MÄLARDALENS UNIVERSITY
Division of Mathematics and Physics

Mälardalen University

Seminar Report

Exploring Binomial Models for Option Pricing: Convergence and Applications

Students:
Mattia Mostarda
Paolo Nardi

Professor:
Jän Roman

Academic year 2025/2026

Abstract

This seminar report studies the convergence of different binomial models, and explores how their accuracy and efficiency change after improving them with smoothing techniques and numerical methods. In particular, the purpose of the work is to find which combination is the better one to price an option, whether it is American or European, put or call. The analysis have been performed using a Python code which implements models and smoothing techniques, and can be tested for options with different parameters. The results have been plotted in order to make them understandable. Our conclusions are that for pricing European options and American calls the better models are Leisen-Reimer with Richardson Extrapolation, CRR with averaging smoothing and Richardson Extrapolation, and Pegging the strike with Black-Scholes Smoothing and Richardson Extrapolation. Instead, we discovered that Black-Scholes smoothing can't be used for American put options. We also found that as the volatility increases the general errors becomes bigger, leading to different results. These differences, in correspondence of different options, suggest that our code is not so efficient and solid, and takes to errors especially when we consider high volatile underlyings.

Contents

Introduction

In finance, an **option** is a contract which gives to the *holder*, the right, BUT not the obligation, to buy or sell a specific quantity of an underlying asset at a specified price, called *strike price*, on or before a specified date, called *maturity*, depending on the style of the option. The option's holder is said to take a long position while the option's writer is said to take a short position.

When you buy an option, the writer takes on the risk of large future price movements, this risk requires a compensation, that is the option price. By the way, valuing an option is complex because its payoff depends on the future behavior of the underlying asset, which is very difficult to foresee. At the moment a lot of different pricing techniques exists and financial analysts have reached a point where they are able to calculate with high accuracy the value of an option.

Among those different techniques, **the binomial method** has attracted the most attention since it is able to handle a variety of conditions for which other models cannot be easily applied, and it is very accurate. In particular, the binomial model discretizes the trend of an underlying asset over a period of time rather than a single point, so it can be used not only to price European options (that can be exercised only at maturity), but also to value American options that are exercisable at any time until maturity, as well as Bermudan options, which can be exercised at specific instances of time.

The binomial pricing model describes the evolution of the asset price in discrete time by using a binomial tree for a certain number of steps between the valuation day and maturity. The option valuation in this method is performed in three steps:

- **Price tree generation:** We work forward to build the tree, assuming that at each step the underlying instrument will move up or down by a specific factor, u or d (by definition $u \geq 1,\ 0 < d \leq 1$ and $ud = 1$). If S is the current price of the asset, in the next period the price will be either $S_{up} = uS$ or $S_{down} = dS$. The up and down factors are computed in different ways depending on the binomial model we are considering. So if we consider N steps, we will have that the price of the stock at node (i, j),
$$S_{i,j} = S_0 u^j d^{i-j}.$$

- **Calculation of option value at final step:** At each final node of the tree, we compute the value of the option at maturity that will be
$$\begin{cases} \max{(S_N - K)}, 0 & \text{for a call option,} \\ \max{(K - S_N)}, 0 & \text{for a put option,} \end{cases}$$
where K is the strike price and S_N the price of the asset at the last step.

- **Backward calculation of the option value:** Once we completed the previous step, we start working backward in each previous node, from the penultimate step to the first, to find the fair value of the option using the risk neutrality assumption. The risk neutrality assumption states that *'today's fair price of a derivative is equal to the expected value of its future payoff discounted by the risk free interest rate'*. Then, starting from the last layer, we have that at the nodes of the previous step the option value is
$$C_{t-\Delta t,i} = e^{-r\Delta t}(pC_{t,i+1} + (1-p)C_{t,i}),$$

where $C_{t,i}$ is the option's value for the i-th node at time t, and

$$p = \frac{e^{r\Delta t} - d}{u - d},$$

is the risk neutral probability.

For a European option, there is no chance of early exercise, so we can apply this method to all the nodes of the tree till the first, where the result will be the real value of the option. Instead, for an American option, since the option may be held or exercised prior to maturity, at each node the value of the option will be the maximum between the value obtained as described in the previous lines, and the exercise value.

In this way, we can use the binomial model in order to price an option. This model is considered a correct and powerful approximation of the real world option value for two main reasons:

1. The binomial option pricing model is based on the idea you can construct a portfolio made of a stock and a risk free bond that replicates the option's payoff, so by the *Law of One Price*, the option and the portfolio must have the same price today to avoid arbitrage. Then, the option's value corresponds to the initial cost of this replicating portfolio, which is exactly the no arbitrage price.

