NEST: Decentralized martingale network

nestprotocol.org

February 10, 2023

Abstract

This article primarily introduces and discusses the NEST protocol's innovative ideas and key mechanisms. We defined a new trading paradigm based on blockchain technology, martingale trades, in order to completely solve various problems of on-chain trades in a decentralized manner. And thus the martingale trading network was formed. Martingale trading is a low-cost and efficient trading paradigm for assets and returns with uncertain values. It is possible to say that martingale trading and martingale networks have given rise to a new economic theory that runs parallel to the market economy.

Contents

1 Background

Bitcoin (BTC) has built a decentralized currency paradigm, producing a completely new decentralized currency network that is being used by an increasing number of countries and people. Ethereum (Eth) has established a decentralized asset paradigm: ERC20 enables the development of on-chain applications such as DeFi, while ERC721 enables the capitalization and globalization of digital collections such as NFT, resulting in the formation of a brand new decentralized asset.

However, in terms of trading, the on-chain and off-chain worlds are essentially the same, with both continuing the Pareto trading paradigm of matching, which is the market trading mechanism that everyone refers to. This mechanism necessitates clear buyers and sellers, and transaction execution is legally protected. Smart contracts have replaced the role of the law's protector on the blockchain, lowering transaction trust risk. On-chain transactions are a novel concept that many people are excited about. However, there are additional issues in the on-chain matching, such as high cancellation costs, poor liquidity, a terminology meaning block generating delays, and so on.

To address these issues, Uniswap proposed the AMM (Auto Market Maker) mechanism, which provides buyers with a certain amount of liquidity by restricting seller behavior. However, arbitrage is easy due to limited price feedback, and a significant volume of TVL (Total Value Locked) brings waste of resources. This, in our opinion, is not the best solution for on-chain tradings. This article will propose a completely new trading paradigm based on current blockchain technology, martingale trading, as well as its corresponding decentralized trading network - martingale network. This new trading paradigm makes good use of the blockchain's technical characteristics of first confirming the existence of assets and then the ownership relationship, and it provides traders with unlimited liquidity through risk sharing. This revolutionary new trading paradigm is truly applicable to the world of on-chain trades.

2 Pareto Tradings and Market Network

The market is a mechanism in the traditional economy in which participants bargain and exchange based on their respective endowments. In

the same game network, thousands of rational people bargain. A Pareto equilibrium is reached when neither party's utility can be improved further without reducing the utility of the other party. We can call this trading mechanism based on bargaining as "Pareto trading". Theoretical equilibrium is not always easy to achieve or verify. Because complex factors such as transaction costs, information symmetry, and asset liquidity must be considered in the real world, equilibrium is only a theoretical concept under ideal conditions. Various intermediaries have emerged as important market participants in this context in order to reduce transaction costs, particularly search and matching costs.

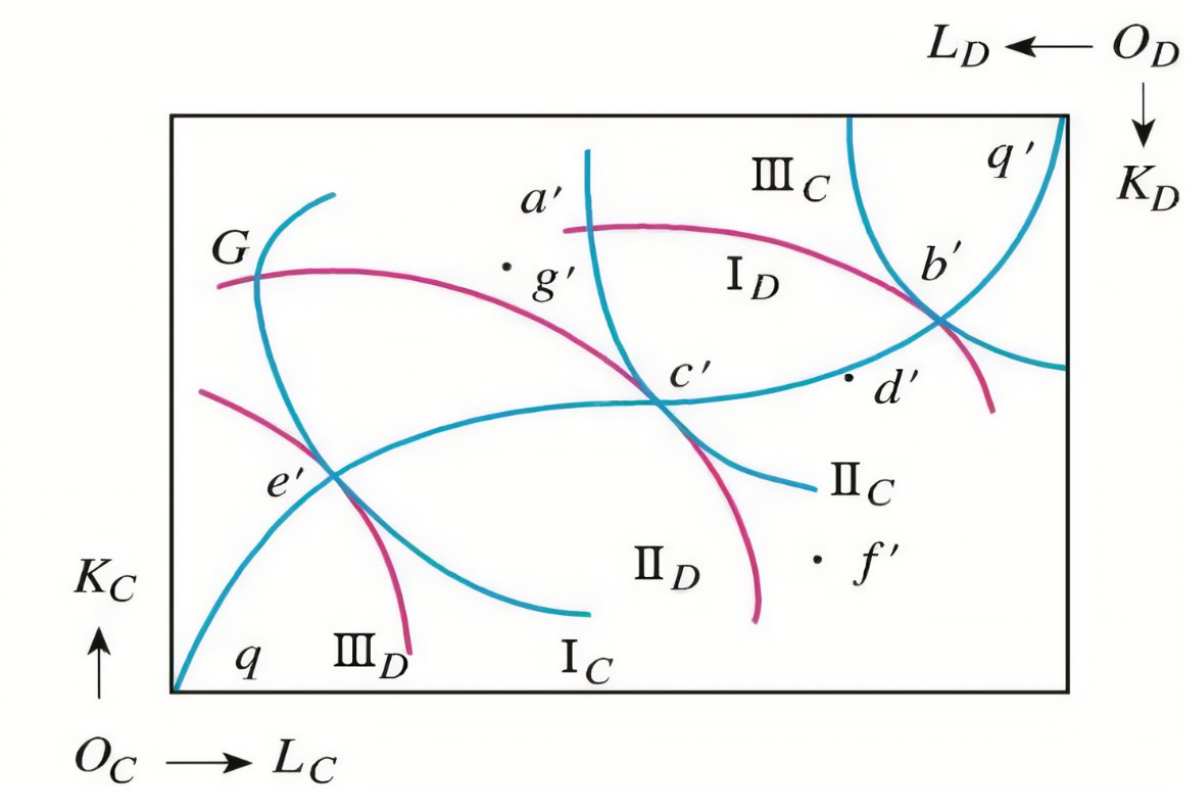


Figure 1: Edgeworth Box

This trading paradigm becomes ineffective when dealing with commodities, assets of uncertain value, or dealing with uncertain returns. For risky assets, for example, we cannot develop a theory based on tangents to utility indifference curves. In this context, we must develop new trading models.

3 Martingale Trading and Martingale Networks

We can define a new trading paradigm outside of Pareto trading when we consider a large number of uncertain returns in economic activities. This trading paradigm is based on the stochastic process concept: a random process X_t is called a martingale if it satisfies $X_t = \mathbb{E}(X_{t+s}|Xt)$ for any time t and any future time $t+s(s \geq 0)$. If a trader pays X_t at time t and receives X_{t+s} at time $t+s$ based on a martingale, we call this a martingale transaction (here s, which ≥ 0 is the time difference between transaction cash outflow and inflow. Outflow and inflow can happen at be the time points infinitely close to each other, or even the same real-time point, as in a smart contract transaction). A martingale network is formed when participants discover various martingales for risk and return and conduct martingale transactions. Although the above definition is straightforward, it reveals an important concept: when dealing with uncertain returns, we must trade using a martingale information flow to achieve a fair result.

The essence of the transaction is the same regardless of the value unit, whether it is apples, oranges, US dollars, or BTC. However, the goal of this article is not to develop a theory based on martingale trading, but to attempt to build a decentralized martingale trading network based on blockchain and digital currency, so we position the transaction target as a digital asset on the chain.

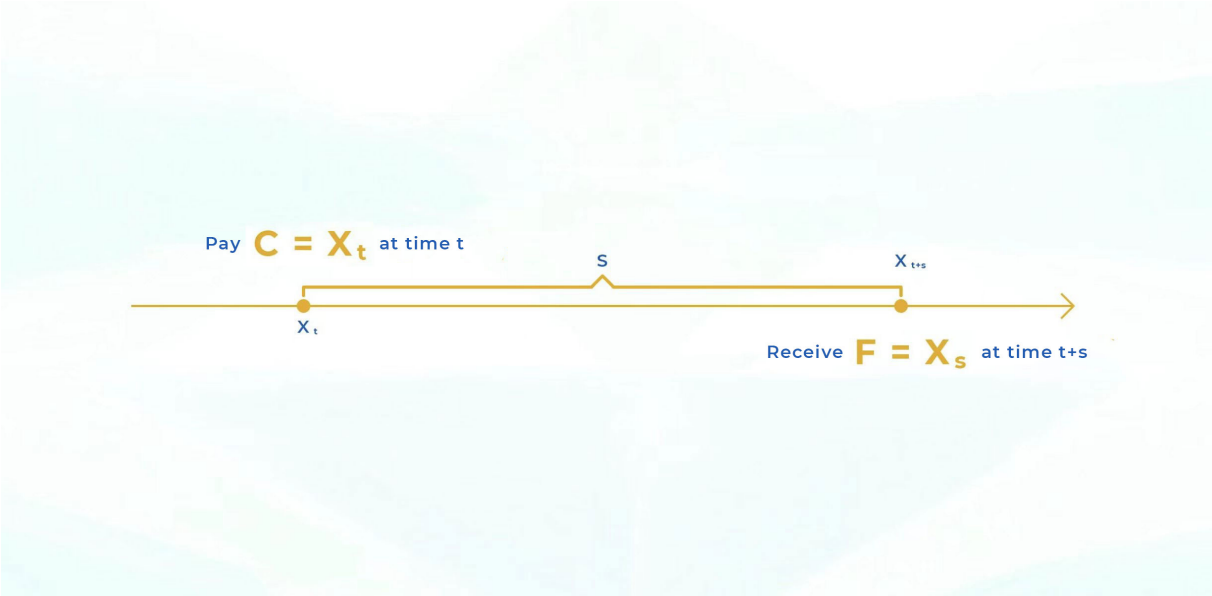


Figure 2: Martingale trades base on alternated value unit

A peer-to-peer exchange between two traders is the most general martingale transaction. This type of exchange, which requires matching, is more efficiently carried out by various intermediaries in the traditional market network. We do not intend to introduce any intermediaries, but rather to provide an unlimited supply of sellers to all traders. The benefit of this is that traders save money on the cost of original matching. The decentralized network we created in this manner has the following characteristics:

1. Every transaction is a martingale transaction: $C = Xt = \mathbb{E}(X_{t+s}|Xt) = F$, where C is the cost of the martingale transaction, and F is the return.

