Bocconi

BOCCONI UNIVERSITY

MAFINRISK:
Specialized Master of Quantitative Finance & Risk Management

COMPUTATIONAL METHODS & MACHINE LEARNING

GROUP MEMBERS:

Federico La Penna, Tommaso Zazzaron, Tobia Gianola

Francesco Saverio Bratta, Mara Popescu, Mariela Kodzhebasheva

Academic year 2024/2025

Contents

Introduction

This project aims to derive reasonable no-arbitrage prices for a set of simple derivatives written on an asset exhibiting "non-standard" dynamics. In Section 1, the report focuses on the theoretical analysis of a given stochastic differential equation (SDE) and the empirical assessment of the underlying time series characteristics, primarily by evaluating Gaussian behavior. Subsequently, the optimal set of parameters fitting the empirical data is obtained through maximum likelihood estimation (MLE), utilizing the Euler discretization scheme of the SDE. Sections 2 and 3 are dedicated to deriving inception no-arbitrage prices for three different types of derivative contracts based on the underlying asset: a European put option (priced using three numerical approaches: Monte Carlo simulation, lattice discretization, and a PDE scheme), the American version of the European put option, and a Down-and-Out barrier variant of the same put option.

1 Section 1: Preliminary Analysis & Parameters Estimation

1.1 Market Variable Analysis

The time series $\{x_t\}_{t=0}^{N}$ of N+1 observations of the daily price of an unknown underlying, $\{X_t\}_{t\geq 0}$, can be modeled by means of the following stochastic differential equation

$$\mathrm{d}X_t = \kappa X_t \mathrm{d}t + \sigma X_t^{\gamma} \mathrm{d}W_t^{\mathbb{Q}} \tag{1.1}$$

where $\{W_t^{\mathbb{Q}}\}_{t\geq 0}$ is a standard Brownian motion under a risk-neutral probability measure and $\Theta = (\kappa, \sigma, \gamma)$ are three non-negative parameters to be estimated.

The primary goal of this first section is to identify a suitable candidate for the underlying specification of $\{X_t\}_{t\geq 0}$. This objective can be evaluated by analyzing and comparing both the observed time series (Figure 1.1) and the structure of the corresponding SDE (1.1).

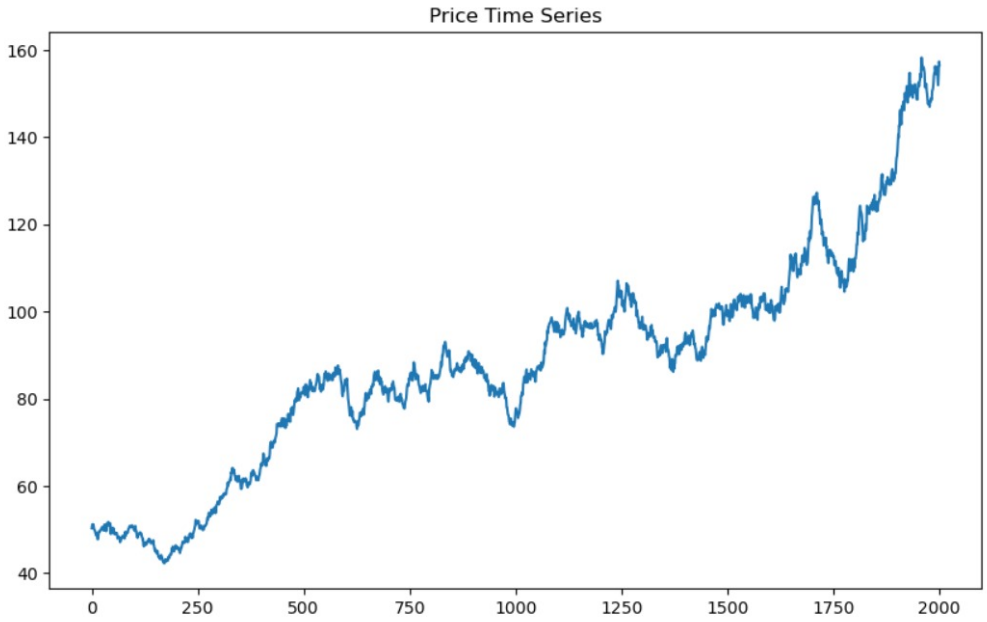

Figure 1.1: Price Time Series.

In the observed time window (Figure 1.1) of 2,000 daily observations, the initial price was 50.33 and the maximum price reached 157.37. The time series appears to exhibit an upward trend. In light of these empirical observations, the analysis now turns to the corresponding SDE.
The SDE displays a relatively simple form that excludes more complex stochastic processes. In particular, processes such as Ornstein-Uhlenbeck or CIR can be ruled out because the SDE lacks a mean-reverting component. Similarly, jump-diffusion models are not applicable since the equation does not include any jump terms, and models like Heston's are excluded based on preliminary parameter analysis indicating that the parameters are not driven by any stochastic process.
The empirical assessment of the time series corroborates these findings, as it shows an upward drift with no evident jump patterns. Based on this combined analysis, it is inferred that X_t represents

an equity price characterized by high volatility and potential for value growth over time

Following the identification of the process, validation was performed to assess whether the shocks in the underlying price are normally distributed. An analysis of the $\mathrm{d}W_t^{\mathbb{Q}}$ term from the discretized SDE confirmed that, under the assumption that the model holds, the shocks to the underlying price are indeed normally distributed. This result was obtained firstly by employing the Euler discretization of the given SDE, which is justified by the process's Markovian nature. The Euler discretization is given by

$$X_{t+\Delta t} = X_t + \kappa X_t \, \Delta t + \sigma X_t^{\gamma} \underbrace{\Delta W_t^{\mathbb{Q}}}_{\substack{\sim N(0,\Delta t) \\ = \sqrt{\Delta t}\, N(0,1)}} \tag{1.2}$$

