, the function returns the probability of minting UST; in the latter case, the function gives the probability of burning UST.

The scheme shown in Figure 1, describing the arbitrageurs' opportunity when using the Terra-Luna protocol, is not used in practice when the market approaches zone $\mathcal{S}$. Normally, when the price of UST is less than 1, this would imply the purchase of discounted UST to make profit. However, if the price undergoes a significant drop triggering a bank run, this is no more true. In a panic phase users simply want to get rid of all the UST already in their possession, without buying new ones to make profit, also because in these cases the value of LUNA received in change is plummeting.

4. Three Terra-Luna Stabilization Proposals

We now propose three mechanisms to improve the stability of the Terra-Luna protocol. They aim to enhance the system's redemption capacity during periods of crisis and high volatility.

4.1. Modifying the *TerraPool*$_\delta$ Mechanism

At the core of Terra's stabilization mechanism is the concept of *virtual liquidity pool* (VLP) [4], that in our case corresponds to $\Pi^{virtual}_{T_s,T_v}$, i.e., the pool containing both the stable (UST) and the volatile (LUNA) tokens, respectively. Initially, it comprises an equal quantity of $Pool_{Base}$ units for T_s and T_v. In equation (3) we introduced the parameter δ, which is equal to the difference between the current quantity of T_s and the baseline quantity $Pool_{Base}$ in the $TerraPool_\delta$ stabilization mechanism. Under these hypotheses, at each iteration n we have:

$$K(n) = Pool^2_{Base} \cdot \frac{1}{P_v(n)} \qquad Q_s(n) = Pool_{Base} + \delta(n) \qquad Q_v(n) = \frac{K(n)}{Q_s(n)} \tag{12}$$

Each swap executed within the pool dynamically alters the value of $\delta(n)$.

In the original implementation of the pool replenishing mechanism of equation (3), $\delta(n)$ goes down to zero only when $n \to \infty$. So, we suggest a different method that acts similarly, but can bring $\delta(n)$ to zero after exactly $PoolRecoveryPeriod$ (PRP) blocks. The idea is very simple. Let's consider the scenario where the initial swap within the pool at time $n = 0$ involves an amount of x stable tokens T_s. Then the replenishment of x tokens in the VLP is spread among a list of x/PRP chunks, which will supply δ in the subsequent PRP time instants of the simulation. More formally, let A be a vector of length PRP initially empty; at the time instant $n = 0$ it is filled with the PRP chunks:

$$A = \left[\left(\frac{x}{PRP}\right), \left(\frac{x}{PRP}\right), ..., \left(\frac{x}{PRP}\right)\right] \qquad |A| = PRP$$

as a consequence of the swap. The first element A_1 of A will determine the value of $\delta(1)$ as follows:

$$\delta(1) = \delta(0) - A_1$$

Simultaneously, during this iteration, a swap operation takes place within the virtual pool denoted as $Swap(T_i, q)$, where $T_i \in \{T_s, T_v\}$. Consider $Q = [q/PRP, q/PRP, ..., q/PRP]$, where $|Q| = PRP$. Consequently, the update of vector A takes place according to the following procedure:

$$\begin{cases} A(n) = A(n-1) + Q & \text{if } T_i = T_s \\ A(n) = A(n-1) - Q & \text{if } T_i = T_v \end{cases} \tag{13}$$

After this, $\delta(n)$ is updated as follows, where now A_1 is the first element of $A(n)$:

$$\delta(n) = \delta(n-1) - A_1$$

Finally, the elements of the vector A are shifted left and the last element is set to zero:

$$A(n) = [A_1, A_2, ..., A_{PRP}] \xrightarrow{shift} A(n+1) = [A_2, A_3, ..., A_{PRP}, 0]$$

This process guarantees that, if no swaps occur, the virtual pool is fully replenished after exactly PRP blocks. Based on this setup, the improvement of the stability we are going to propose pertains to a variant of the VLP replenishing mechanism suggested above. If the variation of δ exceeds $\lambda \cdot Base_{Pool}$ (where $\lambda = 0.05$ in our simulation), the array $A(n)$ is treated as if it had a length of $PRP/2$. Essentially, in a crisis scenario, the virtual pool's redemption capacity doubles. Given that the variation of δ correlates directly with the price $P_s(n)$ of T_s (as shown in [13]), this modification enables the system to recover more rapidly from a peg loss and ensures that δ returns to zero in half the original time.