2. The target of the transaction is the digital asset on the chain, which in this article refers to the NEST token developed based on ERC20. For example, pay X_t NEST at time t, and get X_{t+s} NEST at time $t + s$.

3. All traders trade directly with an infinite supply seller, which is the NEST contract itself. This is what we call the infinite liquidity model - ILM.

4. The digital assets payed by the users to enter the contract will be burned, and the underlining assets for settlement will be issued by the contract.

5. In order to bring the entire network into a state of convergence, we allow the martingales to be the supermartingales: $X_t \geq= \mathbb{E}(X_{t+s}|X_t)$. Simply put, the current cost of payment is greater than the expected value of future returns.

Overall, the martingale network differs from the market network in the following ways:

1. Unlimited supply: As long as you have NEST, you won't have to worry about a shortage of market liquidity making trading difficult. Any transaction based on martingale information flow that traders require can be fulfilled, so supply is unaffected by the size of the counterparty limit.

2. Risk sharing: All NEST holders will bear the risks and rewards of NEST supply increase and reduction, which are features of blockchain and distributed networks. Risk management in the traditional market network is mostly reliant on market makers to hedge and transfer risks to the market. The expense of such hedging is usually excessive.

These distinctions from traditional markets will usher in some novel concepts and phenomena. As a social experiment, we anticipate its value to be as innovative and impactful as BTC/ETH.

4 NEST, NEST tokens and P assets

According to the preceding discussion, anyone can pay X_t NESTs at time t and then receive X_{t+1} NESTs at time $t+s$ in the decentralized martingale value network NEST, with NEST as the value unit and the NEST contract as the universal transaction party, as long as there is a martingale information flow X_t $t \geq 0$. The NEST system is built on public chains like Ethereum, and it is a completely decentralized protocol based on smart contracts. The following is the procedure for implementing it:

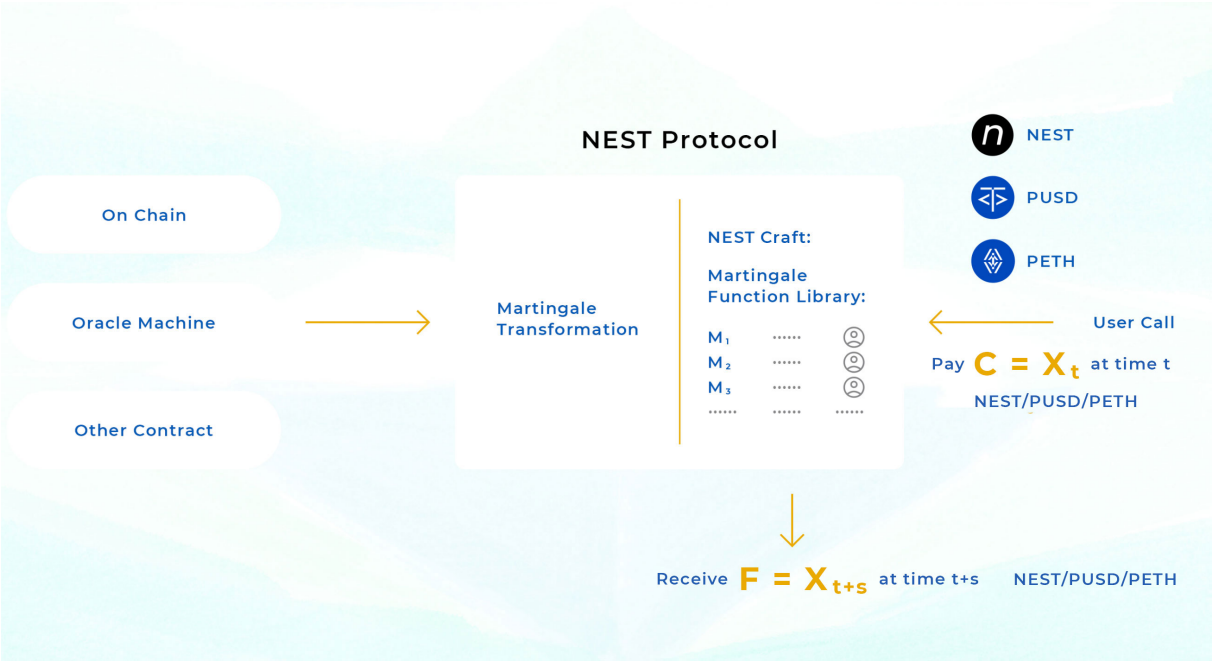


Figure 3: The FlowChart of NEST

In the flowchart above, any given martingalized information flow can be used for martingale trades. Each participant completes his or her own trade

regardless of the overall system's security or stability. Initially, all NEST tokens are generated by oracle mining. Their distribution will gradually become decentralized as they circulate in the market.

In the case of a large amount of martingale information flows, a more natural idea is to linearly combine different martingale information flows, i.e. martingale transactions, in order to derive more applications. This design, which is similar to the Ethereum virtual machine, will greatly expand the scope of NEST's application, transforming it into a chain infrastructure that will allow anyone to create more NEST-based applications. We refer to it as the martingale function library or NESTcraft.

Its schematic diagram is as follows:

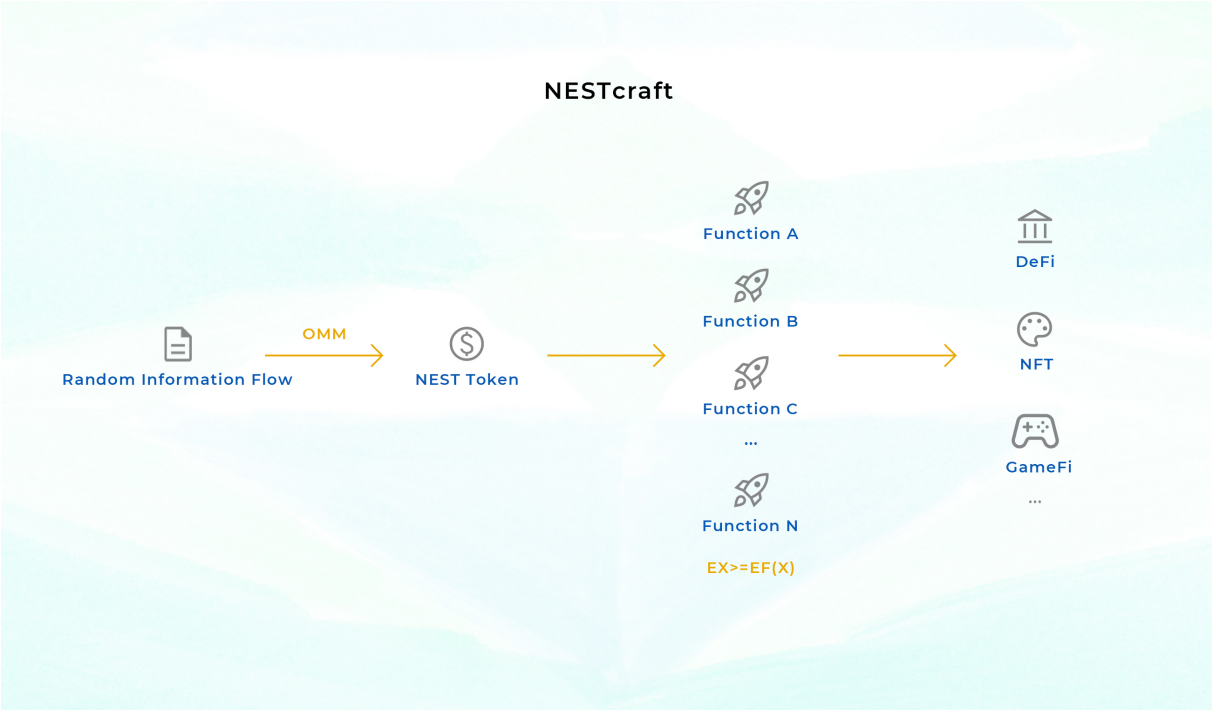


Figure 4: NEST Function Library

When NEST obtains more liquidity, we can use the NEST/USD oracle to convert the transaction's underlying assets from X NESTs to X USD of NESTs, which will meet the needs of many attempts to establish hedging positions based on the fiat currency standard.

A more impactful idea is to introduce, in addition to NEST as the original value unit of the martingale network, PUSD, PETH, PBTC, and other equivalent assets of USD/ETH/BTC as the value unit of martingale exchange, making the entire network application more extensive. However,

the value stability of the aforementioned equivalent assets must be considered, and some liquidity position arrangements must be made in advance. Overall, NEST will provide a more decentralized and defensible stablecoin or comparable asset (equivalent to some off-chain assets).