Subsequently, isolating N(0,1) on the RHS in the above expression and empirically assessing the left-hand side using the observed time series, the normality of the price shocks was confirmed.
Indeed, based on an educated guess for $\Theta = (\kappa, \sigma, \gamma)$, specifically $\kappa = 0.08$, $\gamma = 1$, $\sigma = 0.15$, the empirical probability density function (PDF) of the shocks, which is an unbiased and consistent estimator of the true PDF, closely matches that of a normal distribution (Figure 1.2), implying that the underlying asset's price shocks follow a normal distribution. The empirical PDF has been computed using the central difference estimator of $F'(x)$:

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{F(x+\varepsilon) - F(x-\varepsilon)}{2\varepsilon} \tag{1.3}$$

Where F(x) is the empirical CDF of X.

The Q-Q plot (Figure 1.3) further supports the normality conclusion, as the data points align closely with the 45° line.

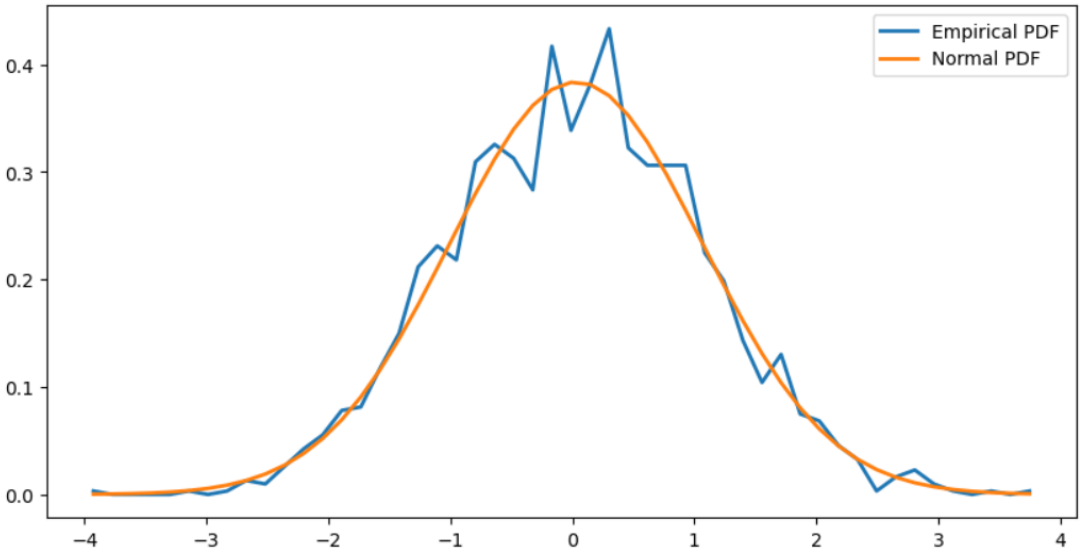

Figure 1.2: Empirical and Theoretical PDF.

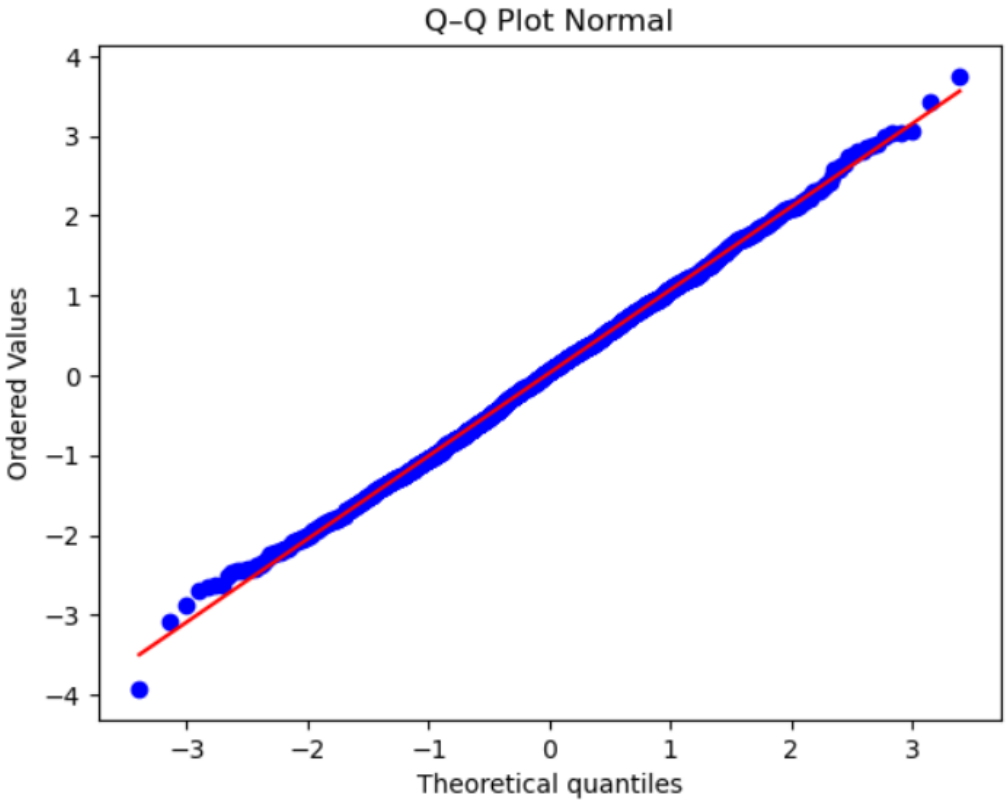

Figure 1.3: Q-Q Plot.

