DeSim: Simulating Derivatives

Ajmal Muhammad & Jesper Kristensen

Advanced Blockchain Research

February 2022

1 Introduction

We present the first part of an article series on building the first version of an Options Simulator. The motivation was to evaluate a host of derivative products regarding their return and general performance. As a specific option type, we applied the simulator to everlasting options, introduced by Paradigm.

This article first goes through the fundamental difference between perpetual futures and everlasting options to create the necessary background. Next, we cover how the options simulator was built and show its internal workings. Finally, we deep dive into everlasting options for crypto assets.

2 Mechanics of Options

An option is a derivative, a product whose value derives from underlying assets. A contract allows the buyer the right but not the obligation to buy or sell the underlying asset by a specific date (expiration date) at a specified price (strike price). There are two types of options: Call and Put. Call (Put) options give buyers the right to buy (sell) the underlying asset by the expiration date at the strike price. For this right, buyers have to pay an upfront premium to the option sellers who are obligated to sell or buy the underlying asset in case the buyers exercise their options contracts.

Futures is another class of derivatives; however, unlike options, in future contracts, both parties (buyer and seller) are obligated to buy or sell the underlying asset by the expiration date at the strike price. The contract duration for options and futures can vary; however, the expiration date is permanently fixed. Hence, buyers interested in long-term exposure to an asset have to roll out new contracts for that specific asset, which could be expensive and cumbersome.

To solve this problem, perpetual futures (*aka* future swap) was introduced in 2016 by crypto exchange BitMEX, which allows traders to speculate on the future price movements of cryptocurrencies. Unlike a typical futures contract, the essential difference is that perpetual futures do not have expiration dates, eliminating the need to constantly re-establish a buy (long) or sell (short) position. For this reason, the price of perpetual contracts anchors to the spot prices of their underlying assets. There is no need to maintain a price peg for futures since the value of the contract and the underlying asset automatically converge as the expiration date approaches. However, due to the absence of expiration dates in perpetual futures, exchanges implement a price anchoring system called the *funding rate* mechanism. This mechanism helps keep the prices of perpetual futures contracts at par with the market prices of the underlying assets through incentivization. For instance, when the price of a perpetual future (i.e., its strike price) is above the spot price of the underlying digital assets (e.g., ETH, BTC, etc.), then the funding rate (strike price - spot price) is positive. In this case, traders holding buy (long) positions would pay a small fee to those selling (shorting) the digital assets. On the other hand, a perpetual future trading (i.e., its strike price) below the spot price of its underlying asset has a negative funding rate. Therefore, those shorting perpetual futures would pay traders holding long positions.

3 Introducing DeSim

To estimate how everlasting options for different underlying assets will perform under various market conditions, we developed a discrete event-driven simulator named DeSim. DeSim is built in the `Python` programming language. Figure 1 illustrates a high level architecture diagram for DeSim, which can be divided into three main blocks, namely *daily open position*, *computations*, and *daily close positions*.

The *daily open position* comprises two modules, mainly for generating daily buyers and sellers for the everlasting options. Both modules employ a uniform distribution model for choosing the number of daily buyers and sellers, the option type (i.e., Put or Call), the contract size, and the strike price for each buyer and seller. However, the stochastic model for these parameters can easily be replaced with other models if needed. Once the daily open positions (for buyers and sellers) are created, they are added to the lists for open buyers and sellers.

The *computations* block consists of several modules that perform the calculation of daily mark price, daily funding fee, buyers and sellers transaction fee, and funding fee payment to sellers. Note that daily mark price denotes the price at which the option is trading in the market. The daily transactions fee and the amount difference between the funding fee paid from buyers and the funding fee paid to sellers are transferred to the treasury module.