4.2. Implementing a UST Reserve Pool

This improvement involves the implementation of a reserve pool that the protocol can use to automatically buy back T_s, i.e., UST, if its price falls below the peg. The reserve pool functions as a collateral for the token T_s. The purpose is to utilize this pool during a crisis to quickly restore $P_s(n)$ to the peg. The reserve pool is filled with $R(n)$ USDT and, in our simulation, we have set $R(n) = 0.2 \cdot (\text{total supply } T_s)$. If $P_s(n)$ falls significantly below the peg, the protocol uses some of the reserves to buy T_s from Π^S. This mechanism ensures that T_s can efficiently recover the peg value even in adverse market conditions, at least as far as the pool has tokens to use. At iteration n, the quantity x of USDT to sell is determined by the following system of equations, derived from formulas (4) and (5), where $\widetilde{Q}_s(n)$ is the quantity of T_s when $P_s(n) = 0.95$:

$$\begin{cases} x = \frac{k}{\widetilde{Q}_s(n)} - Q_U(n) \\ 0.95 = \frac{k}{\widetilde{Q}_s(n)^2 + \widetilde{Q}_s(n)} \end{cases} \implies \begin{cases} x = \frac{k}{\widetilde{Q}_s(n)} - Q_U(n) \\ \widetilde{Q}_s(n) = \frac{\sqrt{\frac{0.95+4k}{0.95}} - 1}{2} \end{cases}$$

4.3. Implementing a BTC Reserve Pool against LUNA Hyperinflation

This approach is an automatization of the attempt made by the Luna Foundation Guard to support the value of UST, when during the agitated phases of the collapse they sold at the market about 80 000 BTC [22]. The proposed solution aims to prevent the hyperinflation of the volatile token T_v, which results from its excessive minting due to the usage pressure of the Terra-Luna protocol of Figure 1. It is well known that the number of LUNA in circulation soars from 340 million to 6.5 trillion at the end of the collapse event [23].

When the value of the stablecoin T_s reaches a critical level – e.g., \$$0.95$ – a queue system is automatically activated, which involves a reserve of BTC. At this point, a user has three options available:

1. sell her/his own T_s directly on the market;
2. use the classic stabilization protocol, burning 1 T_s to obtain \$1 worth of T_v;
3. burn 1 T_s to obtain \$1 worth of BTC.

It is necessary to clarify some aspects. For the correct functioning of the system, the presence of a reliable oracle, which provides the real-time price of BTC, is essential. Moreover, the users will not receive real BTC, as the latter is a native cryptocurrency of another blockchain. Instead, they will receive \$1 worth of wrapped BTC (wrBTC), which is a synthetic token that represents the ownership of a BTC on a blockchain different from the original one. Obviously, it is not possible to prevent users from selling their own T_s on the market. However, it is possible to induce them to undertake option 2 or option 3 through the creation of a double-queue system: queue A collects transactions of users eager to exchange T_s tokens for BTC, while queue B contains transactions of those who prefer to normally use the protocol, obtaining \$1 worth of T_v for each T_s inserted.

Both options offer the benefit of reducing the circulating supply of T_s, but queue A has the additional advantage of avoiding the minting of T_v. A probability function can govern the mechanism by selecting, at each time interval, which queue to activate. Both queues follow the FIFO (First In First Out) principle. The probability function can be a simple logistic function, which essentially depends on two parameters:

- the deviation of the price $P_s(n)$ from the parity;
- the filling percentage of the BTC reserve.