1.2 Maximum Likelihood Estimation

This second chapter aims to analyze the MLE procedure that has been implemented to find the best fit of parameters $\Theta = (\kappa, \sigma, \gamma)$.
Relying on the normality assumption of the shocks $\{z_t\}_{t=1}^N$, and having exploited the Euler discretization of the SDE, the following claim naturally follows:

$$X_{t+\Delta t} = X_t + \kappa X_t \Delta t + \sigma X_t^\gamma \Delta W_t^{\mathbb{Q}} \sim \mathcal{N}(a(t, X_t), b(t, X_t)) \tag{1.4}$$

with

$$\begin{aligned} a(t, X_t) &= \mathbb{E}[X_{t+\Delta t}] = X_t + \kappa X_t \Delta t \\ b(t, X_t) &= VAR(X_{t+\Delta t}) = \sigma^2 X_t^{2\gamma} \Delta t \end{aligned} \tag{1.5}$$

$$\implies X_{t+\Delta t} \sim \mathcal{N}(X_t + \kappa X_t \Delta t, \sigma^2 X_t^{2\gamma} \Delta t) \tag{1.6}$$

To find the maximum likelihood estimation (MLE) for Θ, we pass through the classic minimization of the log-likelihood function. Specifically: As $\{x_i\}_{i=0}^d \sim$ i.i.d. X, its log-likelihood function is:

$$\log \mathcal{L}(\Theta, x) = \sum_{i=1}^{d} \ln f_X(x_i, \Theta) \tag{1.7}$$

The MLE is then:

$$\hat{\Theta}^{MLE} = \arg \max_{\Theta \in \Theta_D} \log \mathcal{L}(\Theta, x) = \arg \min_{\Theta \in \Theta_D} -\log \mathcal{L}(\Theta, x) \tag{1.8}$$

The bounds used for the optimization are:

- $\kappa \geq 0$, as all the parameters are known to be non-negative;
- $0 \leq \sigma \leq 1$, to restrict volatility;
- $0 \leq \gamma \leq 1.5$, to control the diffussive coefficient such that the process does not explode.

The estimated parameters that match the historical prices are therefore:

$$\kappa \approx 0.1544 \qquad \sigma \approx 0.1348 \qquad \gamma \approx 1.0328 \tag{1.9}$$

These values appear reasonable: κ and σ reflect a pronounced drift component and a suitable level of volatility, aligning with the interpretation of the underlying as a stock. Moreover, γ diverges slightly from the classical GBM exponent of 1, indicating a non-explosive behavior akin to a geometric Brownian motion, but not identical to it.

2 Section 2: European Put Option Valuation

As mentioned before, the second part of this report is dedicated to deriving inception no-arbitrage prices for three different types of derivative contracts written on the underlying asset $\{X_t\}_{t\geq 0}$: a European put option, the American version of it, and a Down-and-Out barrier variant of the same put option. This section will deal with the European Put option pricing by means of three different numerical approaches: firstly the Monte Carlo approach, then the Lattice discretization and finally the PDE-scheme.

2.1 Monte Carlo Approach

The pricing of a European put option via the Monte Carlo approach is performed by simulating multiple paths of the underlying asset and computing the discounted expected payoff until inception. The key input data utilized in this simulation are the following:

- **Spot inception price** X_0, which is set to the last available value in the time series (=156.621);
- **Strike price** $K = X_0$;
- **Maturity** $T = 1.2$ years and 250 trading days per year;
- **Risk-free rate** $r = 4\%$ per annum;
- **MLE-based parameters**:
  - $\kappa \approx 0.1544$
  - $\sigma \approx 0.1348$
  - $\gamma \approx 1.0328$

Moreover, the pricing formula for the European put option is:

$$\pi_0^E = e^{-rT}\mathbb{E}^{\mathbb{Q}}\left[\max(K - X_T, 0)\right] \tag{2.1}$$

where:

- π_0^E is the price of the option at time $t = 0$;
- K is the strike price;
- X_T is the price of the underlying asset at maturity T;
- r is the risk-free rate;
- T is the time to maturity.

Modeling and simulating the Asset Price Dynamics

The asset price X_t is implemented algorithmically based on the previously introduced Euler discretization (1.2) of the underlying SDE. The Monte Carlo simulation is executed with 10,000 simulations (denoted N_{sim}) of the asset paths, employing two nested for-loops, as shown in the following plot:

Listing 1: Monte Carlo simulation

```
def MC_BSPutOption(NSim, k, sigma, gamma, dt, K, r, T, c, X0):
    X = np.zeros(shape=(NSim, c+1))
    X[:, 0] = X0
    for i in range(int(NSim)):
        for j in range(int(c)):
            X[i, j+1] = X[i, j] + k * X[i, j] * dt \
                        + sigma * (X[i, j] ** gamma) * np.sqrt(dt) *
                            np.random.normal(0, 1)
    output = np.zeros(shape=2)
    discPayoff = np.exp(-r*T)*np.maximum(0,K-X[:,-1])
    output[0] = discPayoff.mean()
    output[1] = discPayoff.std()*1.96/np.sqrt(NSim)
```

Each path represents a potential future realization of the asset price over time, from the initial price X_0 to the price at maturity X_T.
The implementation has been conducted via a discretization of the time horizon $T = 1.2$ years into $c = 300$ discrete time steps (since there are 250 trading days in a year), in which the asset price is updated iteratively.