2. The binomial model, in particular the Cox-Ross-Rubinstein (CRR) (which we will study in deep later), is the discrete equivalent of the **Black-Scholes Model**, which is the mathematical framework used to calculate the fair price of European call and put options. It consists of solving the following parabolic differential equations,

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = rV,$$

where $V(S, t)$ is the price of the option, whose underlying price is denoted by S.

The key idea is that as we increase the number of steps of the binomial tree, the price computed using the Binomial Model converges to the price calculated by the Black-Scholes Model. This is mainly due to the fact that binomial model assumes that movements in the price follow a binomial distribution, which, for many trials, approaches the log-normal distribution assumed by Black-Scholes. By the way, the power of binomial model is in its flexibility, in fact, if the Black-Scholes can only be applied to European options, the binomial model can also be used to price more complex options, such as the American options (as we will see later on).

As mentioned previously, there are several binomial models. They are all based on the main idea that the price of an asset can only go up or down in a certain time period, but they differ from each other in the way the parameters u and d are calculated. The main goal behind developing different binomial models is to improve accuracy, efficiency, and to achieve faster convergence to the Black-Scholes price, especially for a small number of steps.

In the following chapters, we will deep dive into these different binomial models, exploring their differences, and we will also take a look at some numerical methods to improve the accuracy and convergence of these models to the Black-Scholes. The purpose of this seminar report is to find pros and cons of each model, and to find the optimal combination of a binomial model and numerical techniques for accurately valuing European or American options (calls or puts).

Chapter 1

Exploring Binomial Models for European Option Pricing

In this first chapter, we will present some of the most relevant and common binomial models. We will show the differences among them in terms of parameters, convergence and errors. We will also focus on the main problem of this type of model.

1.1 Numerical Methodology and Benchmarks

To accurately evaluate the convergence and precision of the discrete binomial models, we have to establish a correct benchmark price, for each type of option under study.

1.1.1 Analytical and Numerical Benchmarks

In order to do it, we utilize a mix of analytical and numerical methods. Then, the benchmark prices will be:

- **European Options (Call and Put) and American Call:** For these option types, the benchmark price is determined using the analytical

Black-Scholes-Merton (BSM) equation. Given the parameters of this study (non-dividend paying underlying asset), the value of the American Call option is equivalent to its European counterpart, as early exercise is never optimal. The BSM solution represents the theoretical continuous-time limit to which the binomial models are supposed to converge.

- **American Put Option:** An analytical closed-form solution for the American Put option price does not exist due to the early exercise feature. Therefore, we establish a highly precise numerical price as the target benchmark. This benchmark is calculated using the **Cox-Ross-Rubenstein (CRR) model with a high step count**. We set the number of time steps to a large value, specifically $N = 10^4$ steps, to ensure the resulting price is a stable and accurate approximation of the true theoretical value.

We operate under the assumptions of the Black-Scholes model, including constant volatility and a risk-free environment. In the following lines we will show the set of financial parameters we will use along all the work to price the options, and compare them. The parameters used are:

- **Underlying Price** (S): $S = 100$
- **Strike Price** (K): $K = 100$
- **Risk-Free Interest Rate** (r): $r = 5.0\%$ (or 0.05)
- **Time to Maturity** (T): $T = 1.0$ year
- **Volatility** (σ): $\sigma = 20.0\%$ (or 0.20)

The time step (Δt) for the binomial lattice is defined by the number of time steps (N) used:

$$\Delta t = \frac{T}{N.}$$

1.2 Cox-Ross-Rubenstein Model (CRR)

The Cox-Ross-Rubenstein (CRR) model, introduced in 1979, is the first and most widely used binomial lattice model for option pricing. The CRR model is designed to approximate the continuous geometric Brownian motion of the underlying asset by setting the up (u) and down (d) factors symmetrically based on volatility (σ). It is built to ensure that the model remains simple to implement and converges to the Black-Scholes price as the number of time steps increases.

CRR Model Parameters

The parameters for the CRR model are defined as follows:

- **Up Factor** (u):
$$u = e^{\sigma\sqrt{\Delta t}}$$
- **Down Factor** (d):
$$d = e^{-\sigma\sqrt{\Delta t}}$$
- **Risk-Neutral Probability** (p): The probability p is chosen such that the expected return of the stock matches the risk-free rate ($e^{r\Delta t}$), satisfying the arbitrage-free condition $e^{r\Delta t} = pu + (1-p)d$, so
$$p = \frac{e^{r\Delta t} - d}{u - d}.$$

