Algorithmic Stablecoins: A Simulator for the Dual-Token Model in Normal and Panic Scenarios

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

[1]Dipartimento di Scienze Pure e Applicate, Università di Urbino, Urbino, Italy
federico.calandra@unicam.it, f.rossi51@campus.uniurb.it, marco.bernardo@uniurb.it

[2]Dipartimento di Matematica, Informatica e Geoscienze, Università di Trieste, Trieste, Italy
ffabris@units.it

***Abstract*—In the decentralized finance landscape, algorithmic stablecoins offer a promising solution for stabilizing the value of cryptocurrencies without relying on centralized collaterals. However, models like the dual-token system are vulnerable to depeg events, as demonstrated by the catastrophic collapse of the Terra-Luna ecosystem in 2022, which saw over 50 billion dollars in market capitalization evaporate in just a few days. This work proposes *DualTokenSim*, a Python simulator designed to analyze the behavior of cryptocurrencies based on the dual-token model under both normal and panic scenarios. The simulator uses automated market makers and a stochastic process to simulate price dynamics and user behavior. The aim is to offer an environment in which to explore and analyze solutions for improving the resilience of algorithmic stablecoins during periods of market instability.**

***Index Terms*—algorithmic stablecoin, decentralized finance, simulation, Terra-Luna, dual-token seigniorage model**

I. Introduction

The financial sector, once dominated only by centralized entities, is going to be gradually transformed by the blockchain technology, leading to the rise of *decentralized finance* (DeFi). In this new paradigm, financial services are driven by smart contracts rather than central authorities, thus becoming accessible even to unbanked individuals with just a phone and an Internet connection [1].

DeFi platforms leverage cryptocurrencies as the primary medium of exchange within their ecosystems. However, the inherent volatility of these assets presents challenges for their use in DeFi applications. To address this issue, *stablecoins* were introduced, i.e., special cryptocurrencies specifically designed to maintain stable value by being pegged to a fiat currency or other assets. While collateralized stablecoins ensure stability through financial backing, they introduce a different degree of centralization, which contradicts the decentralized philosophy of DeFi because a company is typically responsible for managing the collateral and facilitating its redemption. This introduces the need for trust in the company behind the stablecoin. A notable example is *Tether*, the company behind USDT, which has faced criticism for its lack of transparency and admitted in the past that USDT is not fully backed by collateral [2]. For these reasons, *algorithmic stablecoins* (ASs) emerge as an innovative and original solution, eliminating the need for collateral while maintaining price stability through algorithms. This makes them an ideal choice for achieving complete decentralization.

There are two main models of AS: *rebasing* and *dual-token* (or *seigniorage*). The *Ampleforth* protocol (AMPL) [3] is an example of the rebasing model, where the total supply of AMPL is automatically adjusted based on its price relative to a fiat currency. This adjustment occurs directly in users' wallets, by increasing the number of their tokens when the price is above the peg or decreasing it when the price falls below. On the other hand, the Terra-Luna ecosystem [4] is an example of the dual-token model, where a *collateral token* (CT), i.e., *LUNA*, is used to absorb fluctuations in the value of the AS, i.e., *TerraUSD* (UST). This process helps stabilize UST's price by minting or burning tokens as needed, often through the exploitation of arbitrage opportunities.

Despite their promise, ASs have faced significant failures in their brief history, such as the resounding collapse of the Terra-Luna ecosystem in 2022, which saw over 50 billion dollars in market value evaporate in a few days [5]. This underscores the vulnerability of a stablecoin without collateral backing.

This article introduces *DualTokenSim*, a Python simulator designed to analyze the behavior of an AS and its CT within a dual-token model, both under normal market conditions and during periods of panic. The price dynamics of the tokens and user behavior during panic scenarios are modeled by means of a stochastic process and leveraging the simplicity of *Automated Market Makers* (AMMs).

DualTokenSim enables the analysis of how an AS can respond to potential solutions designed to enhance its resilience during periods of crisis and allows for the adjustment of various parameters, providing flexibility in exploring different scenarios. The ability to observe the behavior of an AS in a simulated environment is highly valuable, as these stablecoins represent an ideal solution within the DeFi landscape.

The rest of the paper is organized as follows. In Section II, we provide an overview of the basics of AMMs and the dual-token model, along with a brief review of state-of-the-art research on stablecoins. In Section III, we describe the fundamental architecture of DualTokenSim, focusing on how it models the price dynamics of both AS and CT as well as how it captures user panic behavior when AS loses its peg. In Section IV, we outline the validation process used to evaluate DualTokenSim performance against real-world data. In Section V, we present the results obtained from our simulations and assess whether DualTokenSim performs effectively in realistic scenarios. Finally, Section VI concludes the paper.

II. Background

In this section we explore the fundamentals of a dual-token model, used nowadays at least for FRAX and USDD, and how prices can be simulated through a straightforward formula. We shall use the Terra-Luna ecosystem as a reference.

In Section II-A, we discuss the fundamentals of Automated Market Makers, focusing specifically on the constant-product pricing formula, the most widely used mechanism in this class of systems. Subsequently, in Section II-B, we examine the Terra algorithmic market module and its role in maintaining the price stability of its stablecoins. Finally, in Section II-C, we present an overview of existing research based on stablecoins and the tools available for simulating cryptocurrencies behavior.