Finally, the *daily close positions* block constitutes two modules that are employed for closing buyer and seller positions based on probabilistic models. The close buyers' module utilizes a 2-step approach for whether or not to close a position. The first step computes if a position is in profit or loss and then normalized the profit or loss to be in the range of 0 to 1. In the second step, the normalized value is used as a probability value in a model with binary outcomes of 0 and 1, where 0 (1) indicates keeping (closing) the position. Similarly, the close sellers' module utilizes a 2-step approach to closing a position. The first step computes if a position is in profit (due to total funding collected from buyers) or loss (due to the underlying asset movement in an unfavorable direction) and then normalized the profit or loss to be in the range of 0 to 1. Next, the normalized value is used as a probability value in a model with binary outcomes of 0 and 1, where 0 (1) indicates keeping (closing) the position.

4 Modeling Everlasting Options with DeSim

4.1 Rolling over everlasting options

The process of rolling over the position in everlasting options has a slightly different funding mechanism from rolling over a perpetual future position. In everlasting options, the longs (i.e.,

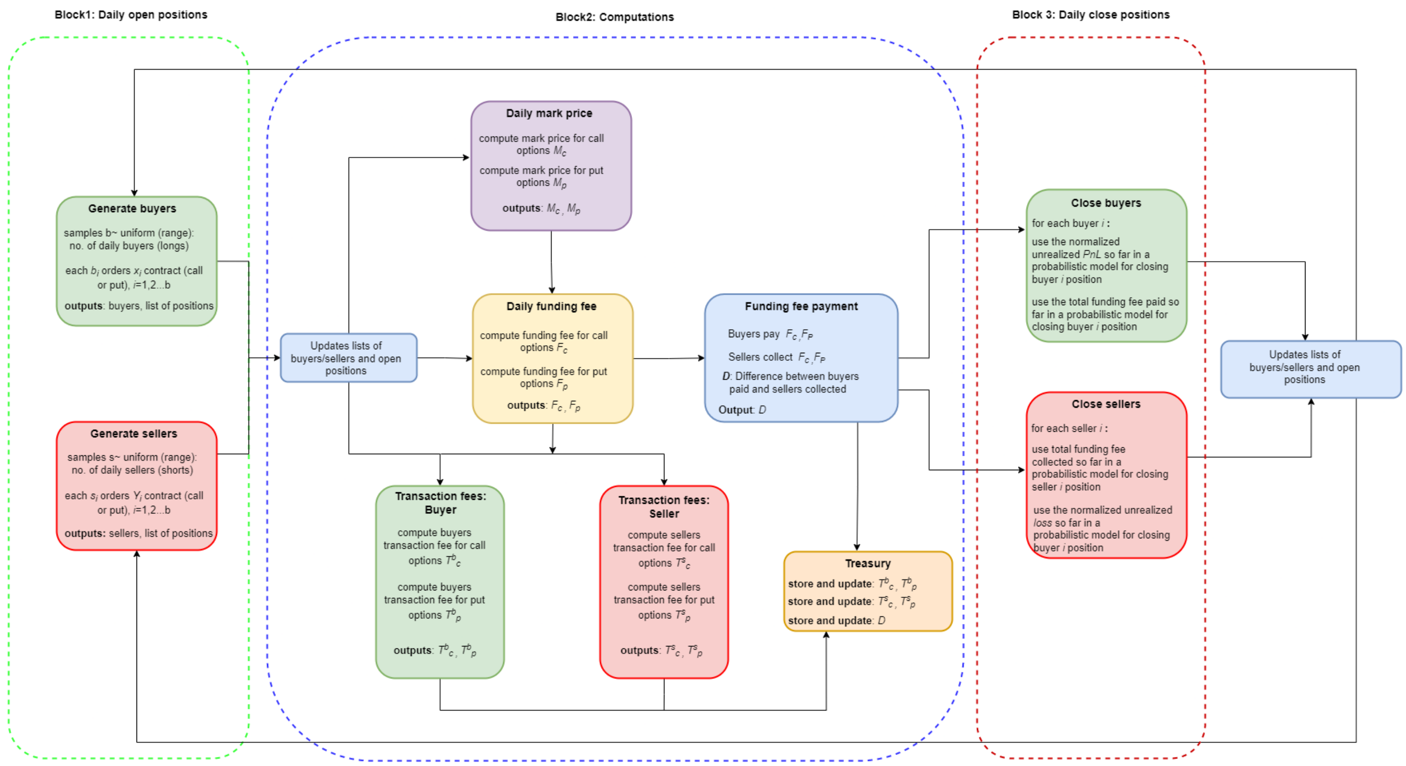

Figure 1: Architecture diagram for DeSim.

the buyers of Call and Put options) have to always pay funding fee F_{fee} equal to p_{m} - p_{off} to shorts (i.e., the sellers of Call and Put options). Here, p_{m} is the mark price or the trading price of the everlasting option just before F_{fee} is paid, p_{off} is the payoff, which is equal to $\max(0, (p_{\text{i}} - p_{\text{s}}))$ for call options and $\max(0, (p_{\text{s}} - p_{\text{i}}))$ for put options, where $p_{\text{i}}/p_{\text{s}}$ denote the spot price/strike price of the underlying asset. To summarize,

$$F_{\text{fee}} = p_{\text{m}} - p_{\text{off}}, \qquad where,$$