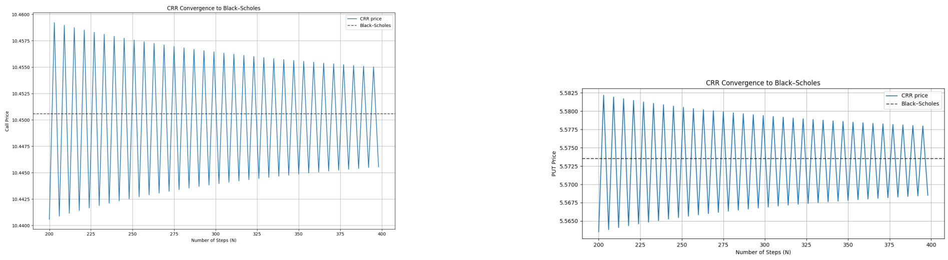


Figure 1.1: Convergence of CRR to BSM price for an European Call and Put option

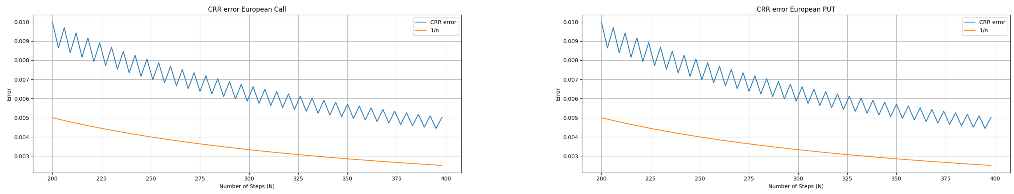


Figure 1.2: Error of CRR model for an European Call and Put option

The figures 1.1 and 1.2 show, respectively, the graph of CRR implemented for an option with the parameters listed before, and the error of the same binomial model compared with its order of convergence $\frac{1}{n}$.

1.3 Tian Model

The Tian model is an alternative binomial model that achieves improved accuracy by considering second order moments for the underlying asset's distribution. This adjustment results in a faster and smoother convergence path compared to the pure CRR model.

Tian Model Parameters

The parameters for the Tian model are defined using the following intermediate variables:

$$M = e^{r\Delta t}$$
$$V = e^{\sigma^2 \Delta t}$$

The Up factor (u) and Down factor (d) are then given by the following equations:

$$u = \frac{M \cdot V}{2}\left[V + 1 + \sqrt{V^2 + 2V - 3}\right]$$
$$d = \frac{M \cdot V}{2}\left[V + 1 - \sqrt{V^2 + 2V - 3}\right]$$

The risk-neutral probability (p) is defined by the standard arbitrage-free condition:

$$p = \frac{e^{r\Delta t} - d}{u - d}$$

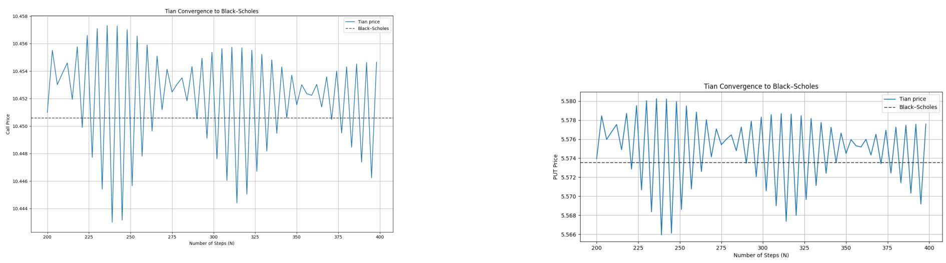


Figure 1.3: Convergence of Tian to BSM price for an European Call and Put option

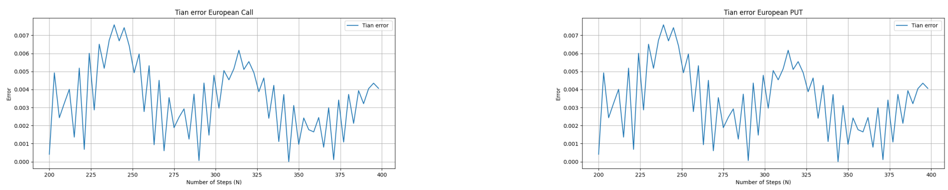


Figure 1.4: Error of Tian model for an European Call and Put option

The figures 1.3 and 1.4 show, respectively, the graph of Tian implemented for an option with the parameters listed before, and the error of the same binomial model.

1.4 Leisen-Reimer Model

One of the latest binomial models is the Leisen-Reimer model. This model has an advantage over the other models in that it exhibits quadratic convergence in the number of time steps (N), at least for European options and American call options, while the other models generally have linear convergence. Therefore, the accuracy is much better. Furthermore, since there are no (or small) oscillations in this model, we can use directly Richardson extrapolation to increase the accuracy even more.

Leisen-Reimer Model Parameters

Firstly, we define:

$$\begin{aligned} a &= e^{r\Delta t}, \\ d_1 &= \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}}, \\ d_2 &= d_1 - \sigma\sqrt{T-t}. \end{aligned}$$

where we recognize d_1 and d_2 from the Black-Scholes equations.