In our simulation, the function $P_A(x)$ returns the probability that queue A is activated, and is defined as follows:

$$P_A(x) = \frac{1}{1 + e^{-\kappa(x-\alpha)}} \tag{14}$$

where κ determines the slope of the curve and α establishes the point on the x-axis where the graph has an inflection point. These parameters can be manipulated to obtain a function that conforms to one's preference. The independent variable x could be set as a combination of the filling percentage of the BTC reserve and the deviation from parity, for example the product of the two. To reproduce the market context before the collapse of the Terra-Luna system, the 100% of the reserve could be considered equal to \$1B in BTC. When the reserve is empty, the probability of queue A being selected is set to zero.

5. Simulation Results

We carried out two kinds of simulations. The first one evaluates the performance of the original stabilization mechanism of the Terra protocol under normal market conditions, characterized by

Collapses		σ_Δ	σ_Δ	σ_Δ
PRP	$Pool_{Base}$	10^{-4}	$5 \cdot 10^{-4}$	10^{-3}
24	10^5	0	2	7
24	$5 \cdot 10^5$	0	0	0
24	10^6	0	0	0
36	10^5	0	1	4
36	$5 \cdot 10^5$	0	0	0
36	10^6	0	0	0
48	10^5	0	3	4
48	$5 \cdot 10^5$	0	0	2
48	10^6	0	0	0

Table 1: Collapses of T_s in 10 simulations

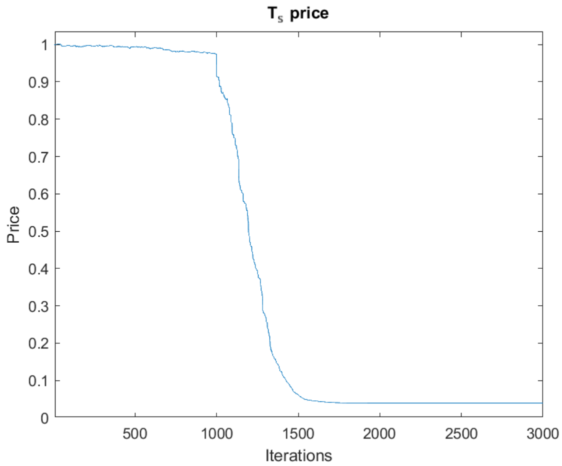


Figure 2: T_s price collapse across iterations

constant volatility σ_Δ. In the second one, we simulated a crisis scenario, marked by escalating volatility, to induce a collapse and to test the efficacy of the three proposed improvements discussed in the preceding sections.

Firstly, we conducted a series of 10 simulations, each with different constant values of parameters σ_Δ, $PoolRecoveryPeriod$, and $Pool_{Base}$, resulting in a comprehensive total of 270 simulation runs. The parameter σ_Δ plays a pivotal role in determining market volatility, while $PoolRecoveryPeriod$ and $Pool_{Base}$ determine the redemption capacity of the stability mechanism. In each simulation, T_s starts at \$1, with a total supply of 10^7, while Π^S and Π^V initially consist of $2 \cdot 10^6$ units of T_s and T_v, respectively. The price of the volatile token is initially set to $P_v(0) = \$100$. Results are shown in Table 1. We considered T_s collapsed when its price falls below \$0.50 and the system is not able to recover. Figure 2 shows the behavior of the price of T_s during a collapse.

In the second series of tests, we conducted 30 simulations with the goal of inducing the system to collapse by gradually increasing the volatility σ_Δ from $\sigma_\Delta = 0.0001$, with a 0.00002 step every 1000 iterations, with a total of 100 000 iterations. Figure 3 illustrates the behavior of P_s under these conditions. The results are shown in Table 2, and make it evident that the original implementation of Terra is unable to withstand such scenarios, with the peg being lost in 28 out of 30 simulations.

Better results are obtained with the modification of the *TerraPool*$_\delta$ replenishing mechanism and the UST Reserve Pool. As for the BTC Reserve Pool improvement, we performed two separate sets of tests. The first one involved the use of the Terra standard stabilization protocol, without any queue system. The second one incorporated the queue system described in Section 4.3. In both scenarios, we conducted 100 runs of the simulation program, monitoring and recording the circulating supply of T_v (LUNA). The results are shown in Figure 4.