A. Automated Market Makers

AMMs play a pivotal role in the DeFi ecosystem by enabling the exchange of one cryptocurrency for another without the need for a central authority acting as an intermediary. This is the leading philosophy of *decentralized exchanges* (DEX), where the exchange process is solely driven by a simple formula and few lines of code [6]. Unlike traditional centralized order-book-based exchanges, AMMs operate by using *liquidity pools* (LPs), i.e., simple smart contracts that hold token reserves.

The simplest type of AMM, known as *Constant Product Market Maker* (CPMM), operates by using a *constant-product* formula, which ensures that the product of the two token reserves in the pool remains constant: $k = x \cdot y$ [7]. Here, x and y represent the reserves of the two tokens in the pool, and k is a constant known as the *invariant* of the pool. The token balances are dynamically adjusted with each swap in such a way that their product remains constant. This mechanism is essential in DualTokenSim, as it provides a simple way to model token trades and observe how their prices fluctuate in response, effectively simulating the dynamics of a free market.

When discretizing time, we can think of each time interval as a trade taking place within a CPMM that changes its state. More precisely, each CPMM is described by a LP Π_{T_a,T_b} consisting of two tokens T_a and T_b. At the discrete-time instant n, the state of the CPMM is defined by:

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

The invariant k actually changes over time due to the impact of transaction fees and variations in the liquidity of the LP contributed by users. For instance, in the case of *Uniswap*, the most used DEX with the highest *Total Value Locked* (TVL) [8], 0.3% of each transaction's value is retained as a fee and added to the LP. This increases the overall liquidity and, consequently, alters the value of k. However, in a simulated environment, we could set the fee to 0% and prevent any liquidity increments by users in the LP, thus effectively keeping k constant.

The CPMM state at a given time n can be written as:

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

A swap can be defined as a function that operates on the state of the CPMM. In this process, a user provides an input quantity q_a of token T_a to the LP and receives an output quantity q_b of token T_b from the LP, effectively purchasing T_b in exchange for T_a. The quantities exchanged are determined by the invariant curve that governs the CPMM. If the swap is performed at time $n+1$, the change in the supply of the output token T_b – which corresponds to the amount q_b purchased by the user – is given by:

$$q_b = Q_b(n) - Q_b(n+1) = \frac{k(n)}{Q_a(n)} - \frac{k(n+1)}{Q_a(n) + q_a} \quad (1)$$

In an AMM, the price of a token is measured in terms of the other token present in the LP. Going into details, the price $P_a(n)$ of token T_a at time n is determined by the ratio of the quantities of the two tokens inside the LP at that specific moment. Expressed in terms of T_b, it is given by:

$$P_a(n) = \frac{Q_b(n)}{Q_a(n)} \quad (\text{in } T_b/T_a) \quad (2)$$

This is what Uniswap refers to as the "mid price" [9]. 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 LP, without slippage[1] of the price.

Each swap alters the supply of the two tokens, thereby influencing their price, as described by Formula 2. This dynamics follows the "principle of scarcity": as the quantity of a token in the LP decreases, its price relative to the other token increases. Conversely, when a token becomes more abundant in the pool, its price tends to decrease. A typical example of the application of this principle is represented by gold and Bitcoin, both of which maintain high value due to their rarity.

B. The Terra Stabilization Mechanism

The Terra-Luna ecosystem was the most prominent example of a dual-token model, with its main AS, i.e., UST, supported by its CT, i.e., LUNA. For a few years the algorithm responsible for maintaining the peg to the reference value of $1 worked effectively, enabling the ecosystem to become the third largest one by market capitalization, surpassed only by Bitcoin and Ethereum. However, vulnerabilities in the algorithm and not ideal management of DeFi services offered by the Terra blockchain, such as the rich rewards promised by *Anchor*

[1]Slippage refers to the difference between the expected price of a trade and the actual price, which occurs when the trade size impacts the price due to insufficient liquidity.

protocol [10], led to a dramatic collapse in 2022, resulting in the loss of more than 50 billion dollars [5].

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 to mint or burn UST in response to price deviations from the peg. An arbitrageur is an individual or entity that engages in the practice of exploiting price discrepancies in different markets to make profits.
When the UST's market price falls below the peg, e.g., \$0.98, arbitrageurs can burn 1 UST obtaining automatically \$1 worth of LUNA from the protocol, making a \$0.02 profit per UST burnt. Conversely, if the 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. The buying or selling pressure on UST generated by arbitrageurs helps realign the value of the AS to the established level of one dollar. This realignment is further supported by the principle of scarcity: specifically, when UST becomes scarcer, its value tends to increase, while excessive availability can lead to a decrease in value.

This dynamics could be particularly problematic during a depeg event. In the case of the Terra-Luna collapse, panic-driven users began burning UST following the design of the protocol. However, this repeated action led to an excessive minting of LUNA, which ultimately caused its value to plummet. The massive increase in the supply of LUNA triggered a severe hyperinflation, with the number of LUNAs in circulation skyrocketing from 340 millions to over 6.5 trillions by the end of the collapse [11]. As a result, the value of both tokens crashed, exacerbating the crisis.

Figure 1 illustrates the steps arbitrageurs take to profit during a de-peg event, demonstrating how a dual-token model works to restore the price of its AS.