Estimating the Option Price

After simulating multiple paths for the asset price and computing the discounted payoffs, the price of the option is the mean of the discounted payoffs across all simulations:

$$\pi_0^E \approx \frac{1}{N_{\text{sim}}} \sum_{i=1}^{N_{\text{sim}}} \text{Discounted Payoff}_i \tag{2.2}$$

The algorithm reports also the confidence interval for the estimated option price, computing it's radius as:

$$\text{Confidence Interval radius} = 1.96 \times \frac{\sigma_{\text{payoff}}}{\sqrt{N_{\text{sim}}}} \tag{2.3}$$

where σ_{payoff} is the standard deviation of the discounted payoffs. Upon running the Monte Carlo simulation, the estimated price of the European put option is:

$$\pi_0^E \approx 2.32938 \tag{2.4}$$

with a 95% confidence level.

2.2 Lattice Methodologies

The lattice technique implementation relies on the assumption of constant and equals ΔX over time. Even if we end up with $\Delta x^- \neq \Delta x^+$ the algorithm is still feasible as it is recombining but at a different level w.r.t the initial $(\hat{X}_t)$. The key assumption that must hold for the algorithm to be feasible is the constancy over time. If the ΔX changes over time, the algorithm is no more recombining, ending up with an explosive implementation (2^n cases $\forall\, n$ step).
For a generic SDE,

$$\mathrm{d}X_t = a(t, X_t)\,\mathrm{d}t + b(t, X_t)\,\mathrm{d}W_t,$$

the lattice discretization allows to implement such process as:

$$\begin{cases} \hat{X}_{t+\Delta t} = \begin{cases} \hat{X}_t + \Delta X, & \text{for } q = \max\Big\{0,\, \min\Big\{1,\, \frac{1}{2} + \dfrac{a(t; X_t)\,\sqrt{\Delta t}}{2\,c}\Big\}\Big\}, \\ \hat{X}_t - \Delta X, & \text{for } 1 - q, \end{cases} \\ \hat{X}_0 = x_0, \end{cases}$$

where $\Delta X = b(t, X_t)\,\sqrt{\Delta T}$. Then, ΔX in order to be constant over time (surely path independent), must have $b(t, X_t)$ constant. Feasible $\iff$ $b(t, X_t) = b$.
Let's look now at the process:

$$\mathrm{d}X_t = \underbrace{k\,X_t}_{a(t,X_t)}\,\mathrm{d}t \;+\; \underbrace{\sigma\,X_t^{\gamma}}_{b(t,X_t)=b(X_t)}\,\mathrm{d}W_t^{Q}.$$

The term $b(t, X_t)$ depends on X_t^{γ} and is therefore not constant. However, it is still possible to use an *invertible function simulation*. The key idea is to find a transformation of $b(t, X_t)$ that makes it constant, simulate the process of $f(X_t)$, and then revert to X_t using the same invertible mapping $f\colon D \to \mathbb{R}$, with $f^{-1}\colon \mathbb{R} \to D$.

1st step: An f is sought so that $b(t, X_t)$ becomes constant:

$$f = \int \frac{1}{b(t, X_t)}\,\mathrm{d}x = \int \frac{1}{\sigma X_t^{\gamma}}\,\mathrm{d}x.$$

This problem then splits into two cases (case $\gamma \neq 1$ and the classical GBM case $\gamma = 1$). However, a maximum likelihood estimation of the parameters indicates that $\gamma \neq 1$ applies:

Case $\gamma \neq 1$:

$$f = \frac{1}{\sigma(-\gamma + 1)}\,X_t^{-\gamma+1} \quad\Longrightarrow\quad f(X_t) = X_t^{-\gamma+1}.$$

We can see that with this transformation, the stochastic differential for $f(X_t)$ has a constant diffusive component:

$$\begin{aligned} \mathrm{d}\big(f(X_t)\big) &= \mathrm{d}\big(X_t^{-\gamma+1}\big) \\ &= (-\gamma+1)\,X_t^{-\gamma}\,\mathrm{d}X_t \;+\; \tfrac{1}{2}\,(-\gamma+1)(-\gamma)\,X_t^{-\gamma-1}\,\mathrm{d}\langle X, X\rangle_t \\ &= (-\gamma+1)\,X_t^{-\gamma}\left(k\,X_t\,\mathrm{d}t + \sigma\,X_t^{\gamma}\,\mathrm{d}W_t^{Q}\right) \;+\; \tfrac{1}{2}\,(-\gamma+1)(-\gamma)\,X_t^{-\gamma-1}\,\sigma^2\,X_t^{2\gamma}\,\mathrm{d}t \\ &= \Big[\underbrace{(-\gamma+1)\,X_t^{-\gamma+1}\,k \;+\; \tfrac{1}{2}\,(-\gamma+1)(-\gamma)\,X_t^{\gamma-1}\,\sigma^2}_{=l\ (\text{call it } l)}\Big]\,\mathrm{d}t \;+\; \underbrace{(-\gamma+1)\,\sigma}_{\text{Constant}}\,\mathrm{d}W_t. \end{aligned}$$