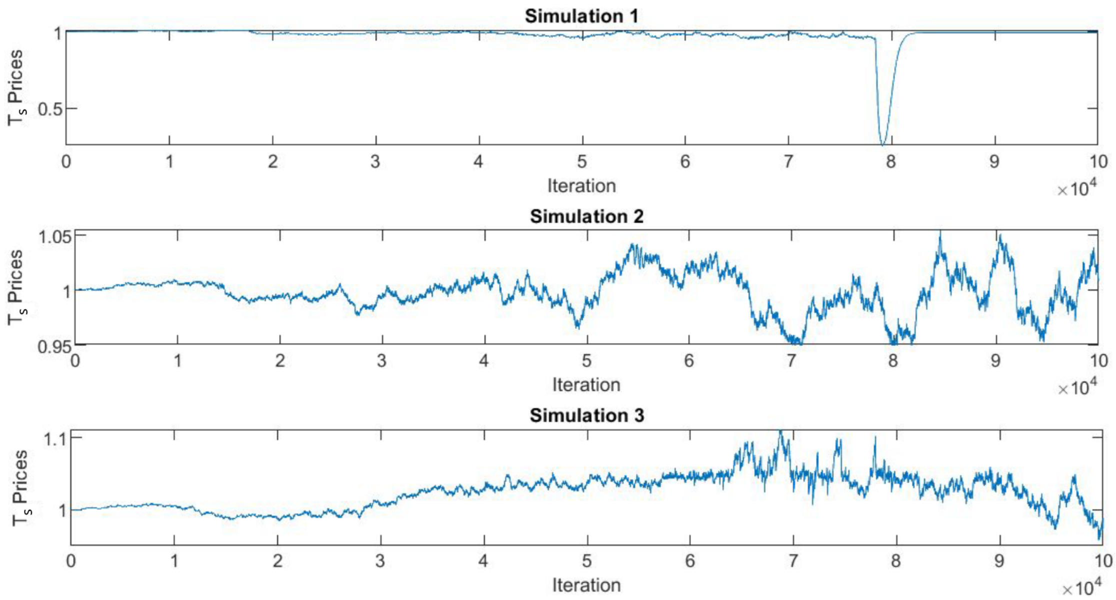


Figure 3: Price variation of T_s under gradually increasing volatility using the original Terra protocol: Simulation 1 depicts a collapse, indicated by the peak of P_s, while Simulations 2 and 3 illustrate the protocol's resilience to volatility crises

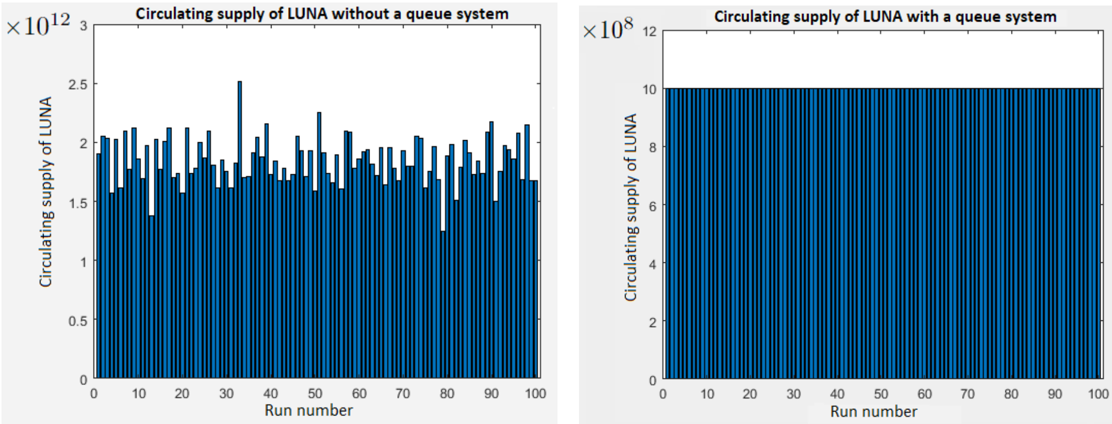
Simulation	Original	*TerraPool*$_\delta$ replenishing	UST Reserve Pool
Total collapses	28	24	7
Mean collapse time	71012	70278	86559