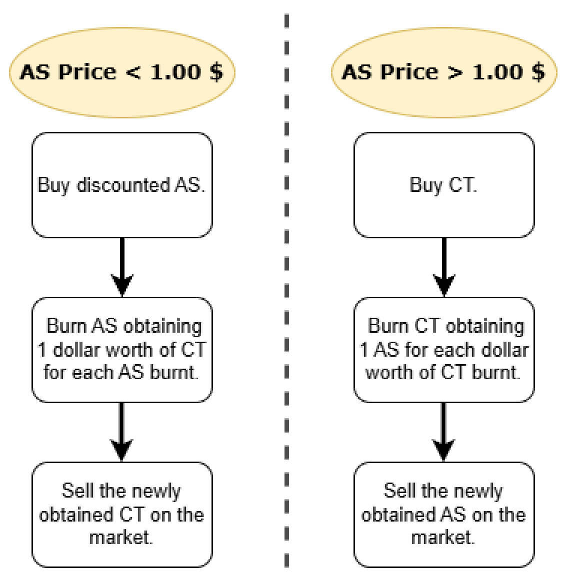

Fig. 1: Strategy used by arbitrageurs in a dual-token model.

The mechanism operates via the *virtual liquidity pool* (VLP) of the protocol, with LUNA's price sourced from validator oracles. The VLP is implemented through a variant of the classical CPMM algorithm described in Section II-A. The corresponding variant of the constant-product formula is defined as:

$$CP = Pool_{Base}^2 \cdot \frac{1}{Price_{LUNA}} \tag{3}$$

where $Pool_{Base}$ is the initial quantity of USTs in the pool, while the fraction $1/Price_{LUNA}$ expresses the price of LUNA in USD as observed in external markets [12]. $Price_{LUNA}$ is repeatedly updated by oracles, implying that the pool actively adapts to market fluctuations.

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

$$Pool_{UST} = Pool_{Base} + \delta, \quad Pool_{LUNA} = \frac{CP}{Pool_{UST}} \tag{4}$$

The dynamics of δ plays a crucial role in adjusting the LP 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 functionality of the market module is its ability to replenish the VLP, progressively bringing δ back towards zero. The rate of this replenishment is determined by the $PoolRecoveryPeriod$ parameter, defined in terms of blocks. At the end of each block – with one block being produced approximately every 6 seconds – δ is updated by changing it to:

$$\delta \cdot \left(1 - \frac{1}{PoolRecoveryPeriod}\right) \tag{5}$$

This formula governs the adjustment of δ, with $PoolRecoveryPeriod$ influencing the pace of the adjustment. This parameter was determined by the Terra community and, just before the time of the depegging 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 [12]. Note, as a consequence, that a full replenishment can be obtained only when the number of blocks tends to infinity.

C. Literature Review

Stablecoins have garnered interest from researchers and financial institutions due to their design and impact on financial stability and regulation. The ECB's Crypto-Assets Task Force addresses stablecoins in report n. 247 [13]. Calcaterra et al. [14] explore core design principles and their interrelations. Ante et al. [15] review 22 articles, highlighting types, benefits, risks, and regulatory challenges, along with research gaps like data scarcity. Clements [16] discusses the fragility of algorithmic stablecoins, citing risks from market incidents like Terra-Luna, while Zhao et al. [17] analyze volatility in algorithmic stablecoins by using theoretical and empirical methods to establish a framework for understanding market conditions.

The Terra-Luna ecosystem has been the subject of several studies, particularly regarding its challenges and the failure in

May 2022. Briola et al. [18] systematically analyze social media to describe the events leading to this failure, highlighting the project fragility and its reliance on the Anchor protocol. They also investigate the crash triggers using transaction data for BTC, LUNA, and UST. Uhlig [19] introduces a new theory and methodology to explain the gradual nature of crashes, offering insights based on this analysis.

Existing tools include ShardingSim [20], a modular simulator for committee-based sharding blockchains, and DAISIM [21], an open-source model of the collateralized DAI stablecoin. Our previous work [22] presents two MATLAB simulators designed to reproduce the dynamics of the Terra–Luna ecosystem. By contrast, DualTokenSim not only flexibly models any dual-token algorithmic stablecoin under diverse market conditions, but is also validated against on-chain data. To our knowledge, no other public framework combines such breadth with real-world validation.

III. DualTokenSim Overview

In this section we outline the fundamentals of DualTokenSim, our simulator developed in *Python*, by focusing on two key aspects: the management of price dynamics through the prototype of a CPMM and the simulation of user behavior during healthy and crisis scenarios. The simulation operates in discrete-time intervals called *iterations*, during which trades and arbitrage actions occur, altering the price of both AS and CT. This tool is highly flexible, as it allows users to adjust several parameters according to their preferences. Python was chosen for its versatility, ease of use, and extensive ecosystem of libraries, which make it ideal for implementing complex simulations and managing dynamic data.

In Section III-A, we detail the implementation of the LPs in DualTokenSim and explain how the prices of both AS and CT could be expressed in USD terms. Then, in Section III-B we briefly describe the management of tokens, their properties, and the distinctive characteristics of each token class. Finally, in Section III-C we discuss the stochastic process governing trades in the LPs, highlighting how these dynamics adapt based on whether AS is in a healthy or depegged state, as determined by its price.

A. Price Dynamics through a CPMM

We require a method to accurately replicate price fluctuations in the free market for both the AS and the CT, which together form the backbone of the dual-token model. This is achieved by leveraging the simplicity of CPMMs, which operate in discrete-time intervals referred to as iterations.

To simulate token prices, a separate LP is maintained for each token. As outlined in Formula 2, within the context of an AMM, the price of one token is expressed in terms of the other token in the same LP. To ensure consistency in pricing, it is essential to establish a fixed reference; this reference could be the USD, which serves as a stable benchmark for analyzing token price fluctuations. Since fiat currencies cannot be directly utilized in DeFi services, including CPMMs applications, we introduce a simplifying assumption. Specifically, we designate the second token in each LP as T_U, a fully collateralized stablecoin pegged to USD. This allows token values to be expressed in USD terms, under the assumption that T_U maintains a constant external value of 1 USD.