Let's now implement the lattice techniques. Given that $\Delta f = (-\gamma+1)\,\sigma\sqrt{\Delta T}$

$$\begin{cases} (\hat{X}_{t+\Delta T})^{-\gamma+1} = (\hat{X}_t)^{-\gamma+1} + (-\gamma+1)\,\sigma\sqrt{\Delta T}, & \text{for } q = \max\Big\{0,\, \min\Big\{1,\, \frac{1}{2} + \frac{l\sqrt{\Delta t}}{2\,\sigma\,(1-\gamma)}\Big\}\Big\}, \\ (\hat{X}_{t+\Delta T})^{-\gamma+1} = (\hat{X}_t)^{-\gamma+1} - (-\gamma+1)\,\sigma\sqrt{\Delta T}, & \text{for } 1-q, \\ \hat{X}_0^{-\gamma+1} = \hat{x}_0^{-\gamma+1}. \end{cases}$$

$$f^{-1}(y) = y^{\frac{1}{-\gamma+1}} \quad \text{and then:}$$

$$\begin{cases} \hat{X}_{t+\Delta T} = \big((\hat{X}_t)^{-\gamma+1} + (-\gamma+1)\,\sigma\sqrt{\Delta T}\big)^{\frac{1}{-\gamma+1}}, & \text{for } q = \max\Big\{0,\, \min\Big\{1,\, \frac{1}{2} + \frac{l\sqrt{\Delta t}}{2\,\sigma\,(1-\gamma)}\Big\}\Big\}, \\ \hat{X}_{t+\Delta T} = \big((\hat{X}_t)^{-\gamma+1} - (-\gamma+1)\,\sigma\sqrt{\Delta T}\big)^{\frac{1}{-\gamma+1}}, & \text{for } 1-q, \\ \hat{X}_0 = \hat{x}_0. \end{cases}$$

Here below is reported the algorithm implementation of such lattice methodology:

Listing 2: Lattice Methodology

```
f_X_sim = np.zeros(shape=(c+1, c+1))
f_X_sim[0,0] = X0**(1-gamma)

for i in range(int(c+1)):
    for j in range(i, int(c+1)):
        f_X_sim[i,j] = f_X_sim[0,0] + (j-2*i)*(-gamma+1)*sigma*np.sqrt(dt)
X_sim = (f_X_sim)**(1/(1-gamma))

prob = np.zeros(shape=(c+1, c+1))

for i in range(int(c+1)):
    for j in range(i, int(c+1)):
        a = (1-gamma)*f_X_sim[i,j]*k +
           0.5*(1-gamma)*(-gamma)*(X_sim[i,j]**(gamma-1))*sigma**2
        prob[i,j] = np.maximum(0, np.minimum(1, 0.5 +
           (a*np.sqrt(dt))/(2*sigma*(1-gamma))))

Eur_Put = np.zeros(shape=(c+1, c+1))
Eur_Put[:,-1] = np.maximum(0, K - X_sim[:,-1])
discFct = np.exp(-r*dt)

for i in range(int(c)):
    for j in range(int(c-i)):
        p_val = prob.iloc[j, c-i-1]
        Eur_Put[j, c-i-1] = discFct * (p_val * Eur_Put[j, c-i] + (1 -
           p_val) * Eur_Put[j+1, c-i])
Put_0_Lattice = Eur_Put[0,0]
```

Upon running the Lattice simulation, the estimated price of the European put option is:

$$\pi_0^E \approx 2.04789 \tag{2.5}$$

2.3 PDE-Scheme Pricing

In an arbitrage-free market, the partial differential equation (PDE) is a requirement on the discounted price process of any traded security asset. Under risk-neutral probability $\mathbb{Q}$ it must be a martingale. For a derivative $f(t, X_t)e^{-rt}$ written on a generic X_t (as in this case) this holds true if the SDE is driftless.
By Itô's lemma, imposing that the drift coefficient equal to zero:

$$\begin{aligned}
\mathrm{d}(f(t,X_t)e^{-rt}) &= (f_t e^{-rt} - re^{-rt}f)\mathrm{d}t + e^{-rt}f_x \mathrm{d}X_t + \frac{1}{2}e^{-rt}f_{xx}\mathrm{d}\langle X, X\rangle_t \\
&= (f_t e^{-rt} - re^{-rt}f)\mathrm{d}t + e^{-rt}f_x(\kappa X_t + \sigma X_t^{\gamma}\mathrm{d}W_t^{\mathbb{Q}}) + \frac{1}{2}e^{-rt}f_{xx}\sigma^2 X_t^{2\gamma}\mathrm{d}t \\
&= e^{-rt}\left[\underbrace{\left(-rf + f_t + f_x(\kappa X_t) + \frac{1}{2}f_{xx}\sigma^2 X_t^{2\gamma}\right)}_{\textit{must be set equal to } 0}\mathrm{d}t + f_x\sigma X_t^{\gamma}\mathrm{d}W_t^{\mathbb{Q}}\right] \\
&\implies rf = f_t + f_x(kX_t) + \frac{1}{2}f_{xx}\sigma^2 X_t^{2\gamma}
\end{aligned} \tag{2.6}$$