Table 2: Number of collapses, with increasing value of the volatility σ_Δ, for the original protocol and for the first two improvement proposals

6. Discussion and Conclusions

The first conclusion we can draw is the structural weakness of the Terra-Luna protocol, since Table 2 shows an astonishingly high number of collapses as a consequence of an increasing volatility σ_Δ of the market (28 out of 30). Also, the huge number of circulating supply of LUNA, of the order of 10^{12} and described in Figure 4a, is a clear warning of this structural weakness. Even if a perfect algorithmic stablecoin should resist all kinds of market destabilization forces, with only "0" in the second and third columns of Table 1, the improvements we obtain with the three proposals shed light on the mechanism one could implement to increase the strength of an algorithmic stablecoin protocol.

The *TerraPool*$_\delta$ method indicates that it is possible to work on the replenishment protocol with a mild rate of success (24 collapses out of 30 instead of 28). Significantly better results are obtained with the use of a UST Reserve Pool (7 collapses out of 30). Also, the method of limiting the LUNA supply with the queue system implementing the BTC Reserve Pool shows a good performance, reducing to 10^9 the original supply of LUNA of order 10^{12}. Note that this huge



(a) Circulating supply of LUNA in the original protocol

(b) Circulating supply of LUNA with the queue system implementing the BTC Reserve Pool

Figure 4: Supply of LUNA without and with the queue system described in Section 4.3

supply was the main cause of the crash in the LUNA value and of the Terra-Luna collapse. Note, moreover, that this last method guarantees an almost constant value of supply for all simulations. These evaluations underscore the critical importance of incorporating reserve pools and refining virtual pool replenishing mechanisms to improve the stability of algorithmic stablecoin systems in the face of market volatility. The utilization of a reserve pool provides a significant cushion against adverse market movements, acting as a stabilizing force during periods of heightened volatility, and the obtained data serve as a clear illustration of the robustness exhibited by this hybrid collateralized-algorithmic stablecoin. It effectively demonstrates the coin's ability to resist challenges posed by a highly volatile market. In general, our solutions extend beyond Terra-Luna, adapting to diverse seigniorage stablecoin frameworks.

As far as the limitations of our approach are concerned, while the introduction of a reserve pool shows promising results, it moves away from the notion of a pure algorithmic stablecoin, transitioning towards a partially collateralized stablecoin model. Future research should address these trade-offs and explore avenues for optimizing stability while maintaining algorithmic integrity. Another limitation is related to the design choices we made. They are an inevitable approximation of the human behavior in financial markets. Anyway, we tried to be as general as possible in designing these phenomena, since our simulations are initializable with different parameters, allowing for flexibility in capturing various market conditions and behaviors. A critical aspect for future exploration lies in refining the parameters and assumptions underlying our simulation model. This could be achieved by incorporating real-world data and historical market trends.

Moving forward, our research suggests several interesting directions. Further investigation into alternative replenishment protocols and reserve pool mechanisms could yield insights into improving the stability of this type of system. Our simulation-based approach could help researchers to effectively design and evaluate new algorithmic stablecoin protocols, offering valuable information about their performance and robustness in various market conditions.

Acknowledgments. This research has been supported by the PRIN 2020 project *NiRvAna – Noninterference and Reversibility Analysis in Private Blockchains.*

Code Repository. The Matlab code implementing our simulations is publicly available at https://github.com/FedericoCalandra/Algorithmic-Stablecoin-Collapse-Simulation.