The schematic diagram is as follows:

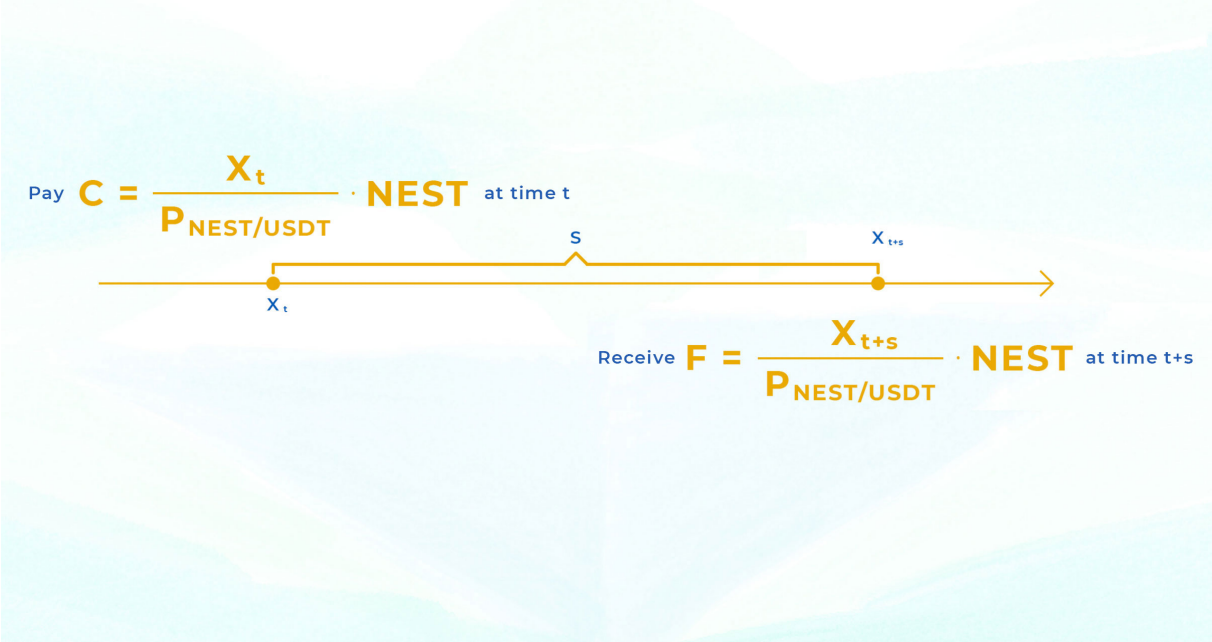


Figure 5: Martingale Tradings

5 Martingale Information Flow

The martingale information flow on the chain is mostly derived from the random data source of the public chain itself, the price information flow provided by the decentralized oracle, or other random information flows. Of course, any definite information flow is a martingale, with only X_t and X_{t+s} being equal.

Given a random information flow with a certain distribution, we can perform some functional transformations to obtain the desired martingale information flow, and this process is known as martingale transformation. In theory, there are a large number of feasible martingale transformations, all of which can be implemented in smart contracts for a given computational complexity. However, it is difficult to obtain a random variable with a stable distribution as we discussed previously. Thus, we try to design a supermartingale model with system convergence (deflation convergence,

the number decreases with time). Even if some distributions are not stable but still easy to estimate its upper and lower bounds of random information sources, we can get a series of super martingale information flow, which can be used for the trading design conditional on that the buyers are willing to accept the supermartingale setting.

The security of random information sources or random variable sources like HASH in the Ethereum POW era is dependent on the distribution of miners and the corresponding incentive design. However, in the ETH2.0 era, we cannot simply use this type of random information source for martingale transformation. Some randomness sources from the beacon chain can be used under controlled conditions (such as size and range), but they must first be converted into martingales.

A type of martingale information flow exists as well, primarily the effective price of some underlying assets. This type of data flow must be obtained via the Oracle machine. Under the assumption that the price is a geometric Brownian motion (GBM), we can design the price information flow as a kind of martingale, which is very convenient to deal with. However, because the actual price information does not fully conform to the GBM assumption, it is necessary to consider appropriately relaxing the parameter value and adjusting the price information flow to an upper martingale. This is a relatively simple implementation that does not rely entirely on rigid assumptions.

It can indeed provide a large category of martingale information flow based on the Oracle machine, but the decentralized design of the Oracle machine is more difficult. The majority of Oracle machines on the market are currently centralized, and there is a risk of node or organization centralization. We created NEST Oracle, a decentralized oracle based on game theory, for this purpose.

We can also obtain some random information flow from other sources on the chain, such as smart contract agreements, and convert it into a martingale. This means that our martingale information flow will be highly scalable and open.

The schematic diagram is as follows:

Figure 6: Martingale Transform

6 NEST Oracle

The NEST oracle is the market's only truly decentralized oracle given a price flow outside the chain. It is the design of a decentralized game so that the game can output a price flow in equilibrium and ensure the price flow is consistent with the price flow outside the chain. The price flow deviation is as small as possible. The NEST oracle solves this problem by using quotation mining, two-way options, a verification cycle, a price chain, and β coefficients. The price series provided by the NEST does not change the distribution of asset prices, but it is close to a discrete sampling model, which is determined by the structure of the decentralized game. The quotation deviation and density are determined by the depth of the arbitrage market and the price of the NEST token. Overall, NEST is a powerful decentralized oracle that preserves the fundamental nature of prices.

The NEST oracle is a completely open game network that can theoretically provide all price information flow, but for the sake of overall network security, the price information flow used for the martingale function will still be limited to a small number of markets that are more effectively

decentralized assets, such as BTC, ETH, and so on. Please see the appendices for the NEST oracle's specific implementation mechanism and performance characteristics.

7 Martingale Functions and NESTcraft

A given random information flow can be transformed by various functions to obtain a series of martingales, which can be used for the NEST martingale transactions. We call these functions for martingalizing random information flows martingale functions. If we relax the martingale standard and choose to accept more supermartingales, the scope of application of the trading will become larger, but unless these martingales are insensitive to traders, it is easy to reduce demand and reduce the enthusiasm for participation. Of course, this only adds a variety of options for traders. As long as someone accepts this design, it will provide more supply on the original basis.

Considering the resource constraints of smart contracts, we will selectively determine some basic transformation function clusters. Generally speaking, polynomial functions, exponential functions, logarithmic functions, and most value functions are often used in reality. We design basic function clusters based on these more common functions. Each basic function corresponds to a martingale cost, which needs to be paid when calling the function.

The formulas are as follows:

- $m_1 = ax + c$:

  The cost: $C = aS_0 + c$, The settlement value: $F = aS_t \exp(-\mu t) + c$.

- $m_2 = ax^2 + c$:

  The cost: $C = a{S_0}^2 + c$, The settlement value: $F = a{S_t}^2 \exp(-2\mu t - \sigma^2 t) + c$.

- $m_3 = ax^{-1} + c$:

  The cost: $C = a{S_0}^{-1} + c$, The settlement value: $F = a{S_t}^{-1} \exp(\mu t - \sigma^2 t) + c$.

- $m_4 = ax^{\frac{1}{2}} + c$:

The cost: $C = aS_0{}^{\frac{1}{2}} + c$, The settlement value: $F = aS_t{}^{\frac{1}{2}} \exp(-\frac{1}{2}\mu t + \frac{1}{8}\sigma^2 t) + c$.

- $m_5 = alnx + c$:

  The cost: $C = alnS_0 + c$, The settlement value: $F = a(lnS_t - \mu t + \frac{1}{2}\sigma^2 t) + c$.

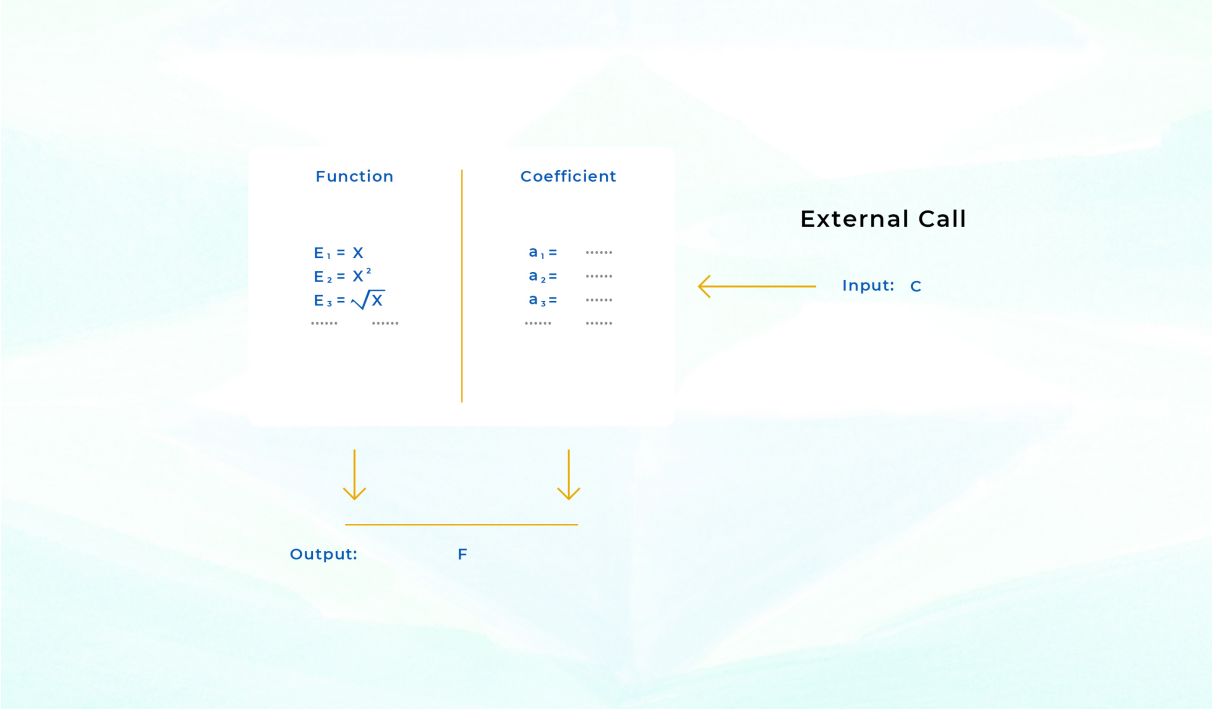
Of course, we can also combine these martingales linearly to output more complex martingales:

$$F = M\Lambda^T + \lambda_0 = \lambda_0 + \lambda_1 m_1 + \lambda_2 m_2 + \lambda_3 m_3 + \lambda_4 m_4 + \lambda_5 m_5$$

where $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}^T$.