Two LPs are used to model the prices of AS and CT. The first LP, denoted by Π^{AS}, simulates the market operations of AS and consists of AS and T_U. The second pool, denoted by Π^{CT}, replicates the market dynamics of CT and consists of CT and T_U. At each discrete-time step in the simulation, random swaps occur within Π^{AS} and Π^{CT}, which alter their states and update the prices of AS and CT accordingly.

B. Token Management

Each token is represented as an object in the context of object-oriented programming, characterized by various attributes. These include the name of the token, its price expressed in USD, the total circulating supply, and the portion of the circulating supply not locked in smart contracts (such as AMMs). This subset of tokens, referred to as `free_supply`, is available for user trading or for exploiting arbitrage opportunities within the dual-token model protocol.

There are various token classes with distinct characteristics. For example, DualTokenSim could use generic volatile cryptocurrencies, and the token T_U mentioned in Section III-A is represented as a dummy token with a constant price of 1 USD and an infinite circulating supply. For managing a dual-token system, there is a dedicated token class representing AS and another for CT. Each AS object is tightly linked to a single CT object.

C. Stochastic Swaps

As mentioned in Section III-A, at each iteration a swap occurs within Π^{AS} and Π^{CT}. Now we need a method to determine the magnitude and type of each swap, i.e., how many units of AS and CT are bought or sold. This is accomplished by utilizing a stochastic process, which introduces randomness to reflect market dynamics.

The magnitude and type of swaps occurring within Π^{AS} and Π^{CT} are influenced by the health status of the market, which depends on the AS price. The market can either be in a *healthy* or *panic* scenario. The healthy scenario represents a normal market condition, where users act in a more rational manner. In contrast, when the market enters the panic scenario, a mechanism is triggered to simulate the irrational behavior of users, whose decisions are driven more by emotions than by rationality. It is understood that the AS tends to maintain its peg within the healthy scenario, while it is more likely to lose the peg when the market enters the panic scenario.

The boundary between the panic and the healthy scenarios is determined by a parameter called `threshold`. Assuming the peg is set at $1, the market is in a healthy scenario if the price of AS lies between $1 - \texttt{threshold}$ and $1 + \texttt{threshold}$. If the price falls below $1 - \texttt{threshold}$ or rises above $1 + \texttt{threshold}$, the market enters a panic scenario. However, the situation where the price of AS exceeds the $1 + \texttt{threshold}$ boundary is less critical, as it is typically driven by excessive enthusiasm, with users often buying AS impulsively. The more concerning situation occurs when the price drops below $1 -$ `threshold`, signaling a potentially harmful situation for AS due to impulsive selling by users. The `threshold` parameter is set to a default value of 0.05, but it can be adjusted as needed.

Both Π^{AS} and Π^{CT} have their own Gaussian distribution, each with a specific mean μ and variance σ^2. A reasonable approach is to start the simulation with a default normal distribution with $\mu = 0$ and $\sigma^2 = 1$. However, μ and σ^2 are parameters adjustable as needed for both LPs.

The Gaussian distribution associated with Π^{AS} (or Π^{CT}) determines the probability of buying/selling AS (or CT), depending on the sign of the number randomly picked at each iteration. A positive number corresponds to a sale, while a negative number corresponds to a purchase. When the mean is zero, the probability of selling the token is 50%. By adjusting the mean and shifting consequently the Gaussian shape, we can control the probability of selling the token, as shown in Figure 2.

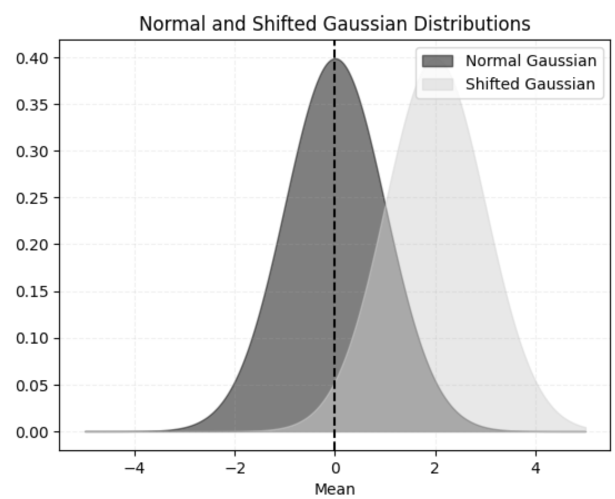

Fig. 2: Normal and shifted Gaussians.

The mean of the Gaussians is updated at each iteration. When the market is in a healthy scenario, i.e., the price of AS is greater than $1 -$ `threshold`, the mean μ remains zero. In such a scenario, purchases and sales of both AS and CT alternate with a 50% probability, suggesting that the peg will be probably maintained, unless disrupted by a series of unfortunate trades.
When the price of AS falls below $1 -$ `threshold` and the market enters a panic scenario, the mean of the Gaussian is updated according to a specific function. The default function used in our simulation is $f(x) = \frac{1}{x}$, since it aims to capture the irrational behavior of participants during such downturns. However, the framework is fully customizable, as it is possible to plug in any alternative function, and even assign different update rules to the AS and CT so that each distribution's mean evolves according to its own dynamics.
As the price of AS approaches zero, the mean of the Gaussian increases more rapidly, thereby raising the probability of selling both AS and CT. The mean is updated at each iteration according to the default function illustrated in Figure 3.

