Pricing financial derivatives using Monte Carlo and Differential Machine Learning

Mikkel Reich, Mathias Porsgaard

Faculty of Social Sciences
University of Copenhagen

1 Introduction

Pricing of financial derivates is a fundamental problem in quantitative finance. A financial derivative is a contract whose value is derived from the performance of an underlying asset, index, or interest rate. Common examples include options, futures, and swaps. The accurate pricing of these instruments is crucial for risk management, investment strategies, and market efficiency. Barring the exception of simple cases like the Black-Scholes model, pricing financial derivatives often involves complex mathematical models and numerical methods. Monte Carlo simulation is one such numerical method that is widely used due to its flexibility and ability to handle high-dimensional problems. However, Monte Carlo methods can be computationally expensive, especially when high precision is required due to the estimators standard error being $1/\sqrt{N}$ convergent. By leveraging advancements in machine learning, we can potentially enhance the efficiency and accuracy of derivative pricing. This paper explores the integration of Monte Carlo simulation with differential machine learning techniques (MCDML) introduced in the CITEDIFFpaper to improve the pricing of financial derivatives.

1.1 EU Call Options and the Black-Scholes Model

Modelling the dynamics of financial markets is a complex task, as markets are influenced by a multitude of factors including economic indicators, investor sentiment, and geopolitical events. For the purpose of pricing financial derivatives, we often rely on simplified mathematical models that capture the essential features of asset price movements. We begin our exploration with the Black-Scholes model for European Call options due to its simplicity and the existence of a closed-form solution, which allows us to benchmark our MCDML methods effectively.

The underlying asset (a stock) is assumed to follow a geometric Brownian motion (GBM), which is a continuous-time stochastic process, defined by the Stochastic Differential Equation (SDE):

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where S_t is the stock price at time t, μ is the drift (expected growth rate), σ is the volatility and W_t is a Brownian motion ((normal) random shocks). So, the stock drifts upward with a rate of μ but is also subject at every instant to random fluctuations due to $\sigma S_t dW_t$ term.

The solution to this SDE is,

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right).$$

From here one can show that at a fixed time T, the stock price S_T is lognormal,

$$S_T = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)T + \sigma\sqrt{T}Z\right), \quad Z \sim N(0,1).$$

Which is the well-known Black-Scholes assumption.

Now, an Option, is a financial derivative that gives the holder the right, but not the obligation, to buy or sell an asset at a specified price (the strike price) on or before a specified date (the expiration date). An European Call Option is a type of option that gives the holder the right to buy an asset at a specified strike price only on the expiration/maturity date. So, the payoff at maturity is,

$$\Pi = \max(S_T - K, 0),$$

where K is the strike price. If the stock ends up above the strike price, the option is exercised and the holder makes a profit of $S_T - K$. If the stock ends up below the strike price, the option is not exercised and the payoff is zero - "thrown away".

In finance, the "fair price" today of such an Option, is the discounted expected payoff under the risk-neutral measure $\mathbb{Q}$,

$$C_0 = e^{-rT}\mathbb{E}^{\mathbb{Q}}[\Pi] = e^{-rT}\mathbb{E}^{\mathbb{Q}}[\max(S_T - K, 0)], \tag{1}$$

where r is the risk-free interest rate. Which simply states, that the future payoff is random, and discounted back to present value, and the expectation is taken over all random paths.

2 Monte Carlo Estimator

The problem that arises, is that we oftent cant compute (1) analytically. So instead, we simulate.

Algorithm 1 Crude Monte Carlo for European Call Option Pricing

Input: Initial stock price S_0, strike price K, time to maturity T, risk-free rate r, volatility σ, number of simulations N.
Output: Estimated option price $\hat{C}_N$
for $i = 1$ to N **do**
    Generate $Z_i \sim N(0,1)$
    Compute stock price at maturity: $S_T^{(i)} = S_0 \exp((r - 0.5\sigma^2)T + \sigma\sqrt{T}Z_i)$
    Compute payoff: $\Pi^{(i)} = e^{-rT} \max(S_T^{(i)} - K, 0)$
end for
Compute estimated option price: $\hat{C}_N = \frac{1}{N}\sum_{i=1}^{N} \Pi^{(i)}$
Return $\hat{C}_N$

2.1 Variance Reduction Techniques

One problem regarding Monte Carlo, is noise, since we are averaging random numbers. It can be shown that the MC estimator for a R.V. Z, defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with finite variance $\sigma^2 = \text{Var}(Z) < \infty$, is unbiased, $\mathbb{E}[\hat{Z}_N] = \mathbb{E}[Z]$, and that a CLT applies for $N \to \infty$, such that the sample mean converges to a normal distribution with mean z and standard error $\sigma/\sqrt{N}$. From here, it is clear the standard error, converges towards zero at a rate of $\sqrt{N}$. So to halve the standard error, we need to increase the number of simulations by a factor of 4. This can be computationally expensive, and so variance reduction techniques are often used to reduce the variance σ^2 of the estimator, and thereby improve precision without having to increase N.

In this paper, we will use two simple variance reduction techniques, Antithetic Variates and Control Variates.

2.1.1 Antithetic Variates

If the randomness is symmetric, we can use opposites. So instead of simulating one $Z \sim N(0,1)$, we simulate two, Z and $-Z$. The paths from these two draws are negatively correlated, and averaging them cancels out some of the noise.

2.1.2 Control Variates

Say we want to estimate $E(X)$ but that X is noisy. We also observe another (related) variable Y, for which we know $E(Y)$. If X and Y are correlated, we can use Y to reduce the noise in our estimate of $E(X)$. The idea is to consider the new variable,

$$X' = X - \beta(Y - E(Y)),$$

Now, we don't know the Option price exactly, but we do know the expectation of the underlying stock price itself, $E(S_T) = S_0 e^{rT}$. So we can use the stock price as a control variate.

2.2 ML Angle

Monte Carlo is accurate, but slow (especially at high precision). Say we want Option prices for many different parameters (e.g. S_0, K, T, r, σ), the idea is then to use Machine Learning to learn the mapping from model parameters (e.g. S_0, K, T, r, σ) to Option prices. This is a regression problem, where we want to learn a function $f : \mathbb{R}^5 \to \mathbb{R}$, that maps the 5 input parameters to the Option price. Without ML, we would have to rerun the simulation for each new set of parameters. With ML, we can train a model on a large dataset of simulated Option prices, and then use the trained model to predict Option prices for new sets of parameters almost instantly.