We call this system the martingale library, or NESTcraft. If the community is willing, it can provide a front-end page for NESTcraft and link to EVM. Only the necessary function combination needs to be written, and the contract can be generated with a simple interaction, and then the martingale trading of the expression can be realized continuously.

The schematic diagram of NESTcraft is as follows:



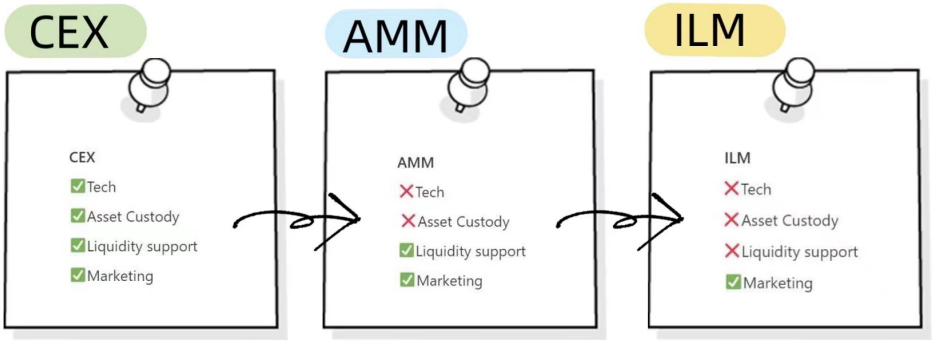
NESTcraft can continue to expand the basic function library according to the needs of the on chain world, thereby improving the application range of the NEST. This is a basic feature of the NEST's greater scalability.

8 Applications

This network offers a large number of application possibilities, we briefly list some below:

1. Decentralized exchange. We all know that derivatives exchanges need to solve technical support, asset custody (risk management), liquidity support and marketing. If you use the NEST protocol, you only need to do a front-end page to develop a decentralized exchange, so it saves the first three costs, you only need to focus on marketing.

   Schematic diagram:



2. Financial derivatives supermarket. Based on the NESTcraft, a large number of new financial derivatives can be designed, such as barrier options, Asian options, two-way options, and a type of interesting income such as square-root return, squared return, and exponential return can also be easily designed.

3. On-chain and off-chain risk hedging. Since there is no supply side for off-chain hedging in many cases, NEST can provide a large number of hedging transactions for many off-chain tradings without being affected by market makers. And the transaction on the chain can also realize one-click hedging based on NEST. The most typical one is the LP one-click hedging of UNISWAP. One-click hedging.

   Let the LP of Uniswap offers the token pairs (U, E) with volume: (x_0, y_o). Then the price is $S_0 = \frac{x_0}{y_0}$, the parameter of AMM is $k = x_0 y_0$.

Assume the price parameter of (U, E) is (μ, σ^2), T is the time of hedge (the time unit is year).

At time t_0 the user pays $x_0(e^{\mu T} - 2e^{\frac{\mu T}{2} - \frac{\sigma^2 T}{8}} + 1)$ of NEST. Then at time t, $t \in [0, T]$, the settlement returns $\sqrt{k}(S_t/\sqrt{S-0} + \sqrt{S_0} - 2\sqrt{S_t})$ of NEST. If the contract is settled after time T, it returns $\sqrt{k}(S_T/\sqrt{S-0} + \sqrt{S_0} - 2\sqrt{S_T})$ of NEST.

4. The economic framework of Metaverse and GameFi. Since the NEST provides a series of martingale functions, such fair games around deterministic mathematical relations, probability relations, and random processes can be developed by calling the NEST functions, so as to realize the unified value measurement of different games. Even if a game developer stop developing, its core value is still preserved in NEST, and can be integrated and exchanged across other NEST-based games.

5. Lottery, prop synthesis, etc. Some basic designs based on randomness can be realized with only some distribution functions.

6. Other unique applications. An open network will always bring more possibilities. Many valuable applications will be proposed more creatively when the basic functions exist. We expect that there will be many new applications and new things beyond the description in this article.

NEST's application space is extended on top of the ETH: ETH can handle general deterministic functions, while NEST's mechanism handles random functions. Because the ETH's token issuance adopts a deterministic algorithm and is independent of on-chain applications, its feedback on application-side information is insufficient. On the other hand the NEST's token issuance (increased issuance) follows the scenario, which is closer to guarantee market-clearing currency target at any time.

9 Equilibrium and Price

Here we mainly use the NEST token to discuss the balance of the system. In the beginning, all the NEST tokens were generated through oracle mining. Later, after the NEST circulated to the market, it gradually moved towards dispersion and decentralization. When more and more people participate

in the NEST network, the number of transactions increases and the trading value increases. As a result, the trading results gradually conform to the law of large numbers. The expected supply of the NEST will become smaller and smaller under the action of the supermartingale, while its demand will continue to expand, thus forming an internal mechanism of continuous price rise. We can see from the figure below that the supply of the NEST will fluctuate around a curve that converges downward. We call this fluctuation the second-order moment feature. As mentioned in the risk management below, we can use some second-order moment management to reduce the amplitude of fluctuations.

The schematic diagrams follows:

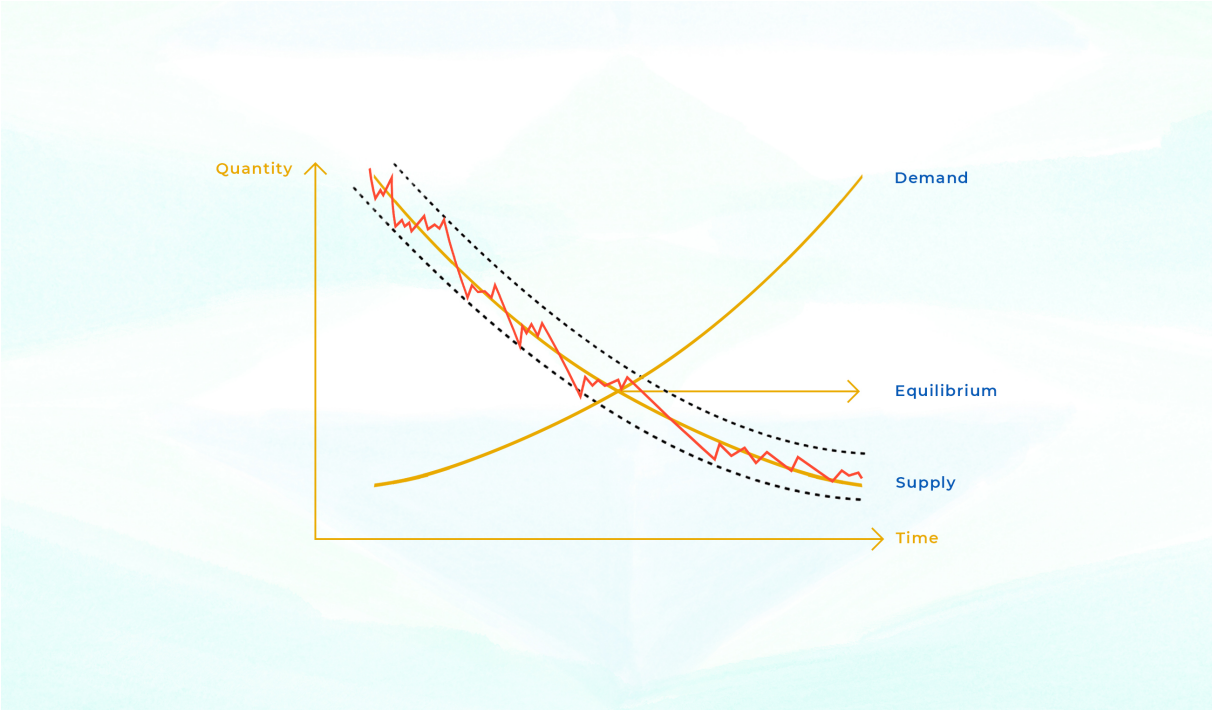


Figure 7: The Equilibrium between Supply and Demand

10 Risk Management and Time Value

For the NEST network, although its total supply can be continuously reduced through some supermartingale designs, there will always be a short-term fluctuation, such as a small probability event, market failure, etc., which will cause short-term supply fluctuations around the expected value. In order to reduce the magnitude of fluctuations, especially in response to

extreme changes, we can perform second-order moment management. The general ideas are:

1. Dynamic Parameter Setting for Martingale
2. Cutoff Management
3. Scale Control
4. Design automatic "price fix" algorithms for some applications

We can also design a time value in the NEST network to control the overall development of the network and create a source of income on the chain. This type of design is simple. For example, if the system guarantees users a certain return on staking NEST each year, a competitive NEST interest rate can be generated to assess users' willingness to hold the NEST tokens. This design is extremely flexible, with variables such as scale, rate of return, and time period being completely customizable. More financial applications can be derived once the NEST tokens has been sufficiently dispersed.