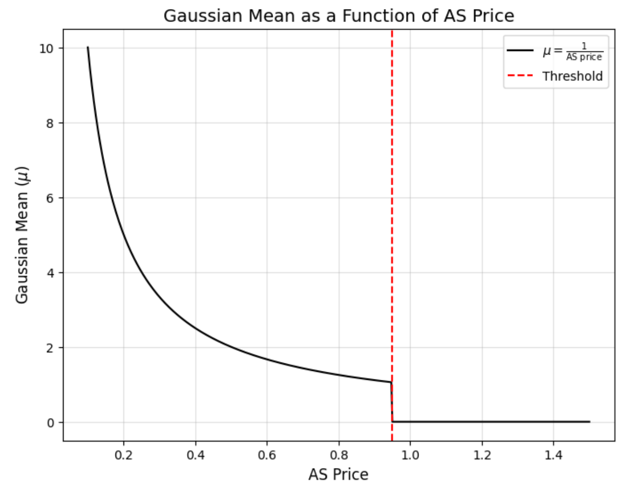

Fig. 3: Function used to update the Gaussian mean based on the AS price.

A normal Gaussian is not sufficient to determine the magnitude of the swap. At each iteration, the Gaussian distribution is scaled by a factor called `volatility`. The number drawn from the Gaussian is multiplied by this factor and the result represents the amount of the token, expressed in dollars, to be sold if positive or bought if negative. The `volatility` factor directly influences the magnitude of these swaps. To maintain uniformity, the amount of AS or CT to trade (depending on whether we are considering Π^{AS} or Π^{CT}) must be expressed in dollars.
The Gaussian is actually truncated to ensure that trades remain physically feasible. Specifically, this truncation prevents selling an amount exceeding the `free_supply` of the token or purchasing an amount larger than the portion locked in the specific LP. The limits of the truncated Gaussian are defined as follows, where the specified variables can refer to either AS or CT:

$$a = \frac{(\texttt{free\_supply} - \texttt{total\_supply}) \cdot \texttt{token\_price}}{\texttt{volatility}}$$

$$b = \frac{\texttt{free\_supply} \cdot \texttt{token\_price}}{\texttt{volatility}}$$

The `volatility` parameter also affects these bounds, thereby influencing the range of possible trade amounts. DualTokenSim allows for the replication of real trading volumes by treating `volatility` as a list of values, each corresponding to a specific trading volume within a given time interval. In summary, the dollar amount of the token to be traded is determined by sampling from a truncated Gaussian distribution, which depends on four parameters:

$$\texttt{dollars\_trade\_amount} = \texttt{truncnorm}(a, b, \mu, \sigma)$$

This dollar amount is then converted into the corresponding number of tokens, i.e., `trade_amount`, which represents the number of tokens to be traded in the LP:

$$\texttt{trade\_amount} = \frac{\texttt{dollars\_trade\_amount}}{\texttt{token\_price}}$$

Simultaneously with the trades occurring in the market, the arbitrage operations described in Section II-B are simulated too. These operations modify the total supply of both AS and CT and generate buy and sell actions within the LPs with the goal of making a profit.

IV. Validation

Validation is an essential step in the development of any simulator. The accuracy and reliability of the output of a simulator depend on its ability to replicate real-world phenomena effectively. In a dual-token AS system, validation ensures that the simulator accurately captures the interactions between the AS and its CT under various market conditions, including normal market scenarios and collapse events.

To validate DualTokenSim, we modeled the collapse of UST that occurred in May 2022. This event provides a comprehensive test case due to its complexity and the availability of detailed real-world data on trading volumes and token supplies. This replication tests the robustness of DualTokenSim and helps us understand the mechanisms behind peg loss and market panic. The goal is to improve the reliability of DualTokenSim for analyzing and predicting similar systems under stress conditions.

A. Challenges in Validation

Validating a simulator like ours that models financial markets presents several challenges. Financial markets are complex systems influenced by a variety of factors, including trader behavior and psychology, market sentiment, liquidity, and external economic indicators. Capturing the full scope of these dynamics in a simulation is inherently challenging. We have made several approximations in modeling market dynamics. In DualTokenSim, we employ two LPs to represent market activities. However, this approach could be a simplification compared to real-world markets, where liquidity is distributed across numerous pools, exchanges, and participants with varying strategies and motivations.

A major challenge is replicating market behavior during extreme volatility. Panic selling, herd behavior, and irrational responses can disrupt normal trading patterns, thus making accurate modeling difficult. Additionally, aligning the parameters of DualTokenSim with real-world data requires careful calibration due to the often noisy and incomplete nature of market data.

The dual-token model is characterized by complex interactions between the AS and the CT, governed by algorithmic rules that lead to non-linear behaviors. As a consequence, fine-tuning of the simulation parameters using real data is not a trivial task.

B. Validation Approach

We simulated the Terra ecosystem from May 1, 2022, to May 30, 2022. The validation process involved the following key steps:

1) *Data Acquisition*: We collected real-world daily trading volumes and circulating supplies for UST and LUNA over the period leading up to and during the collapse [23].
2) *Mapping Data to Simulator Parameters*: The collected data were used to calibrate the parameters of DualTokenSim, particularly the `volatility` parameters and the initial conditions for the LPs.
3) *Simulation Execution*: DualTokenSim was run over the simulated 30-days period, generating transactions for both the AS and the CT based on the calibrated parameters.
4) *Results analysis*: we compared the obtained prices and supplies variation against the real data of UST and LUNA.