We then introduce the transformed probabilities $\tilde{p}$ and $\bar{p}$:

$$\tilde{p} = B(d_2, N)$$

$$\bar{p} = B(d_2 + \sigma\sqrt{T-t}, N)$$

where B is the inverse of the binomial distribution and N is the number of time steps. We use the Peizer-Pratt method to invert the binomial distribution:

$$p = B(z, n) = \frac{1}{2} \mp \frac{1}{2}\left[\frac{1}{4} - \frac{1}{4}\exp\left\{-\left(\frac{z}{n+\frac{1}{3}}\right)^2\left(n+\frac{1}{6}\right)\right\}\right]^{\frac{1}{2}}$$

where the sign is the sign of z. We substitute p for $\tilde{p}$ and $\bar{p}$ by setting $z = d_2$ and $z = d_2 + \sigma\sqrt{T-t}$ respectively.

Finally, we obtain the u and d factors using the transformed probabilities:

$$u = a \cdot \frac{\tilde{p}}{\bar{p}}$$
$$d = a \cdot \frac{1-\tilde{p}}{1-\bar{p}}$$

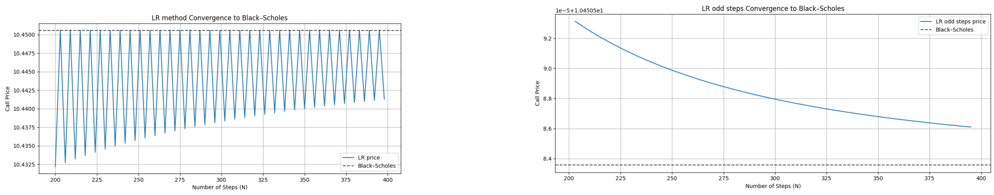


Figure 1.5: Convergence of LR and LR odd steps to BSM price for an European Call option

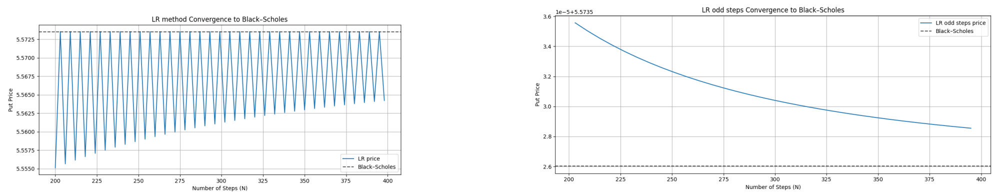
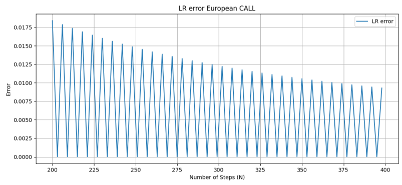


Figure 1.6: Convergence of LR and LR odd steps to BSM price for an European PUT option



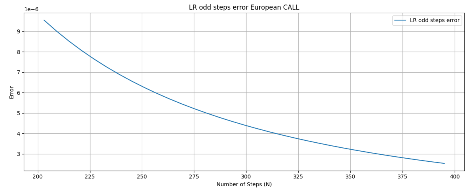


Figure 1.7: Error of LR and LR odd steps model for an European Call option

It can be seen that for both European Call and Put options, Leisen Reimer model shows oscillations as the previous two models. However, considering only the odd nodes, the convergence became smoothed and highly accurate, with errors in the order of 10^{-6}. This is because, since we derive the LR model using the inverse of the binomial distribution, is correct to consider only the odd nodes, in order to have a method that converge accurately to BSM price. Therefore, from now on, we are going to call the Leisen-Reimer model considering only the odd steps as LR.

1.5 Comments

Thus overall, both for European call and put options, CRR model converges oscillating to the BSM price, with the error that decrease as $\frac{1}{n}$. With regards to Tian model, the convergence is not uniform, and there are still oscillations, while for LR model, the graphs show that the approximation is highly accurate without oscillations. The oscillations we see in CRR and Tian are known as *'Sawtooth Effect'*

1.5.1 Convergence

The numerical and graphical evidence strongly support the conclusion that the **Leisen-Reimer model** is superior for pricing European options to the other models without any improvement. It achieves the highest accuracy and stability for a given number of time steps (N), effectively eliminating the problematic sawtooth effect seen in the CRR and Tian models.