11 Advantages and Room for Improvement

In the previous discussion we have explained how the NEST as a decentralized (super) martingale network, provides a new paradigm different from market networks. This paradigm has the following advantages:

1. As a decentralized value network, it provides a transaction and risk management paradigm different from the market network, which can deal with risky assets or uncertain return transactions more concisely and at a low cost, including the ability to provide almost unlimited liquidity, no need for matching, no information search cost.
2. As a new generation of DeFi infrastructure, there is no need to provide TVL, which will not cause a waste of resources, there is no corresponding transaction slippage, and the cross-chain cost is lower (no need to copy TVL).
3. As an open and programmable network, it can provide services for more application scenarios and has great scalability.

4. Its supply has been deflationary for a long time due to the supermartingale setting, so it has an inherent price increase momentum.

5. When the network participants' currency holdings are sufficiently dispersed, more and more transactions will achieve equilibrium, forming a perfect currency form that almost guarantees market clearing at all times.

Of course, the NEST network also has many areas for continuous improvement, such as ingesting more information flows, improving the security of the game network, more flexible second-order moment management, and lower-cost martingale function clusters, etc. In terms of performance, the NEST's on-chain efficiency has room for improvement.

12 Summary

Based on the concepts of martingale trading and martingale networks, we designed and released the NEST protocol. The NEST protocol includes three main modules: NEST Oracle, NEST Assets, and NESTcraft. The NEST oracle provides the on-chain world with prices obtained through completely decentralized games. NEST assets are generated through information capitalization, providing rewards for the bidders in the oracle module, and providing currency units for martingale tradings on the NEST. Its internal cost mechanism ensures that the supply of the system is converged, and it has an internal price increase logic, and the benefits and risks of its value are shared by all holders. NESTcraft converts various on-chain random sources into a rich martingale function library, solves the problem of insufficient liquidity for investors through the ILM (Infinite Liquidity Maker) mechanism, and provides a variety of customizable martingale trading options. It provides basic conditions for the establishment of a martingale trading network.

The NEST protocol can be used for a variety of purposes, including decentralized contract exchanges, financial derivatives supermarkets, on-chain and off-chain risk hedging, Metaverse, GameFi's economic framework, lottery, item synthesis and a few others. The NEST protocol solves the trading problems of random assets and income that traditional Pareto trading cannot solve, and it greatly improves the efficiency of on-chain

tradings and reduces costs by utilizing the characteristics of blockchain technology.

Appendix A NEST Oracles

1 Introduction: The Challenge of Price Oracles

Price oracles commonly used in the DeFi industry generally reflect the asset price of centralized exchanges by "trusted" nodes, where the price is "uploaded" to the chain for usage by DeFi protocols. There is a basic problem with verifying such price data. Some DeFi projects utilize price data gathered from decentralized exchanges, however, because transaction volume is minimal, the pricing data is readily manipulated and vulnerable to attack. This creates a clear market need for an Oracle solution that directly checks the pricing to ensure the information is correct and timely but is also prohibitively expensive to attack. This system should also be decentralized to reduce the risks of centralization.

Oracle price data must meet the following key requirements:

- Accuracy: The price data on the oracle should truly reflect the market price.
- Price sensitivity: The price data on the oracle should react fast enough to market movements.
- Attack resistance: The cost of distorting or affecting the real price is extremely high for any attackers.
- Direct verification: The verifier can be any third party, and no centralized review or threshold is required.
- Distributed quotation system: no centralized review or threshold is required, and anyone can freely join or leave at any place and at any time.

2 NEST Solution

NEST provides a creative solution, including collateral asset quotation, arbitrage verification, price chain, beta coefficients, and other modules to form a complete NEST protocol. Taking the Ethereum network as an example, the schematic dia-

gram of the NEST protocol is described in Figure 1 below and we will discuss the details in the following subsections.

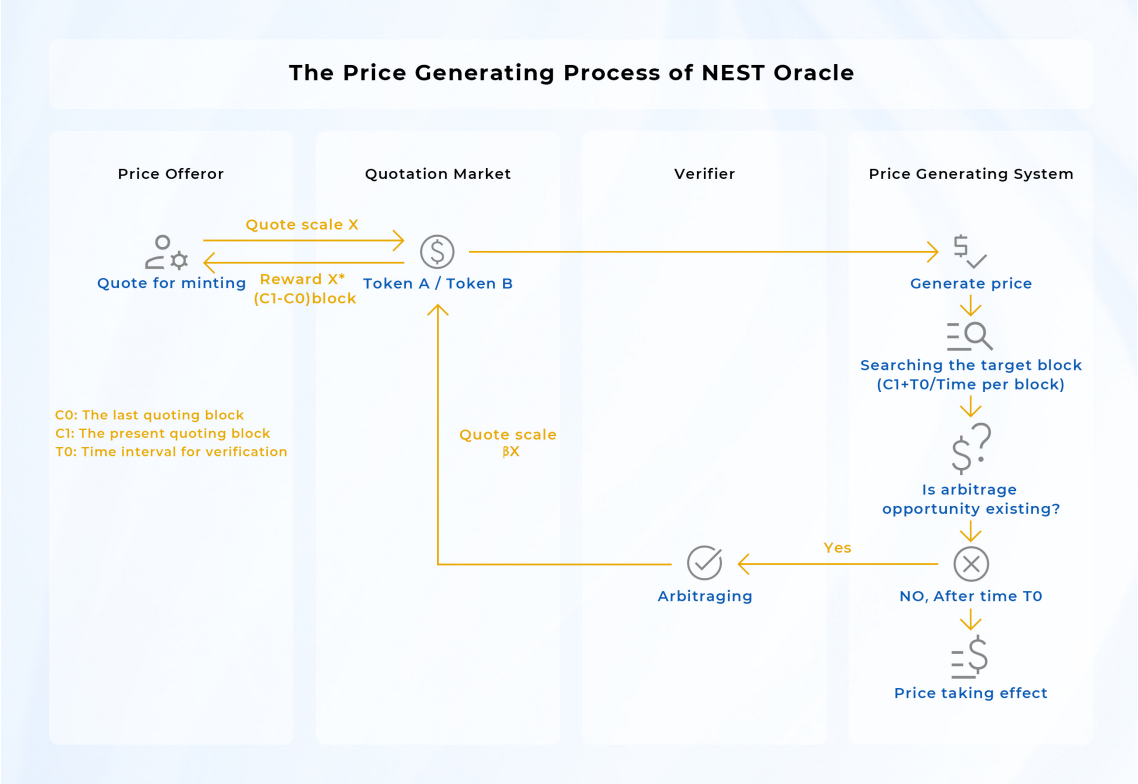


Figure 1: Diagram of NEST Protocol

2.1 Price Model of NEST Oracle

NEST oracle is the only truly decentralized oracle on the market today: given an off-chain price stream, how to design a decentralized game such that the game equilibrium can output a price stream with the smallest possible deviation from the off-chain price stream. NEST oracle solves this problem with quotation mining, two-way options, validation cycles, price chains and β factors. NEST provides a price sequence that does not change the distribution of asset prices but approaches a discrete sampling model, which is determined by the structure of the decentralized game, where the quote deviation and quotation density depend on the depth of the

arbitrage market and the price of the NEST token. Overall, NEST provides an efficient decentralized oracle that maintains the fundamental traits of asset prices. In practice, we tend to use highly efficient market prices, and hence choose the most liquid underlying assets such as BTC and ETH, etc.

The basic price model follows the Geometric Brownian Motion (GBM) model. Considering the characteristics of prices deviation and discrete time, we correct the prices using the k-factor as follows,

$$k = \max(\frac{|p_2 - p_1|}{p_1}, 0.002) + \sqrt{t} \cdot \max(\sigma, \sigma_0) \tag{1}$$

where p_2 and p_1 represent the current and previous prices respectively, t, measured by second, represents the difference between the time transaction happens and the time p_2 becomes effective. Furthermore, σ the instantaneous volatility follows

$$\sigma = \frac{|p_2 - p_1|}{p_1\sqrt{T}}$$

where T represents the time-lapse between p_1 and p_2 becoming effective. σ_0 denotes the regular volatility, set by the protocol (generally different values for different financial products).

The correct procedure follows

- when it comes to a call option, the long price is $(1 + k)p$ while the short price is $\frac{p}{1+k}$
- when it comes to a put option, the long price is $\frac{p}{1+k}$ while the short price is $(1 + k)p$

where p represents the base price.

Since price is verified on-chain, NEST has provided an open and transparent ecosystem for everyone. One of the most important points is openness: anyone can start a price information flow and motivate price providers to mint any kind of token. For example, a project can set up the price pair of its own token to USDT, and motivate others to provide price information by rewarding them with this token. This would help any project to expand the number of minters in its ecosystem.

2.2 Roles of NEST Protocol Actors

Participants in the NEST protocol are as below:

- Price Makers: The participants who submit price quotations to the protocol. This includes miners who quote prices for mining and verifiers who complete the transaction and quotation.
  - Miners: Providing quotations to receive NEST (ERC-20 Token). Miners are denoted as O, and anyone can become a miner.
  - Verifiers: If the quotation price deviates from the market price, the verifier can trade a quoted asset at the quoted price to earn revenue. The verifier needs to "force" a quotation at the time of the transaction and does not need to pay a commission nor participate in mining. Verifiers are denoted as A, and anyone can become a verifier.
- Price Callers: The contract or account that "calls" the NEST protocol quotations and pays the fee is called a price caller. Price callers are denoted as C. Any contract or account can become a price caller, but this will generally be reserved for other DeFi protocols and institutions.