The simulation approach is based on a numerical PDE-scheme, where the implementation is done after a discretization of the domain of $f(\cdot)$.
As $f(\cdot) : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}$, in order to simulate we impose restrictions on $\mathbb{R}^+ \times \mathbb{R}^+$, becoming $[0, T] \times [0, \overline{X}]$. By means of uniform partitions m_t and m_x, we can now construct the grid to simulate the PDE-scheme:

$$\mathbb{G}\{(i\Delta_t, j\Delta_x) : i = 0, \ldots, m_t; j = 0, \ldots, m_x\} \tag{2.7}$$

with $\Delta_t = {}^{T}/_{m_t}$ and $\Delta_x = {}^{\overline{X}}/_{m_x}$.
We can now exploit the central difference estimator and the backward difference estimator for the first and second order derivative in the PDE:

$$f_t = \frac{f^{i,j} - f^{i-1,j}}{\Delta_t} \tag{2.8}$$

$$f_x = \frac{f^{i,j+1} - f^{i,j-1}}{2\Delta_x} \tag{2.9}$$

$$f_{xx} = \frac{f^{i,j+1} - 2f^{i,j} + f^{i,j-1}}{\Delta_x^2} \tag{2.10}$$

Substituting these discrete partial derivatives into Equation 2.6:

$$rf = \frac{f^{i,j} - f^{i-1,j}}{\Delta_t} + \frac{f^{i,j+1} - f^{i,j-1}}{2\Delta_x}(\kappa X_t) + \frac{1}{2}\frac{f^{i,j+1} - 2f^{i,j} + f^{i,j-1}}{\Delta_x^2}\sigma^2 X_t^{2\gamma} \tag{2.11}$$

Then, isolating $f^{i-1,j}$ for the numerical implementation on the grid and recalling that the boundary condition at T is $f(T, x) = F(T)$. Then, $(T, j\Delta_x)$ contains the value of the payoff at maturity $(\forall X_t)$:

$$f^{i-1,j} = \Delta_t\left[\frac{f^{i,j}}{\Delta_t} - rf + \frac{f^{i,j+1} - f^{i,j-1}}{2\Delta_x}(kj\Delta_x) + \frac{1}{2}\frac{f^{i,j+1} - 2f^{i,j} + f^{i,j-1}}{\Delta_x^2}\sigma^2(j\Delta_x)^{2\gamma}\right] \tag{2.12}$$

Listing 3: PDE Scheme

```
# Bounds
G = np.zeros(shape=(mt+1, ms+1))
# bottom
G[-1, :] = np.maximum(0, K - nodes_X)
# right
G[:, -1] = 0
# left
G[:, 0] = np.flip(K * np.exp(-r * nodes_t))

for i in range(1, mt+1):
    for j in range(1, ms):
        G[mt-i, j] = (G[mt-i+1, j] / dt - r * G[mt-i+1, j] +
                      (G[mt-i+1, j+1] - G[mt-i+1, j-1]) / (2 * dx) * (k *
                         j * dx) +
                      0.5 * sigma**2 * (j * dx)**(2 * gamma) *
                      (G[mt-i+1, j+1] - 2 * G[mt-i+1, j] + G[mt-i+1,
                         j-1]) / (dx**2)) * dt

Grill = pd.DataFrame(G)
P0_PDE = G[0, 40]
```

Upon running the PDE-scheme simulation, done by setting boundaries on the discretized grid (as shown in the Listing 3 box, on the bottom boundary lies the payoff condition, in the right boundary 0, while on the left boundary the inception value with $S \downarrow 0$) and by setting hyperparameters that guarantees stability, the estimated price of the European put option is:

$$\pi_0^E \approx 2.06318 \tag{2.13}$$

2.4 Comparison

This section finally presents a visual comparison of the three inception put prices (2.4, 2.5, and 2.13) obtained using the Monte Carlo approach, lattice discretization, and a PDE scheme. The 95% confidence intervals are also shown, and the graph displays the number of simulations on the X-axis. All simulation methods seem to converge as the number of simulations increases, consistently remaining within a 95% confidence interval. This observation supports the correctness of the obtained results, suggesting a reliable inception value that oscillates around 2.1.