$$p_{\text{off}} = max\{(p_{\text{i}} - p_{\text{s}}), 0\}, \ for\ call\ option$$

$$p_{\text{off}} = max\{(p_{\text{s}} - p_{\text{i}}), 0\}, \ for\ put\ option$$

To grasp this concept, consider the \$3000 strike price (i.e., p_{s}=\$3000) everlasting ETH Put option with F_{fee} paid once per day. If ETH is trading at \$2900 in spot market (i.e., p_{i}=\$2900), the current p_{off} of the Put is \$3000 - \$2900 = \$100. If the everlasting Put is trading (i.e., p_{m}) for \$150 at time instant before F_{fee} is paid, then the longs (buyers) would have to pay the shorts (sellers) p_{m}-p_{off} = \$150 - \$100 = \$50 per day.

Similarly, if ETH is trading at \$3100 (i.e., p_{i}=\$3100) for the same strike price (i.e., p_{s}=\$3000) everlasting ETH Put option, then the Put's p_{off} is \$0. If the everlasting Put is trading for \$50 the instant before F_{fee} is paid, then the longs would have to pay the p_{m} - p_{off} = \$50 - \$0 = \$50 per day to shorts.

4.2 Modeling everlasting options for ETH

We modeled the ETH options with different strike prices using DeSim and backtested the model using spot price data for ETH-USDC of the last year. Table 1 highlights the parameters and their values used by the simulator. The model ensures that a long (short) picks a bigger (smaller) strike price value than the daily spot price while opening a new position. The model set the initial size of the treasury to 1 Million USDC. Note that the treasury will act as a counterparty for all the longs and shorts options. The daily mark price (p_{m}) is calculated using the following equation:

$$p_{\text{m}} = \begin{cases} p_{\text{i}}[1 + a(l - s)], & \text{if } l > s \\ p_{\text{i}}, & \text{otherwise} \end{cases}$$

where l and s are the total long and short positions, and a is the daily mark price coefficient. Similarly, the daily funding fee F_{fee} is computed using the following equation:

$$F_{\text{fee}} = f \cdot (p_{\text{m}} - p_{\text{i}})$$

where f is the funding fee coefficient.