2.3 Quotation Mining and Price Verification

One can easily start a quotation channel via NEST protocol where he/she needs to set the quotation pairs (one channel allows multiple pairs), quotation scale, commission fee, the token and scale of the collateral, etc.

Taking ETH/USDT as an example, miner O intends to quote a price of 1 ETH = 100 USDT. At this time, miner O needs to input the collateral NEST and the quoted assets, ETH and USDT, into the quoted contract. The scale is x ETH and $100x$ USDT, and the paid commission is λx ETH. Miners participate in mining based on a commission scale to earn NEST. The whole process is completely open and transparent, that is, anyone can assume the role of O, and the price and scale are set independently.

After miner O submits the collateral, assets and price to the quoted contract, verifier A believes that the price presents an arbitrage opportunity, and can trade either ETH or USDT at the quote from miner O, which is 1 ETH = 100 USDT. This mechanism ensures that the maker's price is either the fair price in the market or the equivalent price of the two assets recognized by himself/herself. In the view of miner O, 1 ETH and 100 USDT are equivalent, so it does not matter which asset the verifier trades. This process is the price verification period.

Essentially, miners, through quoting, also provide either bullish or bearish two-way options during the verification period, with the strike price as its quoted price. Verifiers, then, execute this option if they find that there is an arbitrage opportunity. Therefore, if miners want to minimize their costs, they need to report the price that is least likely to be traded during the verification period. This allows the miner's quotation has a certain ability to forecast future prices. For the verifier, whether they choose to arbitrage (execute) depends on the difference between the quote and market price. We call the minimum difference the verifier will take action on the "minimum arbitrage space"; this value also depends on the length of the verification period and the transaction cost.

The formula for quote mining is expressed by the following formula: Maker O quotes p, that is, 1 ETH = p USDT, the asset scale is x ETH, so the corresponding USDT quantity = $x \cdot p$. The commission scale for participating in mining is $w = \lambda \cdot p$, and verifier A can use the price p to trade $x \cdot p$ USDT for x ETH.

2.4 Price Verification Period

Opened quotes have an allotted period of time attached, denoted as T_0. This time determines the period of risk the maker takes and the price sensitivity. After the verification period, quotations that have not been traded are called "effective quotations" which includes two variables - price and quotation scale (p, x). Effective quotations form the block price mentioned in section 2.6. However, the price quoted that is already traded by the verifier will not be adopted. If a certain quoted price

is partially traded, the remaining part is also an effective quote, i.e. (p, x'). After the price verification period is complete, the maker's remaining assets will be made available to withdraw at any time.

The verification cycle affects miners, quotation costs, and price accuracy. The longer the time, the higher the option cost, and the more difficult it is to predict the future price. Judging by current DeFi market demands for price data and the volatility of mainstream assets, a reasonably set T_0 is between 5 to 10 minutes (pending adjustments and optimization based on the ETH network capacity and verifiers, scale, with the optimal time being within 1 minute). Note that if a price has passed the verification cycle, it indicates that there is no arbitrage space between this price and the current market equilibrium price (the minimum arbitrage space is determined by T_0 and transaction costs), thus representing the approximate current price; the existence of T_0 does not mean a delay in prices.

2.5 Price Chain

According to the above agreement, the verifier needs to force a new price after accepting the transaction of a maker. To put it simply, the verifier needs to offer a new price to close the opening left by the rejected price. For example, verifier A_1 and maker O accept the transaction with the price of p_0 (the maker O's quotation scale is x_0 with the collateral scale of y_0), so A_1 needs to quote a price p_1 to the contract immediately with the asset scale of x_1, and transfer x_1 ETH and $x_1 \cdot p_1$ USDT together with the collateral y_1 to the contract. Commission and mining participation rewards are not paid at this time. If verifier A_2 accepts the transaction with A_1, A_2 needs to quote the price p_2 with the asset scale of x_2 and the collateral scale of y_2. A continuous price chain with T_0 as the maximum quotation interval is formed:

$$p_0 \to p_1 \to p_2 \cdots$$

the quoted asset chain is

$$x_0 \to x_1 \to x_2 \cdots$$

and the collateral asset chain is

$$y_0 \to y_1 \to y_2 \cdots .$$

2.6 Block Price

The NEST Oracle determined price is recorded on the blockchain, with each block recording a price. The effective price in the block is generated by a certain algorithm. The price is called the block price or NEST-Price. Assuming the effective quotation of a block is (p_1, x_1), (p_2, x_2), $(p_3, x_3) \cdots$ the block price is

$$P = \frac{\sum_{i=1}^{M} p_i \cdot x_i}{\sum_{i=1}^{M} x_i}$$

where M represents the number of effective quotations in this block. If there are no effective quotations in a current block, the price of the most recent block will be used.

2.7 Price Sequence and Volatility

Each block of the Ethereum network corresponds to a price on NEST, thereby forming a price sequence. The price sequence has important functions, including:

- Provide an average price for DeFi operations, including the arithmetic average price of N consecutive blocks $j = 1, \cdots, N$

  $$P_s = \frac{\sum_{j=1}^{N} P_j}{N} ,$$

  or the weighted average price of N consecutive blocks:

  $$P_m = \frac{\sum_{j=1}^{N} P_j \cdot Y_j}{\sum_{j=1}^{N} Y_j}$$

  where $Y_j = \sum_{i=1}^{M_j} x_{ij}$ represents the total asset scale of all effective quotations in block j and M_j the number of effective quotations in block j.

- Provide volatility indicators for most DeFi derivatives, such as rolling volatility of 50 consecutive quotes, or various other volatility indicators customized for DeFi purposes.

- Other statistics.

2.8 Attack-Resistant Algorithm

If the scale of DeFi assets calling the NEST-Price is very large, there is a huge opportunity for attacks. An attacker may tamper with a normal quote, p_0, and changed it to p_1, or the attacker may trade maliciously, hoping that the price will not be updated (as prices cannot be adopted and updated once the price has been traded). With attackers willing to sacrifice the price difference between P_1 and P_0 in exchange for greater profits, the price-setting mechanism becomes invalid. So how does NEST prevent these kinds of attacks?

By increasing the cost for attackers. First, the price chain itself is an attack-resistant mechanism: attackers must offer an alternative price and the corresponding assets at this price after attacking the price. After the attack, attackers must either offer the same "correct" price or leave an arbitrage opportunity. There must be a verifier in the market to recognize the arbitrage opportunity and revise the quote.

Secondly, in order to amplify the cost to the attacker, we arrange every verifier's quotation asset scale as follows: the scale of the verifier's transaction is x_1, and the scale of the simultaneous quotation is $x_2 = \beta x_1$ with $\beta > 1$. Therefore, the verifier must quote at a price more than double the scale of the quotation. Notice that we only allow this amplification for quotation asset up to 4-round verification. On the other hand, we also enlarge the collateral asset in the same way but without 4-round limitation. As an example of $\beta = 2$, the quoted asset chain and the collateral asset chain in section 2.5 follow as

$$x_0 \to \beta x_0 \to \beta^2 x_0 \to \beta^3 x_0 \to \beta^4 x_0 \to \beta^4 x_0 \to \cdots \to \beta^4 x_0 \to \cdots$$

and

$$y_0 \to \beta y_0 \to \beta^2 y_0 \to \beta^3 y_0 \to \beta^4 y_0 \to \beta^5 y_0 \to \cdots \to \beta^n y_0 \to \cdots$$

respectively.

Attackers either offer huge arbitrage opportunities to the market (the scale increases by levels, making this kind of attack almost ineffective) or must continue to use an extremely high volume of assets to self-deal based on the market price to delay the opportunity for price adoption. For example, assuming that the verification period is set as $T_0 = 5$ minutes, if miner O makes one quotation at present, to prohibit this quotation become the effective price in coming 1 hour, the attacker needs at least $6144y_0$ collateral asset and $32x_0$ each quoted asset. Furthermore, the attacker needs at least $12284y_0$ collateral asset and $300x_0$ each quoted asset to paralyze NEST quotation for 1 hour if the miners make the quotation every 5 minutes in the coming 1 hour. Notice that the quotation channel zero set $y_0 = 100,000$ NEST. Only focusing on the collateral asset, 1,228,400,000 NEST makes this attack plan almost impossible to fulfill considering that the total circulation of NEST is not over 3 billion. This kind of attack-resistant ability cannot be achieved by centralized exchanges.

2.9 Incentives and Economics

Miners obtain NEST Tokens through paying ETH commissions and taking certain price fluctuation risks. Verifiers earn profits directly based on the calculation of price deviation while also bearing the risk of the quoted transaction, so for the verifiers, the cost/benefit is relatively clear. For the miners, the model of quotation mining requires a corresponding economic foundation. ETH contributed by miners is denoted as X, and will be returned back to NEST holders regularly, usually on a weekly basis. This process builds an automatic distribution model, so that each NEST Token has intrinsic value, which is verifiable on-chain. Only relying on the quotation miner's ETH is not enough to complete the logical closed-loop system, which returns to the original intention of constructing the price oracle. The fact that the on-chain price is a core demand for all DeFi products means it is often regarded as the most integral part of DeFi infrastructure. DeFi developers and users should pay the corresponding fees when using NEST-Price denoted as Z. Therefore, the

value of NEST is denoted as X+Z. In general, the cost of obtaining NEST is X and NEST creates value for NEST holders throughout the whole ecosystem. The value of NEST is typically greater than the overall cost. For each miner, the cost is uncertain, so there exists a trading possibility. Under the assumption that the overall value is greater than the overall cost, NEST holders with different costs can compete with each other to achieve organic equilibrium, which is similar to the equilibrium found in the stock market. All tokens in the entire NEST ecosystem are generated by mining, and there is no reservation or pre-mining. All costs of generating NEST will be returned to NEST holders, and NEST is only used for incentives. The NEST model achieves complete decentralization, as anyone can join the system, and its characteristics are similar to that of Bitcoin. The NEST protocol upgrades the DAO method, where adjustments need to be first proposed and then approved by a 51% majority via community voting before being implemented.