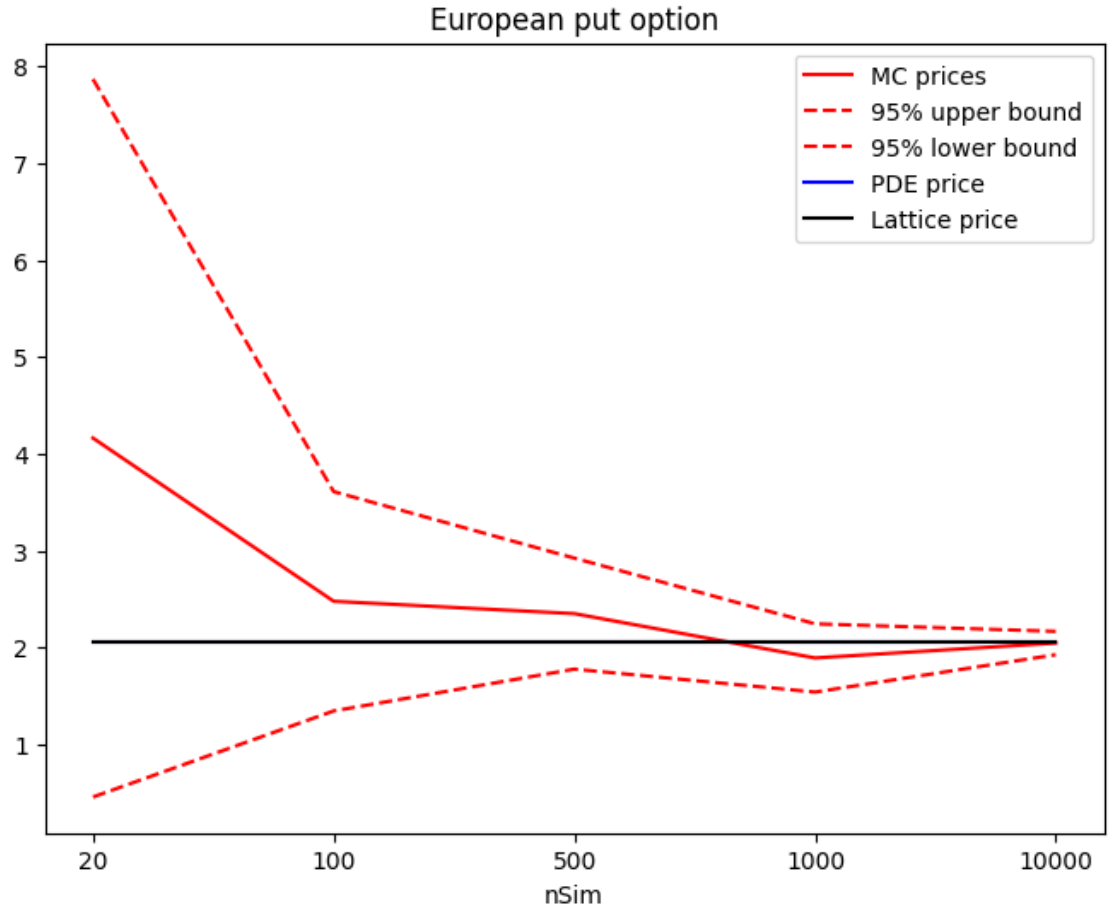

Figure 2.1: Numerical Method Image

3 Section 3: American Put Option & Down-and-Out Put Valuation

This section will deal with the pricing of the other two early mentioned derivatives contracts written on the underlying $\{X_t\}_{t\geq 0}$, namely the American version of the previously analyzed European Put option and the Down-and-Out version. The American Put inception price will be computed by both a Lattice and a PDE-scheme approach; while the Down-and-Out put option inception price will be computed by a Monte Carlo approach.

3.1 American Put Option Valuation

Lattice Approach

In general, the pricing formula for a American Put option is the following:

$$\pi_0^A = \sup_{\tau\in[0,T]} \mathbb{E}^{\mathbb{Q}}\Big[e^{-r\,\tau}\big(K - X_\tau\big)^+\Big].$$

The "sup" condition is a technical generalization that allows to not specify any assumptions about the function's characteristics in advance. However, because the domain is a compact set and X is an integrable process in L^1, the formula reduces to a straightforward dynamic maximization problem. The two components that must be taken into account in such maximization problem are the continuation value (CV) and the immediate payoff (IP):

$$\mathrm{CV} = e^{-r\Delta t}\Big(qS_x^u(t+\Delta t) + (1-q)S_x^d(t+\Delta t)\Big), \quad \mathrm{IP} = X\big(S(t)\big) \tag{3.1}$$

Indeed, they are assessed for the pricing recursion, by taking for each step the maximum between the two:

$$\pi_t^{\mathrm{A}} = \max\{\mathrm{CV}_t, \mathrm{IP}_t\} \tag{3.2}$$

Implementing this recursive maximization, a set of optimal decision points rise. The plot below (Figure 3.1), visually shows the optimal decision tracking of the American Put written on $\{X_t\}_{t\geq 0}$. The green dots in the figure represent the points in which it is more convenient not to exercise (IP exploitation), but to continue keeping the option, as greater is the continuation value. On the other hand the red dots show the moments at which the immediate payoff is more convenient and the early exercise is the optimal decision.
The distinctive pattern observed in the plotted points is a direct consequence of the inherent nature of the function. When ($\gamma > 1$), the lattice implementation tends to exhibit explosive behavior. Although an optimal set of hyperparameters can mitigate this phenomenon, it cannot entirely eliminate it. Nevertheless, the pricing approach remains reliable, as the outcomes are appropriately weighted by the actual probability of these scenarios occurring.
Upon running the Lattice simulation, the estimated price of the American put option is:

$$\pi_0^{\mathrm{A}} \approx 4.35 \tag{3.3}$$

In general, the observation that the American option is priced higher than the European option is consistent with theoretical expectations. The additional exercise rights associated with the American option enhance its value relative to the European option.