By closely aligning the inputs of DualTokenSim with actual market data, we aimed to reproduce key aspects of the UST collapse, such as the loss of peg by the stablecoin, increased selling pressure, and the consequent impact on the value of CT. The validation required careful mapping of real-world data into our framework and the empirical calibration of parameters to replicate market behaviors. DualTokenSim operates in discrete-time steps. We chose to model the collapse by using a block-level granularity, with each iteration corresponding to a block generation event on the Terra blockchain. Since the Terra blockchain was built on the Cosmos ecosystem, a new block was produced approximately every 6 seconds. Consequently, each iteration of the simulation represents 6 seconds of real-world time. For each iteration, transactions are generated for both AS and CT by using the stochastic process presented earlier (III-C). To align the simulated trading volumes with the real-world daily trading volumes V_{daily}, we create a list of volatility values. During each iteration, we select a value v from this list to compute the next stochastic trading amount used in the swap for each token. The number of daily iterations is:

$$N_{\text{iterations}} = \frac{24 \cdot 60 \cdot 60}{6} = 14{,}400$$

The average volume per iteration is then:

$$V_{\text{iteration}} = \frac{V_{\text{daily}}}{N_{\text{iterations}}}$$

In DualTokenSim, the quantity q of each transaction is defined as:

$$q = \frac{|p| \cdot v}{P_{market}}$$

where P_{market} is the current market price of the token and p is a random variable sampled from a normal distribution:

$$p \sim \mathcal{N}(\mu, \sigma^2)$$

with mean μ and variance $\sigma^2 = 1$. The absolute value $|p|$ ensures that q is non-negative, while the sign of p determines the transaction direction (buy or sell). To match the expected per-iteration trading volume $V_{\text{iteration}}$, we set the volatility parameter v in such a way that:

$$V_{\text{iteration}} = \mathrm{E}[q \cdot P_{\text{market}}] = \mathrm{E}[|p|] \cdot v$$

where E[.] represents the expectation of the corresponding random variable Since p follows a normal distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$, the expected value of $|p|$ is given by the mean of the folded normal distribution:

$$\mathrm{E}[|p|] = \sigma\sqrt{\frac{2}{\pi}} \approx 0.7979 \quad \text{for } \sigma = 1$$

If we solve for v, we obtain:

$$v = \frac{V_{\text{iteration}}}{\mathrm{E}[|p|]} = \frac{V_{\text{iteration}}}{0.7979}$$

This calculation allows building the volatility list and then adjusting the volatility parameter v for each iteration, ensuring that the expected transaction volume matches the observed trading volumes over the simulation period. In the simulation code, this mapping is implemented in the function `calculate_volatility_array`, which computes v for each token based on its daily trading volumes.

To induce the system to collapse, we applied a selling pressure to UST by adjusting the mean μ of the normal distribution from which p is sampled. Initially, $\mu = 0$, which implies an equal likelihood of buy and sell transactions. Starting on the fifth simulated day (May 5, 2022), we increased μ to 0.1. This adjustment shifted the normal distribution, thus resulting in a greater proportion of positive p values (remember that $p \geq 0$ corresponds to sell transactions). Consequently, UST experienced significant sell pressure in the simulation. The choice of μ was determined through an iterative process of running the simulation and comparing the outcomes to real-world price trajectories of UST and LUNA during the collapse. By fine-tuning μ, we aimed to replicate key features of the event, including the rate of price decline, the volume of sell transactions, and the timing of the loss of peg by the stablecoin.

This empirical calibration involved balancing the sensitivity of the selling pressure to price changes with the overall stability of the simulation. A higher value of μ results in a more pronounced selling response to price declines, potentially leading to unrealistic market behaviors if set too high. Conversely, a lower μ may underrepresent the severity of panic selling observed in the real event.

C. Liquidity Pools Setup

To replicate the market dynamics during the collapse of the Terra ecosystem, we implemented three distinct LPs: a stablecoin-reference pool, a collateral-reference pool, and a VLP connecting AS and CT. Each pool was initialized with parameters derived from real-world data, so as to ensure consistency with observed market conditions.

The stablecoin-reference pool was constructed by using the AS and the reference token (USD). The initial quantity of the stablecoin in the pool $Q_{\text{pool,AS}}$ was calculated as the difference between the total initial supply of the stablecoin Q_{AS} and its free supply $Q_{\text{free,AS}}$ (i.e., the quantity available for trades):

$$Q_{\text{pool,AS}} = Q_{\text{AS}} - Q_{\text{free,AS}}$$

The quantity of the reference token Q_{USD} in the pool was then computed as $Q_{\text{USD}} = Q_{\text{pool,AS}} \cdot P_{\text{AS}}$, where P_{AS} is the initial market price of the stablecoin. A constant product formula governed the pricing mechanism of the LP, with a transaction fee of 0.3%. The collateral-reference pool was initialized by following the same procedure as the stablecoin-reference pool. In both cases, we determined that the quantity of free tokens is 80% of the total token quantity.

The VLP uses a seigniorage mechanism similar to that of the Terra ecosystem (II-B). The VLP is configured with the actual parameters from the Terra blockchain at the time of its collapse [24]. Specifically, the recovery period of the pool is set to 36 blocks and the base quantity of the stablecoin is set to $6.7215 \cdot 10^7$.