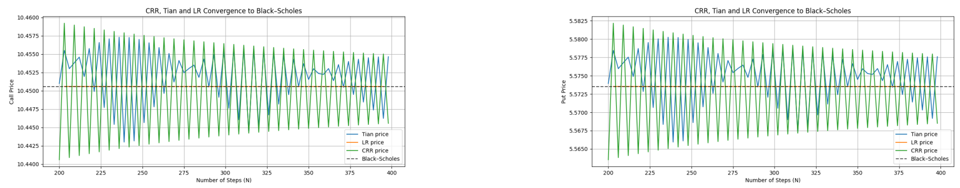


Figure 1.8: Convergence of all the three methods to BSM price for an European call and put option

1.5.2 Errors

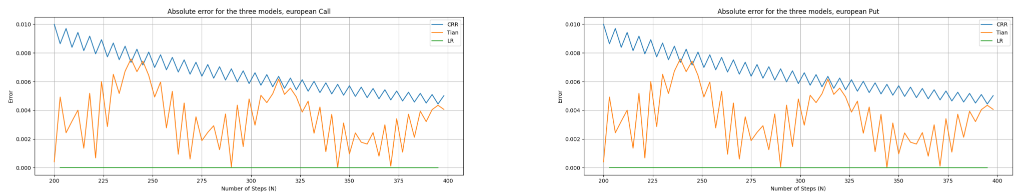


Figure 1.9: Errors of of all the three methods for an European call and put option

These graphs, confirm what we have said before, in fact we can clearly see that the errors for the Leisen-Reimer model are very low from the first steps. We can say that there is no comparison between LR and the other models.

1.6 Sawtooth effect

The sawtoothing is known as the "odd-even effect". This phenomenon results in a very large change in the option price V_n when moving from an odd number of steps (e.g., n) to an even number of steps (e.g., $n+1$). This effect occurs because the final nodes in the lattice move relative to the exercise price (K) of the option, where there is a discontinuity in the option payoff function. The binomial approximation to the normal distribution changes significantly when adding a single step, that results in an overestimation of the option and then in an understimation, this discrepancy leads to the oscillations typical of the model.

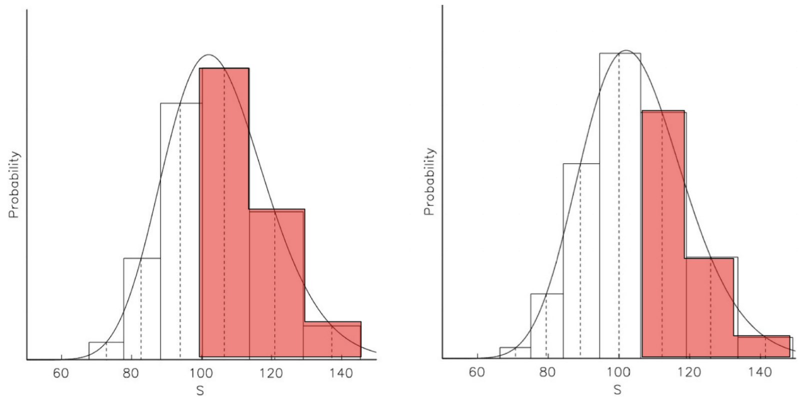


Figure 1.10: Probability density functions for a binomial model with 5 and 6 steps

In order to mitigate the Sawtooth effect, in the following chapter we are going to explore some techniques to smooth the convergence, called *'Smoothing techniques'*. We will also look at some numerical methods, to improve, even more, the convergence of the models we just presented.

Chapter 2

Techniques for Accelerating Convergence of the Binomial Model

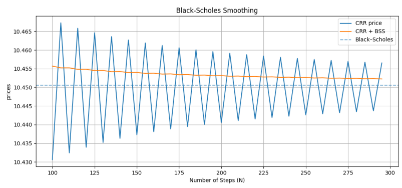
At the end of the previous chapter, we saw that the discontinuity of the option's payoff function at the strike price represents a source of numerical errors and instability. In this chapter, we will see some smoothing techniques and numerical methods that inhibit this effect and make the error converge smoothly as the number of steps increases.

2.1 Black-Scholes Smoothing

The first method we analyze to get rid of the oscillations that alter the convergence of pure binomial models, is the **Black-Scholes smoothing**. As the name of the method suggests, we use the Black-Scholes formula to calculate the values in the nodes of the penultimate layer of the tree, which are the closest to maturity. Then, starting from these new values, we perform the backward calculation as always.

As we said before, the main problem with binomial tree is in the last layer when we compute the option's payoff. In fact, since payoff is a non-smooth function, it causes slow and oscillatory convergence. With this method we don't consider at all the last step, and indeed at each node immediately before maturity we value a very short dated European option, so as to approach to maturity with a smooth function and then eliminating a large part of the error due to the payoff kink.

Since with this model we obtain a much smoother distribution one step prior maturity, the resulting lattice works effectively with an almost smooth terminal condition. The result is the error still behaves like $\frac{1}{n}$, but it approaches the theoretical limit more steadily.