The p_{m} and F_{fee} are calculated daily based on the total number of active longs (l) and shorts (s) options. Note that a high l (i.e., total active long positions) will lead to high p_{m}, which in turn increases the daily funding fee F_{fee}. The daily funding fee paid by a long position holder also depends on the contract size of the holder. For instance, a long position holder with contract size 3 will have to pay thrice the funding fee compared to the ones with contract size 1. Similar criteria

Table 1: Parameters and their values used by the simulator

Parameter	Value
Number of simulation runs (N)	500
Daily number of buyers	min:20, max:50
Daily number of sellers	min:20, max:40
List of strike prices for Call options	[2000, 2500, 3000, 3500, 4000, 4500, 5000]
List of strike prices for Put options	[1000, 1500, 2000, 2500, 3000]
Spot price	last year daily ETH-USDC spot price
Contract size	min:1, max:5
Daily mark price coefficient (a)	0.01
Daily funding fee coefficient (f)	0.01
Transaction fee	0.15%
Initial size of the liquidity pool	1 Million USDC
Cut from daily funding fee paid to shorts	2%

also applied to short position holders (i.e., an option of contract size 3 will be paid thrice than the option of contract size 1).

Based on the model described above, we perform a preliminary Monte Carlo simulation over DeSim to investigate the expected value of the profits generated over different realizations of possible scenarios. Indeed, since the model evolves randomly over time (e.g., the number of new daily buyers and sellers are random variables with a prescribed distribution), for it to lead to any reasonable and robust information one would need to (i) run multiple realizations of the model and (ii) compute the average of a given quantity of interest over these realizations.

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space and for $\omega \in \Omega$, denote by $\mathcal{G}(t;\omega)$ the random variable understood as a realization of the total daily funding fee revenue at a time $t \in [0, T]$, obtained from a single run of the model (in words, the mapping $(t, \omega) \mapsto \mathcal{G}(t;\omega)$ is akin to one whole run, with a time horizon of one year). Our goal is to compute $\mathbb{E}[\mathcal{G}(t)]$ (i.e., the expected value of the total revenue), which can be approximated by the usual Monte Carlo (MC) estimator:

$$\mathbb{E}[\mathcal{G}(t)] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \mathcal{G}^{(i)}(t),$$

where for any $i, j = 1, \ldots, N_s$, $\mathcal{G}^{(i)}(t), \mathcal{G}^{(j)}(t)$ represent two independent realizations of $\mathcal{G}(t;\omega)$.

Furthermore, denoting by S the price of the underlying asset (i.e., ETH), we aim at estimating

$\mathbb{E}\left[\frac{\partial \mathcal{G}}{\partial S}\right]$ (i.e., the expected change in total profit w.r.t changes in the underlying asset price). Naively, since for some fixed $\omega \in \Omega$, $\mathcal{G}(t) = \mathcal{G}(s(t))$, it then follows from the chain rule that $\frac{\mathrm{d}G}{\mathrm{d}t} = \frac{\partial \mathcal{G}}{\partial S}\frac{\mathrm{d}S}{\mathrm{d}t}$, and thus

$$\frac{\mathrm{d}\mathcal{G}}{\mathrm{d}t}\left(\frac{\mathrm{d}S}{\mathrm{d}t}\right)^{-1} = \frac{\partial \mathcal{G}}{\partial S},$$

where each term can be naively approximated using first-order, backward finite-differences as

$$\frac{\mathrm{d}S}{\mathrm{d}t}(t) \approx \frac{S(t) - S(t-h)}{h},$$
$$\frac{\mathrm{d}\mathcal{G}}{\mathrm{d}t}(t) \approx \frac{\mathcal{G}(t) - \mathcal{G}(t-h)}{h},$$

where $h \in \mathbb{R}$ corresponds to a time increment on the price of the underlying (typically, $h = 1$ day).

We begin with a simple array of experiments: we are interested in visualizing the mean total revenue $\mathbb{E}[\mathcal{G}(t)]$ and the mean sensitivity of $\mathcal{G}$ with respect to S, for different modeling choices for the values in Table 1. To that end, we run a Monte Carlo simulation with $N = 500$ samples for each of the following scenarios discussed below.