Finally, the panic scenario is triggered when the price falls below \$0.98 (i.e., `threshold` = 0.02). The panic functions that govern the selling pressures are defined as follows:

$$f_{\text{UST}} = \frac{1}{x/3} - 2.961224, \quad f_{\text{LUNA}} = \frac{1}{x \cdot 10} - 0.002041$$

To reflect the effective dynamics of the Terra collapse, we implemented a termination condition for the VLP mechanism. Specifically, the stabilization algorithm is halted when the LUNA market capitalization remains below 2% of the UST market capitalization for more than 5,000 consecutive iterations. This condition mirrors the actual deactivation of the Terra's stability mechanism on May 12, 2022, when the severe devaluation of LUNA rendered the algorithmic stabilization protocol ineffective.

V. Results

In this section we present the results of the simulation, highlighting key findings and their consistency with real-world data observed during the collapse of the Terra ecosystem in May 2022. The results are summarized graphically in Figure 4.

First, by appropriately tuning the simulation parameters, we were able to accurately replicate the timing of the collapse, which occurred on May 12, 2022. This is evident in the stablecoin price graph in Figure 4, where the simulated price trajectory of UST closely follows the real price decline, capturing the sudden depegging event with precision.

Second, the simulated price trajectory of LUNA also aligns closely with the real price behavior. The rapid decline in LUNA's value, reflecting the system's inability to stabilize the stablecoin, is consistent between the simulation and historical data.

Third, the final prices of UST and LUNA at the end of the simulation (May 30, 2022) were 0.211635 and 0.057334 USD, respectively. While these differ slightly from the real prices on the same date (0.025112 for UST and 0.000127 for LUNA, as detected on CoinMarketCap [23]), they remain within an

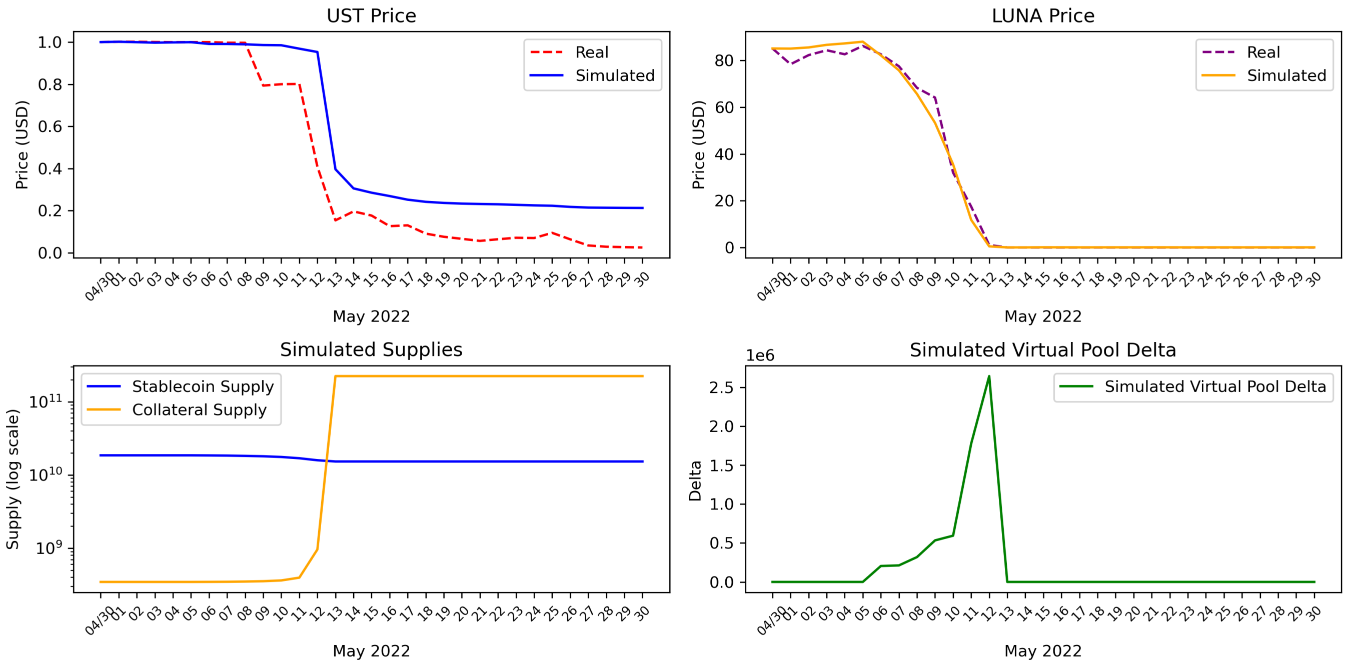

Fig. 4: Comparison of simulated and real data during the Terra-Luna collapse. The top-left panel shows the simulated and real price histories of UST (stablecoin), while the top-right panel illustrates the simulated and real price histories of LUNA (collateral token). The simulated values are sampled every 14,400 iterations (corresponding to 24-hour intervals), while the real values correspond to the closing prices recorded at 23:59:59 of each day. The bottom-left panel presents the market capitalizations of UST and LUNA, both simulated and real, while the bottom-right panel displays the evolution of the virtual pool δ, highlighting shifts in system dynamics over time.

acceptable range given the complexity of the system and the number of parameters one can control in the model.