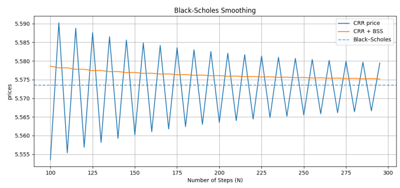
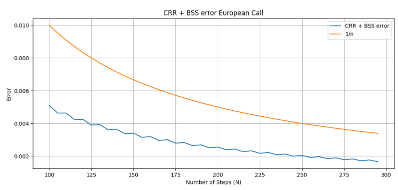


Figure 2.1: Convergence of CRR with Black-Scholes Smoothing to BSM price for a European Call and Put option



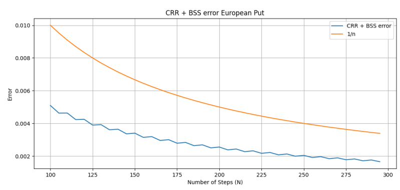


Figure 2.2: Error of CRR with Black-Scholes Smoothing for a European Call and Put option

By looking at figure 2.1, it's clear that the Black-Scholes smoothing considerably improves the speed of convergence of the classic CRR, and the sawtooth effect is practically absent. In figure 2.2, the error no longer oscillates and approaches 0 faster than the pure CRR binomial model.

2.2 Pegging the strike

In the previous chapter, we talked about the sawtooth effect, in particular we underlined how this behavior is mainly due to the strike price falling between two nodes of the tree, causing an overvaluation or undervaluation of the option with each increment in the number of steps. For this reason, the binomial model has a non monotonic convergence. To solve this problem, the researchers have proposed to smooth the tree making the strike coincide with a fixed node of the last layer. It's easy to understand that if the strike price always aligns with the central node of the final layer, when we add more steps the same quantity of nodes are positioned over and under the strike price, and so, we are able to eliminate the alternating pattern. Clearly, in order to *"peg the strike"* we have to consider an even number of steps.

Pegging the Strike Model Parameters

To reach the strike-pegging, we have to modify the up and down factors such that $S_0 u^i d^{n-i} = K$ for some integer i, in particular in order to align the strike with the central node we have to consider $i = \frac{n}{2}$, for even n. The new factors are,

$$u = e^{\sigma\sqrt{\Delta t} + \Delta t \ln \frac{K}{S_0}}, \quad d = e^{-\sigma\sqrt{\Delta t} + \Delta t \ln \frac{K}{S_0}},$$

in this way, the strike price is an actual node at maturity.

Aligning the strike price at a node has a remarkable effect on the convergence. The pricing error continues to behave like $1/n$ but the price converges monotonically as n increases, and the sawtooth effect decreases. Another important benefit of this method is that the implementation requires only a change in the up and down factors and to make sure to consider only even steps.

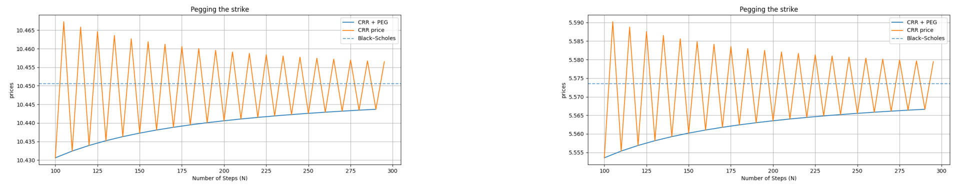


Figure 2.3: Convergence of CRR with Pegging the strike to BSM price for a European Call and Put option

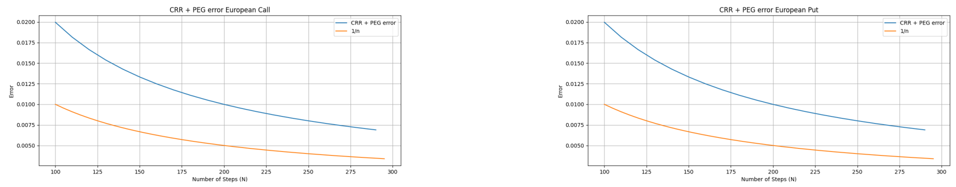


Figure 2.4: Error of CRR with Pegging the strike for a European Call and Put option

From figures 2.3 and 2.4 we see that oscillations are drastically reduced, and both errors and prices approach to theirs limits monotonically.

2.3 Average Smoothing

Averaging Smoothing is an easy way to mitigate the sawtooth effect of a binomial model. The core idea behind this model is to get rid of the oscillations of the CRR binomial model by doing the average between the overvaluations and underestimations of the option, that occur as a consequence of the shifting grid around the strike price as the number of steps increases.