2.10 The New Characteristics of Latest NEST

The most recent version of NEST is NEST 4.4. The new characteristics of NEST 4.4 compare to the early versions are:

- Improved techniques: allow price offering for multiple assets, in one smart contract, one can start the price information flow for more than ten different assets. In this way, gas fee can be saved handsomely. The efficiency of uploading information is much better.

- Improved economic models: cancel the quotation commission fee. Calling quotation price from NEST is also free now. In the meantime, the mint production is reduced to 1/6 compared to before. The circulation increases slower, slower than 3% per year. In the long run, these changes will guarantee the increasing value of NEST. The total number of NEST will not exceeding 3,000,000,000 (3 billion). The threshold of price information offering is lower, only 0.01 ETH and assets of the same value is needed to be deposited.

3 The Application of NEST-Price

Although NEST focuses on on-chain price data, it can also design price-equilibrium products including the following:

(1). Equilibrium Token: A digital asset that represents economic equilibrium formed by excess collateralization and market arbitrage mechanisms. This can also represent the equilibrium exchange relationship between prices. Equilibrium tokens can be regarded as on-chain valuation units composed of token generation contracts, arbitrage mechanisms, and feedback correction mechanisms. The important significance of equilibrium tokens is in their unique foundation, which increases or decreases following the changes of the entire public chain, such as the Ethereum blockchain. Secondly, they can be proven on chain with a risk-reward structure different from ETH.

(2). Decentralized Transactions: Traditional decentralized transactions are mainly based on peer-to-peer quotation matching. This is fundamentally flawed, as the core of modern exchanges is bilateral auctions, which have the characteristics of forced ordering and forced transactions at prices for both parties. This type of feature involves calculation characteristics, which do not match the current serial queuing mechanisms of the blockchain. A meaningful decentralized transaction would be a market-making system, that is, a two-way forced acceptance of quotations, which can be achieved perfectly with the NEST quotation mechanism.

(3). Automatic Settlement Mortgage Loan: Due to on-chain data, a loan contract that involves liquidation or automatic settlements can quote prices and automatically trigger restrictions, so that loan behavior is not limited to the options of contract structures.

(4). Futures: A distributed futures model is similar to an equilibrium token currency, but it also introduces arbitrage from any third party. This can am-

plify the transaction scale of forward transactions or directly earn revenue from transaction price fluctuations. This was impossible to design before now. All general futures require a centralized institution to perform forced liquidations, but distributed futures do not bear the risk of centralization.

(5). Volatility Products: Derivatives based on the volatility of equilibrium prices are used to hedge or smooth derivatives risks due to the on-chain equilibrium price sequence.

The above only takes the most basic products in finance as an example. Through using NEST-Price, a complete spectrum of decentralized financial products that differ from previous basic peer-to-peer transactions can be designed. Due to the introduction of global variables, the entire DeFi ecosystem is set to enter a new era. As for why DeFi needs global variables, this is because of the nature of finance and general equilibriums, rather than partial equilibriums. A simple local supply and demand relationship is insufficient; there needs to be an effective and complete pricing system based on the whole market arbitrage mechanism. This is not possible for the commodity economy, as simple peer-to-peer transactions cannot solve fundamental financial problems. However, in order not to bear the risk of centralization but also to have generally equal characteristics, global variables like "price" are needed. This variable cannot be introduced centrally, so our oracle scheme is a fundamental part of the infrastructure underpinning the entire field of decentralized finance.

4 Quotation Risk of NEST-Price

As with all financial products and services, NEST-Price is not without risk. Whilst many risks are unable to be described or recognized due to their inherently personal nature, here is a brief description of the quotation risk of NEST-Price:

(1). Due to the existence of the minimum arbitrage, there may be some risks when using NEST-Price for financial services that require extremely high price accuracy. This should be taken into account when designing.

(2). The market arbitrage mechanism is not aggressive enough, which is reflected in inadequate efforts by arbitrageurs. When there is a huge opportunity for arbitrage, no one notices it. This requires higher market acceptance and recognition as the industry develops further.

(3). Although the price cannot be attacked directly, the price mechanism can be attacked indirectly through attacks on NEST. For example, attackers can take more than 51% of the NEST tokens and then modify important parameters to invalidate the quotation mechanism. This problem can be prevented by limiting key parameters while increasing the NEST market's size, making 51% of attacks more difficult to achieve.

(4). The risk of code vulnerabilities or significant external changes. If there are vulnerabilities in the underlying Ethereum code, the NEST system code, or a significant change in the external environment, the price caller will be affected. This can be corrected through on-chain governance and contract forks.

Appendix B The Accuracy of the NEST Price

The Accuracy of the NEST Price

NEST Research Academy

September 2020

ABSTRACT

This short article develops a model to estimate the difference between the NEST price and a source price, e.g. price from an exchange. Under plausible assumptions, we show that the difference can be as small as 0.003 when volatility is small. It can even be lower if the transaction cost in the blockchain gets lower.

1. Model Setup

A *price-provider* is an individual who inputs a price into the NEST system and waits for a certain number of blocks passing to be verified by other individuals. The operation is equivalent to write an American type call and put option that anyone else can exercise it by using the input price as the exercise price. Thus, the price-provider shall minimize the value of this option by carefully choosing an input price. Precisely, the price-provider's objective problem is

$$P^* = \arg\min_P \left(\max_\tau E^Q[e^{-r\tau}|S_\tau - P|] \right), \tag{1}$$

where $\tau \leq T_0$ is a stopping time and T_0 is a fixed time horizon[1] , P is the input price decided by the price-provider. In other words, the price-provider has to minimize the value of one American type option by choosing an appropriate exercise price P. Here asset price $S_t, t \geq 0$ shall be referred to the price in an exchange at time t. Thus, the market is complete and we price the derivative in a risk-neutral framework by taking the expectation under the risk-neutral probability Q.

Denote the solution to the above problem by $P^* = P(S_0; \sigma)$, where σ is the volatility of the source price sequence S_t. Noting that the price-provider inputs a price optimally based on all of his information from a centralized market and/or from the decentralized world.

1.1 Arbitrageur

The price-provider writes an American option when he inputs a price K. It seems that anybody can exercise the option without any cost. However, the NEST requires that the one (arbitrageur) who exercises the derivative must input another price and lock in as much as β times the original asset requirement. In other words, to exercise one option, the arbitrageur

[1]For the NEST system, the time horizon T_0 actually is random because the time interval between two successive Ethereum blocks is. The framework in this note can be extended to study this case.

has to write β units of the same type of American options, where $\beta > 1$ is a specific multiplier.

One arbitrageur who wishes to make profit from the derivative can construct (sell) a portfolio in the outside market that replicates the derivative. Then the arbitrageur can make a risk-free profit the same as the value of the derivative. However, there is risk that the arbitrageur can not obtain the opportunity to exercise the derivative because it is competitive to take the arbitrage. Therefore, instead of making the risk-free profit, a realistic strategy is to make a *quick* profit in the sense of statistic arbitrage as follows.

The arbitrageur does nothing but waits until the difference between the outside asset price and the input price P is sufficiently large. Then he exercises the option and buys or sells in the exchange simultaneously to make money without any risk. Such an opportunity may not be available for all time, but in long time there are many chances. So statistically the arbitrageur can make money.

We calculate the following objective function for the arbitrageur:

$$\max_{\tau} E[(|S_\tau - P| - A)1_{|S_\tau - P| > A, \tau < T_0}], \tag{2}$$

where A represents all costs of the transaction, including Ethereum transaction fee and the value of the derivative multiplied by β. The stopping time τ in the above indicates that the arbitrageur will wait for the best time to take the arbitrage. However, considering the competitive environment, most likely, the profit is taken when the first time a target is reached. So the objective function turns to be

$$E[(|S_\eta - P| - A)1_{\eta \leq T_0}], \tag{3}$$

where $\eta = \inf\{t : |S_t - P| - A > \epsilon\}$ and ϵ is the minimum target profit of the arbitrageur. Along with the arbitrage-taking method (3), the corresponding loss (or the cost of inputing a price) of the price-provider is

$$E(|S_\eta - P|1_{\eta \leq T}).$$

The price-provider shall minimize the cost by choosing an appropriate K. That is, the objective function of the price-provider is

$$\min_P E[|S_\eta - P|1_{\eta \leq T_0}].$$

In fact, we should price it in a risk-neutral sense:

$$V^*(0) = \min_P E^Q[e^{-r\eta}|S_\eta - P|1_{\eta \leq T_0}],$$

where r is the risk-free interest rate. It yields that the price-provider can construct a portfolio in the outside market to hedge this derivative, so that his loss is a deterministic value same as V^*.

2. A Solution of the Model

Given the design of the NEST, we let

$$A = \beta V^*(\eta),$$

where $V^*(\eta)$ denotes value of the same derivative at time η. We let ϵ be the transaction fee in the blockchain (the gas fee).