4.2.1 Scenario 1

We begin by fixing the range of possible daily buyers (respectively sellers) in such a way that it favors the possible number of buyers; i.e.,

$$\begin{aligned} \texttt{range\_daily\_number\_buyers} &= [20, 50], \\ \texttt{range\_daily\_number\_sellers} &= [20, 40]. \end{aligned} \tag{1}$$

Notice that the rest of the values in Table 1 are fixed for this preliminary case. We obtain the results presented in Figure 2. As we can see, from the left and right figures, by constructing a simple synthetic case for which, on average, there is a slightly larger number of buyers than sellers, our simple Monte Carlo experiment suggests that the model is favorable. Also, it seems that when the ETH index price rises, the revenue of the everlasting options drops, which could be due to the closing of long positions in profit (see Figure 3 for several active longs and shorts positions).

Furthermore, from Figure 2 (right), we can see that, on average, daily changes in the index price (ETH) do not incur significant changes in the total daily profit, with sharp variations accounting for about a $\sim 3\%$ variation in the expected daily revenue. However, this plot also evidences a non-negligible number of realizations for which the daily change in underlying price can potentially generate losses in daily revenue of about 10%.

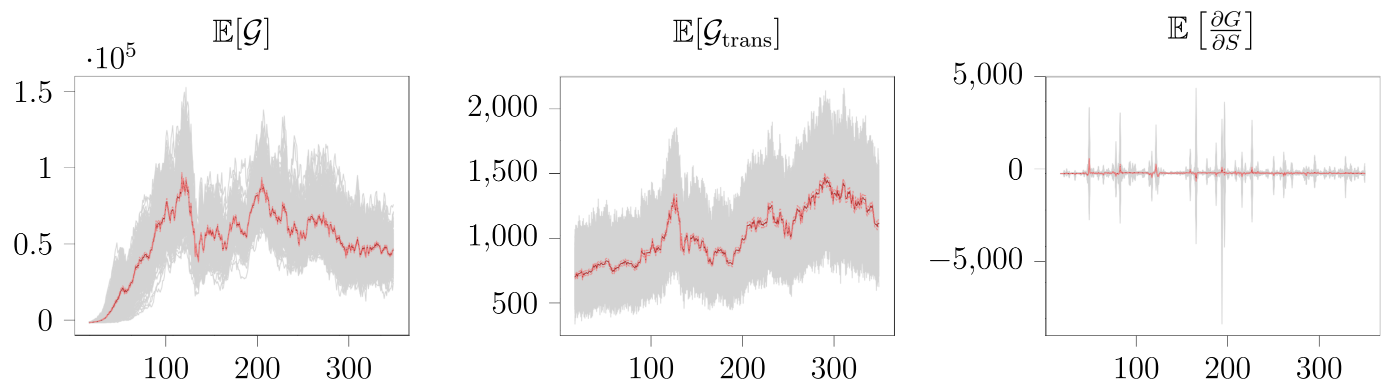

Figure 2: (Left) Mean total revenue $\mathcal{G}$. (Middle) Mean daily transactional revenue. (Right) Mean sensitivity. Gray lines represent individual realization of the process. Red solid curves represent the mean values, while red, dashed lines represent a 95% confidence interval.

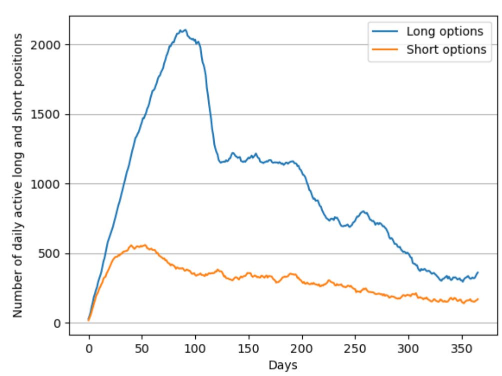

Figure 3: Number of active longs and shorts position.