The classic method expects that we compute the price of the option using a tree with N steps and then again with one more step. Then, the final smoothed price is the average of these results,

$$P_{smoothed} = \frac{P_N + P_{N+1}}{2}.$$

This method cancels out a considerable part of the error due to oscillations, and improves the convergence, which is closer to $O(\frac{1}{n^2})$. Another positive feature of this technique is that it doesn't require complex changes to the tree parameters, and so, it can be easily implemented.

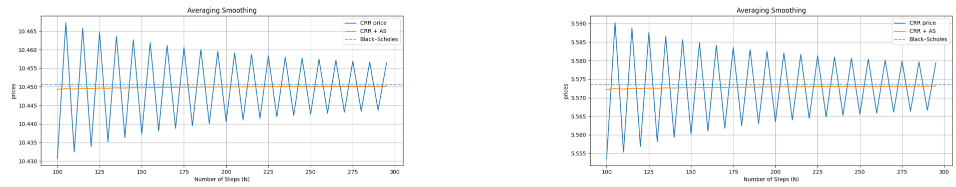


Figure 2.5: Convergence of CRR with Averaging Smoothing to BSM price for a European Call and Put option

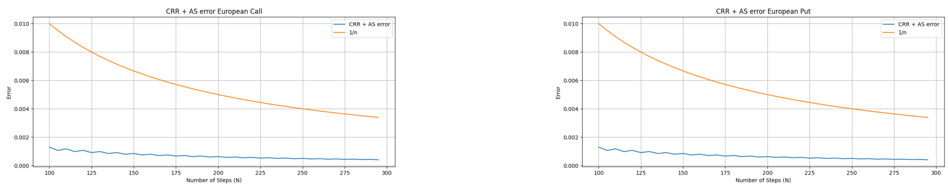


Figure 2.6: Error of CRR with Averaging Smoothing for a European Call and Put option

If we analyze the figure 2.5 we understand the CRR with the averaging smoothing converges to the real value of the option very fast and without any sort of oscillation. In the graph 2.6, we can also see the comparison between the error of CRR with the averaging smoothing and the function $\frac{1}{n}$. The errors of this method are way below $\frac{1}{n}$.

2.4 Richardson Extrapolation

Richardson Extrapolation is a numerical technique used to reduce the error of a numerical method. Suppose we have a numerical method with an error of order p, then

$$F = F(h) + O(h^p) = F(h) + ch^p + O(h^{p+1}).$$

If we consider two values of h, we will have

$$\begin{cases} F = F(h_1) + ch_1^p + O(h_1^{p+1}) \\ F = F(h_2) + ch_2^p + O(h_2^{p+1}), \end{cases}$$

if we multiply the first equation by h_2^p and the second by h_1^p and then subtract them, we obtain

$$F = \frac{h_2^p F(h_1) - h_1^p F(h_2)}{h_2^p - h_1^p} + O(h_2^{p+1}),$$

in this way we have increased the error order, and so the accuracy of the method. In the context of binomial option pricing model we can apply this technique to improve the convergence of the error from $1/n$ to $1/n^2$, in this case we will have that

$$V_{RE} = 2V(2N) - V(N),$$

where V is the price option computed with a classical binomial model.

An important remark to make is that Richardson Extrapolation can be used only with numerical method which have an error which has a smooth, monotonic and asymptotic power series expansion. For this reason, we can't apply Richardson Extrapolation to the CRR without smoothing it.

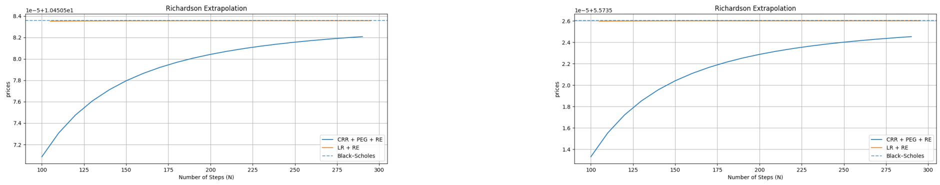


Figure 2.7: Convergence of CRR with Pegging the Strike and Richardson Extrapolation to BSM price for a European Call and Put option

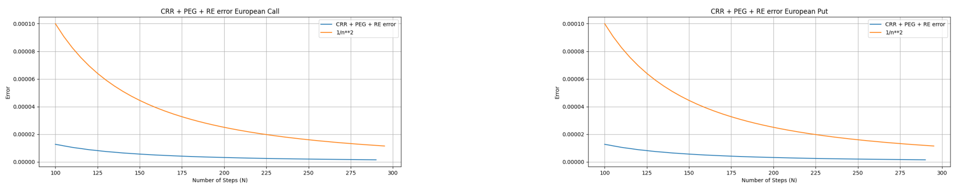


Figure 2.8: Error of CRR with Pegging the Strike and Richardson Extrapolation for a European Call and Put option

In figures 2.7 and 2.8 we see Richardson Extrapolation applied to the CRR smoothed with "Pegging the strike" and to Leisen and Reimer model. We see that the first model approaches the real value of the options very fast, and the error is very low, around the order of 10^{-5}, with a little more than 200 steps.