Lastly, the changes in token supplies, a critical metric for dual-token algorithmic stablecoin systems like Terra-Luna, were captured effectively. The simulation observed a reduction in UST supply from $18.49 \cdot 10^9$ to $15.24 \cdot 10^9$ tokens, and an increase in LUNA supply from $3.453 \cdot 10^8$ to $2.232 \cdot 10^{11}$ tokens. These variations are consistent with historical data, which report a final UST supply of $11.27 \cdot 10^9$ tokens and a final LUNA supply of $6.536 \cdot 10^{12}$ tokens.

VI. Conclusions

Our Python simulator DualTokenSim effectively replicates the collapse of the Terra-Luna ecosystem by modeling price dynamics, the surge in LUNA supply, and market behavior during the depegging event. Its open-access nature allows for ongoing improvements and collaboration within the research community, with all code and technical details to be shared online.

Key areas for enhancement include:

- Model refinement, based on incorporating more market factors and different arbitrage dynamics for greater realism.
- Validation and improvement proposals, which serve as a testbed for evaluating modifications to the VLP mechanism and new stabilization techniques.
- Automating parameter fine-tuning, by using machine learning or optimization algorithms for more accurate and efficient parameter calibration.
- Quantitative stability evaluation using the Mean Squared Error (MSE) between the stablecoin price and its peg in balanced market scenarios.
- Stress-testing under extreme market conditions, including network congestion, flash crashes, and liquidity shocks.

One of the most promising applications of DualTokenSim is its ability to test new dual-token AS protocols under a wide range of market scenarios. By simulating stress conditions and analyzing the performance of proposed designs, developers can identify weaknesses and refine stabilization mechanisms before deploying them in live markets.

Acknowledgment This research has been supported by the PRIN 2020 project NiRvAna – Noninterference and Reversibility Analysis in Private Blockchains. The scholarship of the first author at the Italian PhD Program in Blockchain and Distributed Ledger Technology is funded by PNRR – Piano Nazionale di Ripresa e Resilienza according to D.M. 118/2023.

Code Repository Our open-source simulation code is available at https://github.com/FedericoCalandra/DualTokenSim.

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] CoinDesk, Tether lawyer admits stablecoin now 74% backed by cash and equivalents, available at: https://www.coindesk.com/markets/2019/04/30/tether-lawyer-admits-stablecoin-now-74-backed-by-cash-and-equivalents. Accessed: 2025-01-07 (2019).

[3] Ampleforth, #ampl the decentralized unit of account, https://www.ampleforth.org/, accessed: 2024-01-07 (2022).

[4] E. Kereiakes, D. Kwon, M. Di Maggio, N. Platias, Terra money: Stability and adoption, White Paper (2019).

[5] J. Liu, I. Makarov, A. Schoar, Anatomy of a run: The terra luna crash, Tech. rep., National Bureau of Economic Research (2023).

[6] V. Mohan, Automated market makers and decentralized exchanges: A DeFi primer, Financial Innovation 8 (1) (2022) 20.

[7] Y. Zhang, X. Chen, D. Park, Formal specification of constant product (xy= k) market maker model and implementation, White paper (2018).

[8] DappRadar, Top DeFi TVL – DEX, available at: https://dappradar.com/rankings/defi?category=defi_dex. Accessed: 2025-01-07 (2025).

[9] Uniswap, Pricing in uniswap, https://docs.uniswap.org/sdk/v2/guides/pricing, accessed: 2025-01-07 (2018).

[10] Anchor protocol, accessed: 2025-01-07.
URL https://docs.anchorprotocol.com/anchor-2

[11] 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.html, accessed: 2025-01-07 (2022).

[12] terra.money, Liquidity parameters, https://classic-agora.terra.money/t/liquidity-parameters-3/3895, accessed: 2025-01-07 (2022).

[13] 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).

[14] C. Calcaterra, W. A. Kaal, V. Rao, Stable cryptocurrencies: First order principles, Stanford Journal of Blockchain Law & Policy 3 (2020) 62–64.

[15] L. Ante, I. Fiedler, J. M. Willruth, F. Steinmetz, A systematic literature review of empirical research on stablecoins, FinTech 2 (1) (2023) 34–47.

[16] R. Clements, Built to fail: The inherent fragility of algorithmic stablecoins, SSRN (October 2021).

[17] W. Zhao, H. Li, Y. Yuan, Understand volatility of algorithmic stablecoin: Modeling, verification and empirical analysis, in: Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers 25, Springer, 2021, pp. 97–108.

[18] 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.

[19] H. Uhlig, A Luna-tic stablecoin crash, Tech. rep., National Bureau of Economic Research (2022).

[20] Y. Wu, Y. Wang, F. Yan, W. Chen, Shardingsim: A modular committee-based sharding blockchain simulator, in: 2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, 2024, pp. 273–278.

[21] S. Bhat, A. B. Kahya, B. Krishnamachari, R. Kumar, Daisim: A computational simulator for the makerdao stablecoin, in: 4th International Symposium on Foundations and Applications of Blockchain 2021 (FAB 2021), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[22] F. Calandra, F. P. Rossi, F. Fabris, M. Bernardo, Making Algorithmic Stablecoins More Stable: The Terra-Luna Case Study, in: 6th Distributed Ledger Technology Workshop (DLT 2024), Vol. CEUR-WS 3791, 2024, pp. 19:1–19:14.

[23] CoinMarketCap, Coinmarketcap: Cryptocurrency prices, charts and market capitalizations, https://coinmarketcap.com, accessed: 2025-01-06 (2025).

[24] Terra Community Forum, Terra community forum, https://classic-agora.terra.money/t/liquidity-parameters-3/3895, accessed: 2025-01-06 (2025).