References

[1] K. Qin, L. Zhou, Y. Afonin, L. Lazzaretti, A. Gervais, CeFi vs. DeFi — Comparing Centralized to Decentralized Finance, arXiv preprint arXiv:2106.08157 (2021).
[2] Ampleforth, #ampl the decentralized unit of account, https://www.ampleforth.org/, 2022. Accessed: 2024-03-10.
[3] E. Kereiakes, D. Kwon, M. Di Maggio, N. Platias, Terra money: Stability and adoption, White Paper (2019).
[4] terra.money, Liquidity parameters, https://classic-agora.terra.money/t/liquidity-parameters-3/3895, 2022. Accessed: 2024-03-01.
[5] Y. Zhang, X. Chen, D. Park, Formal specification of constant product (xy= k) market maker model and implementation, White paper (2018).
[6] A. Briola, D. Vidal-Tomás, Y. Wang, T. Aste, Anatomy of a stablecoin's failure: The Terra-Luna case, Finance Research Letters 51 (2023) 103358.
[7] F. van Echelpoel, M. T. Chimienti, M. Adachi, P. Athanassiou, I. Balteanu, T. Barkias, I. Ganoulis, D. Kedan, H. Neuhaus, A. Pawlikowski, et al., Stablecoins: Implications for monetary policy, financial stability, market infrastructure and payments, and banking supervision in the Euro area, ECB Occasional Paper, European Central Bank 247 (2020).
[8] C. Calcaterra, W. A. Kaal, V. Rao, Stable cryptocurrencies: First order principles, Stanford Journal of Blockchain Law & Policy 3 (2020) 62–64.
[9] A. Kosse, M. Glowka, I. Mattei, T. Rice, Will the real stablecoin please stand up?, BIS Papers, Bank for International Settlements 141 (2023).
[10] L. Ante, I. Fiedler, J. M. Willruth, F. Steinmetz, A systematic literature review of empirical research on stablecoins, FinTech 2 (2023) 34–47.
[11] J. Choi, H. H. Kim, Stablecoins and central bank digital currency: Challenges and opportunities, Oxford Research Encyclopedia of Economics and Finance (2024). URL: https://ssrn.com/abstract=4756822. doi:10.2139/ssrn.4756822.
[12] R. Clements, Built to fail: The inherent fragility of algorithmic stablecoins, SSRN (2021).
[13] J. Cho, A token economics explanation for the de-pegging of the algorithmic stablecoin: Analysis of the case of Terra, Ledger 8 (2023) 36–44.
[14] J. Liu, I. Makarov, A. Schoar, Anatomy of a Run: The Terra Luna Crash, Technical Report, National Bureau of Economic Research, 2023.
[15] G. Kurovskiy, N. Rostova, How Algorithmic Stablecoins Fail, Technical Report, SNB BNS, 2023.
[16] H. Uhlig, A Luna-tic stablecoin crash, Technical Report, National Bureau of Economic Research, 2022.
[17] S. Ferretti, M. Furini, Cryptocurrency turmoil: Unraveling the collapse of a unified

stablecoin (USTC) through Twitter as a passive sensor, Sensors 24 (2024) 1270. URL: https://doi.org/10.3390/s24041270. doi:10.3390/s24041270.
[18] V. Mohan, Automated market makers and decentralized exchanges: A DeFi primer, Financial Innovation 8 (2022) 20.
[19] Uniswap, Pricing in uniswap, https://docs.uniswap.org/sdk/v2/guides/pricing, 2018. Accessed: 2024-02-28.
[20] Blockchair Wallet Data, Blockchair, https://blockchair.com/, 2023. Accessed: 2024-03-01.
[21] BitInfoChart, Bitcoin rich list, https://bitinfocharts.com/top-100-richest-bitcoin-addresses.html, 2024. Accessed: 2024-03-02.
[22] Coindesk, Lfg reserves dwindle to just 313 bitcoins from 80k after UST crash, https://www.coindesk.com/business/2022/05/16/luna-foundation-guard-left-with-313-bitcoin-after-ust-crash, 2022. Accessed: 2024-03-02.
[23] T. Block, Luna supply soared to 6.5 trillion coins before Terra's latest halt, https://www.theblock.co/post/146762/luna-supply-soared-to-6-5-trillion-coins-before-terras-latest-halt, 2022. Accessed: 2024-03-02.