Aware of the way the option is exercised, the price-provider actually considers the objective problem as follows.

$$V^*(0) = \min_P E^Q[e^{-r\eta}|S_\eta - P|1_{\eta \leq T_0}] = \min_P E^Q[e^{-r\eta}(A+\epsilon)1_{\eta \leq T_0}] = \min_P E^Q[e^{-r\eta}(\beta V^*(\eta)+\epsilon)1_{\eta \leq T_0}]. \tag{4}$$

We assume that the asset price follows a Brownian motion with drift:

$$S_t = S_0 + \mu t + \sigma Z_t,$$

where Z_t is a standard Brownian motion. Then $V^*(\cdot)$ is identical at any time. The recursive formula (4) is simplified (for a stationary solution under constant state variables μ and σ)

$$V^* = \min_P E^Q[e^{-r\eta} 1_{\eta \leq T}](\beta V^* + \epsilon). \tag{5}$$

Exploiting the density function of η, the first hitting time of Brownian motion, we can evaluate the expecation in (5) and solve for V^* and P^* numerically.

Set $\mu = r = 0$, $\epsilon = 0.003$ (the gas fee of one transaction in the Ethereum divided by 10 (ETHs)), $S_0 = 1$, we obtain the following results.

For $\sigma = 0.0001,\ 0.001,\ 0.003$ per second:

$\beta = 1.5$: $V^* = 0.0030, 0.0104, 0.0327$; probability of arbitrage: 0.0726, 0.3353, 0.3765

$\beta = 2$: $V^* = 0.0003$, 0.0092, 0.0291; probability of arbitrage= 0.0792, 0.4301, 0.4755,

$\beta = 3$: $V^* = 0.0002$, 0.0074, 0.0233; probability of arbitrage= 0.0894, 0.6064, 0.6696,

where the probability of arbitrage is defined by $E^Q[1_{\eta \leq T}]$. For all of these cases, the optimal input-price $P^* = S_0 = 1$. Since S_t is assumed to be a Brownian motion without a drift, this answer is obvious.

The sensitivity analysis regarding verification during time T, probability of arbitrage, β, volatility σ are shown in Figure 1 and 2.

2.1 Difference between NEST Price and Price of Exchange

By the preceding analysis, the difference between the NEST price and the price from an exchange is bounded by $a := \beta V^* + \epsilon$. Figure 3 indicates the upper bound can be as small as 0.003. The upper bound can be decreased if the transaction (arbitrage) cost in the blockchain becomes small. Alternatively, We may increase the asset requirement of inputing

a price to decrease the relative weight of ϵ. For example, if we increase the asset requirement to 50 ETHs, the difference bound turns to be 0.002 only.

Figures

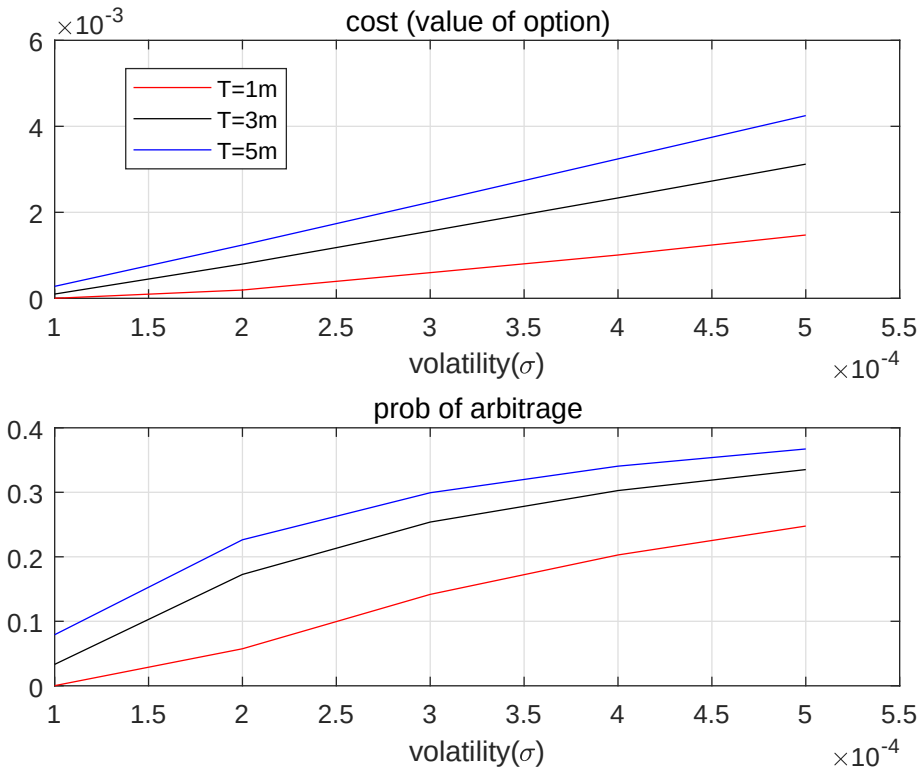


Figure 1. This figure depicts effects of volatility σ on cost of price-inputing and probability of arbitrage.

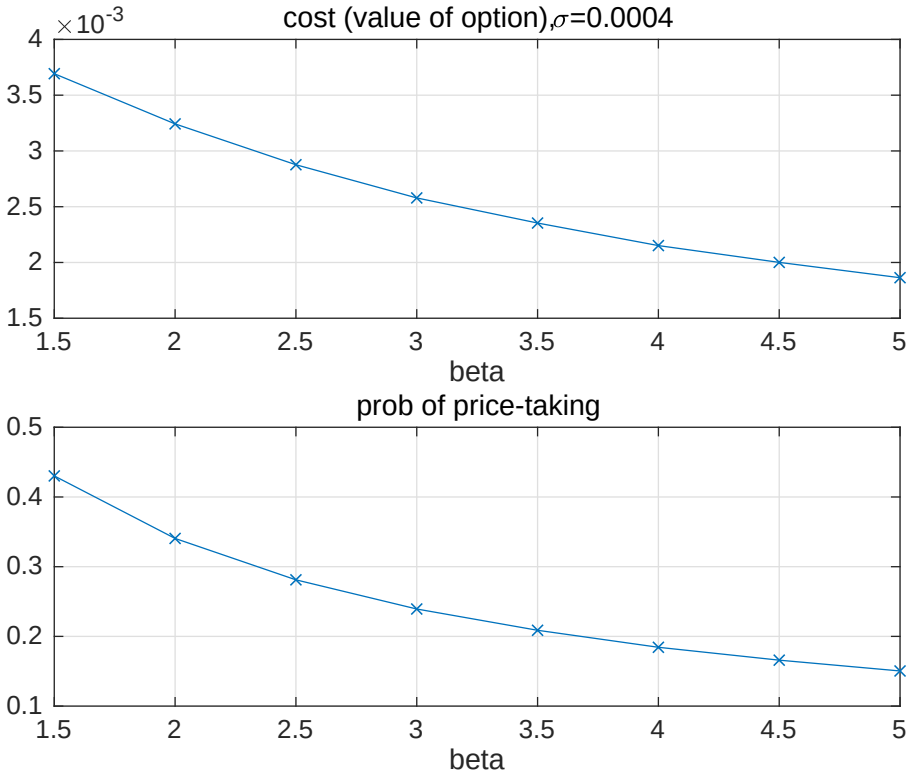


Figure 2. This figure depicts the effect of β on cost of price-inputing and probability of arbitrage.

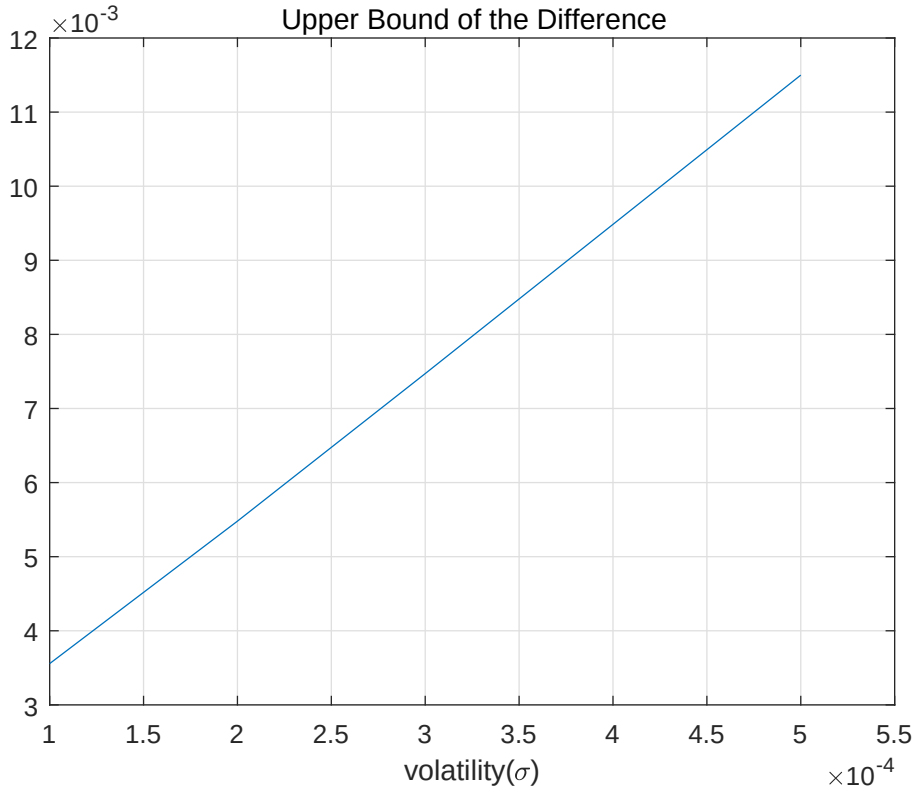


Figure 3. This figure shows the upper bound of difference between the NEST price and the price of an exchange at the same time.