4.2.2 Scenario 2

We now simulate a sort of buyer-seller parity; where, for each t_i in the Monte Carlo simulation, we set the number of buyers b_i and sellers s_i at that as

$$b_{t_i} \sim \lceil \mathcal{U}(R) \rceil \,,$$
$$s_{t_i} = b_{t_i},$$

with $R = \texttt{range\_daily\_number\_buyers} = [20, 50]$ as in (1). That is, at each day t_i, we first sample the number of daily buyers b_{t_i} uniformly from the range R, and set the number of daily sellers s_{t_i} equal to the number of daily buyers. We repeat a similar experiment as before and present our results in Figure 4. Notice that in this case the mean $\mathcal{G}$ is lower than on the previous scenario, while the sensitivity remains about the same.

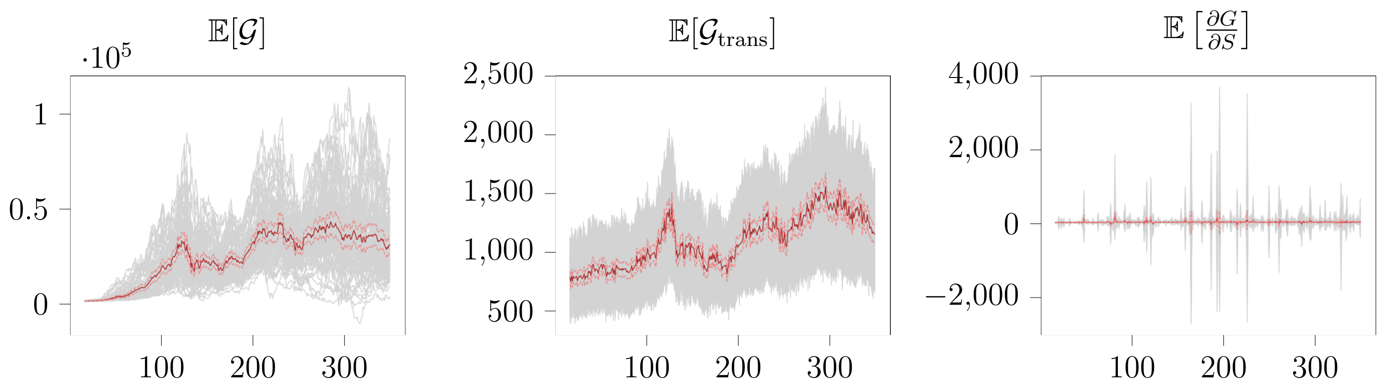

Figure 4: (Left) Mean total revenue $\mathcal{G}$. (Middle). Mean daily transactional revenue. (Right). Mean sensitivity. Gray lines represent individual realization of the process. Red solid curves represent the mean values, while red, dashed lines represent a 95% confidence interval.

5 Modeling perpetual futures with DeSim

DeSim can not only be employed to test everlasting options for different types of underlying assets (e.g., crypto assets, forex, etc.) but also to test perpetual futures contracts with a slight modification of some modules. Specifically, in the *computations block*, the daily mark price and daily funding fee modules would need to be modified for future perpetual, accordingly. Moreover, the funding fee payment module would collect fees from sellers and pay them to buyers when the value of the funding fee is negative. Finally, the probabilistic models for closing buyers' positions need to be updated by considering the unrealized loss incurred by the buyers due to the price movement of the underlying asset in an unfavorable direction and daily payments received due to negative funding fees.

6 Conclusion

In this article, we covered the background of everlasting options, compared them to perpetual futures, and introduced our derivatives simulator DeSim, a discrete event-driven simulator for testing the performance of derivatives such as everlasting options on different underlying assets. We utilized DeSim to test how everlasting ETH options will perform using last year's ETH-USDC spot price data. Finally, we highlighted how DeSim can be utilized for testing perpetual futures contracts by slightly modifying some of the modules of DeSim. Lastly, the second part of this two-part article series will cover additional functions and features, including a new data module added to DeSim.