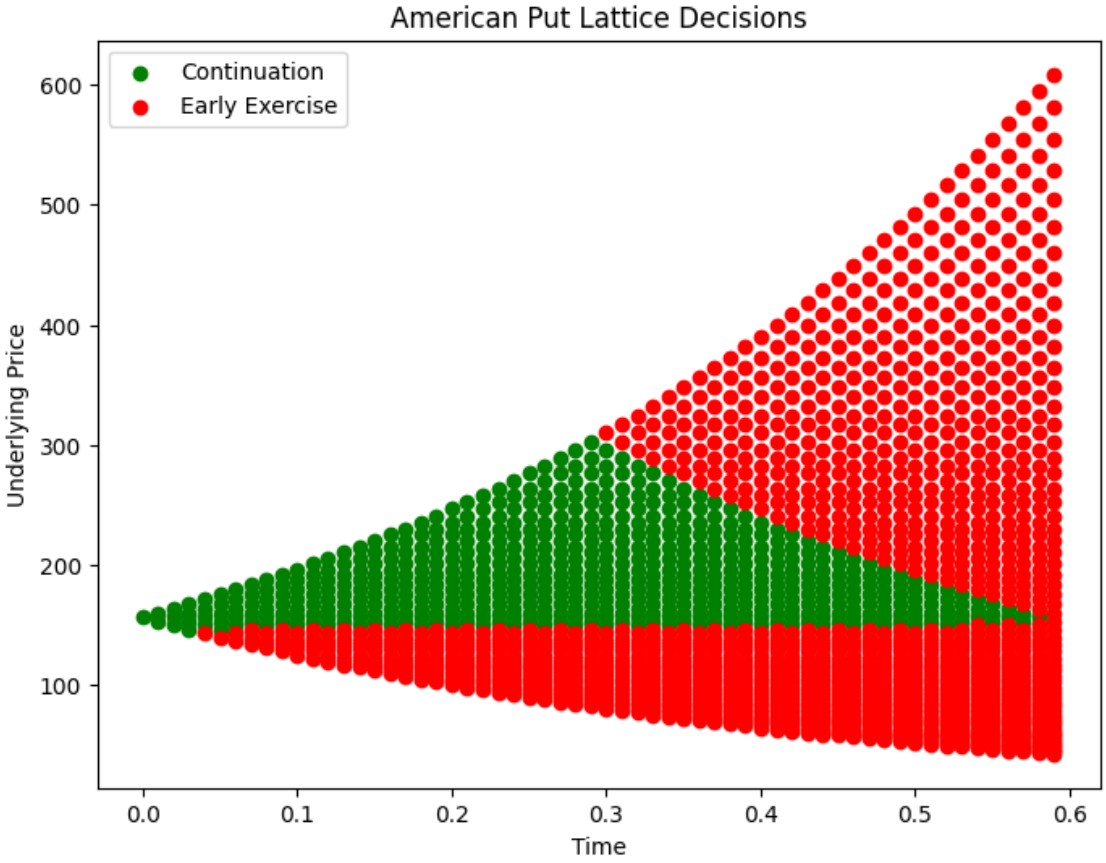

Figure 3.1: American Put Lattice Decisions.

3.2 PDE-Scheme Approach

As regards the PDE, the overall procedure adopted is the same as in Section 2.3, except for the introduction of the immediate payoff in the computation of the function. More precisely, in addition of the previously mentioned PDE-discretization, we add a maximum condition w.r.t the immediate payoff IP. Therefore, the following rise:

$$f^{i-1,j} = \max\left\{ \Delta_t\left(\frac{f^{i,j-1}}{2}\left[\sigma^2 j^2 - (r-q)j\right] + f^{i,j}\left(\frac{1}{\Delta_t - r - \sigma^2 j^2}\right) + \frac{f^{i,j+1}}{2}\left[\sigma^2 j^2 - (r-q)j\right]\right),\ \max\{K - j\Delta_x, 0\} \right\}.$$

By this implementation, the American put price is

$$\pi_0^{\mathrm{A}} \approx 4.34 \tag{3.4}$$

very close to the result obtained using the lattice methodology.

The exercise premium, meaning the cost of the "American right" computed as the differences between the American option price and the European one, is equal to 2.306 for the Lattice technique and to 2.27 for the PDE-approach.

3.3 Down-and-Out Barrier Put Option Pricing

A *down-and-out European put option* is an exotic derivative that blends the features of a standard European put option with a barrier constraint. Analogous to a conventional European put, it confers the right to sell an underlying asset at a predetermined strike price exclusively at expiration. However, the "down-and-out" feature imposes a condition whereby, if the asset's price falls to or below a specified barrier at any point during the option's lifetime, the option is immediately nullified and rendered worthless. Consequently, this path-dependent attribute complicates the pricing process relative to plain vanilla options, as the option's value depends not only on the terminal asset price but also on whether the barrier was breached. The payoff of such an option, as shown in Figure 3.2, is:

$$\begin{cases} (K - X_T)^+ & \text{if } \min_{t\in[0,T]} S_T \geq B \\ 0 & \text{else} \end{cases} = (K - X_T)^+ \mathbb{I}_{\{\min_{t\in[0,T]} S_T \geq B\}} \tag{3.5}$$

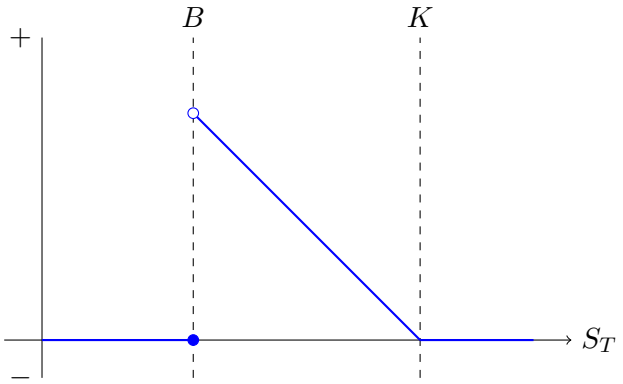

Figure 3.2: Payoff of a down-and-out barrier put option.

To determine the option's price, a Monte Carlo simulation was employed, analogous to the methodology described in Section 2.1. The simulation incorporates a condition whereby, if the underlying asset's price reaches the specified barrier level, the option's payoff is nullified. Here below is shown the algorithm implementation for the knocked-Out condition:

Listing 4: Monte Carlo Simulation for Knocked-Out Condition

```
1 if X[i,j+1] <= B:
2     knocked_out = True
3     break
4 if knocked_out:
5     payoff[i] = 0
6 else:
7     payoff[i] = np.maximum(K - X[i,-1], 0)
```

It is noteworthy that the resulting payoff array contains many zeros, as the payoff is set to zero whenever the underlying asset's price falls below the barrier. The inception value is the following, which is reasonable as it gives fewer rights than the regular put.

$$\pi_0^{DO,E} \approx 0.204 \tag{3.6}$$