2.5 Combinations of different smoothing techniques and numerical methods

In this section, we compare different combinations of binomial model and one or more smoothing techniques. Our main goal is to identify which is the most efficient way to "boost" a binomial model, in order to achieve a monotonic and fast convergence to the real value of the option.

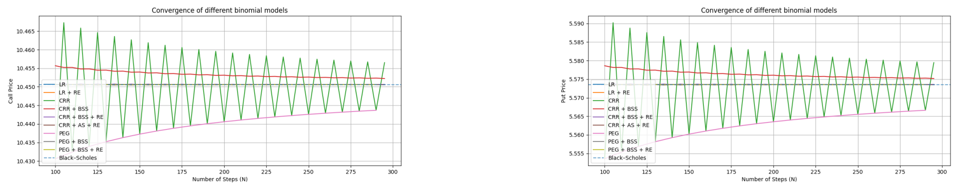


Figure 2.9: Convergence in the different binomial models to BSM price for a European Call and Put option

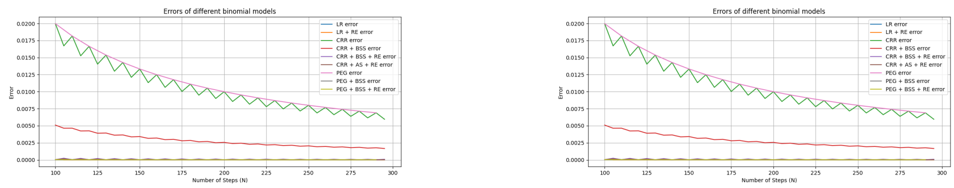


Figure 2.10: Errors in different binomial models for a European Call and Put option

From figures 2.9 and 2.5, we understand that the binomial models that were smoothed and had the Richardson extrapolation applied are the ones who better approach the real value of the option, in fact they are all crowded together around the Black-Scholes value. Also in the graphs where error has been plotted, we see that the same models are very near to zero.

In order to better understand the behavior of some of these models, we compare only the most powerful, as we see in the following figures.

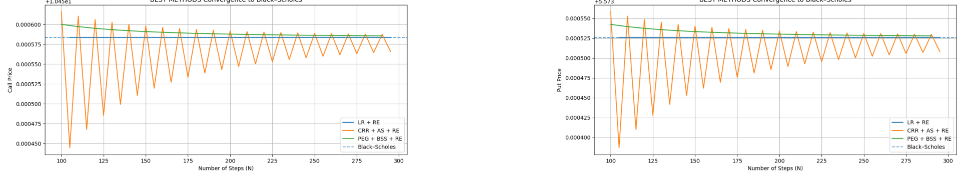
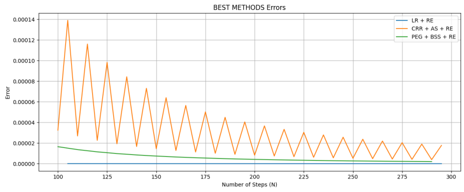


Figure 2.11: Convergence in the best binomial models to BSM price for a European Call and Put option



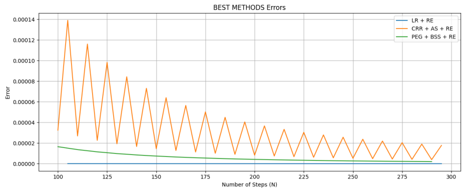


Figure 2.12: Errors of the best binomial models for a European Call and Put option

In particular, from figure 2.5 it's remarkable how the Leisen-Reimer with Richardson Extrapolation and the pegging the strike with Black-Scholes Smoothing and Richardson Extrapolation are the most effective binomial option pricing models among the ones we have seen. Their error is in fact from the first steps of the order of 10^{-6}.

Chapter 3

Exploring Binomial Models for American Option Pricing

In this chapter we want to understand how binomial model are used to price American options, and what are the differences compared to the Europeans. We will see that, also in this case, we can use smoothing and numerical techniques to improve the convergence of the models. In particular, we will focus on the major differences between American Call and Put options.

3.1 Comparing American and European Options

The most relevant difference between American and European options is the possibility for the firsts to exercise the contract before maturity at every possible time interval. Despite this can be seen as a little difference, it changes completely the way we deal with the problem of option pricing. This is the reason why we can't apply all the results we have obtained before with European options to the Americans.

The different possibility of exercise for American options makes impossible to use the Black-Scholes model for them. In fact, for European options the Black-Scholes model calculates the expected payoff at maturity and discounts it back to today, without considering the possibility of early exercise. Since for American options early exercise is possible, we face a mathematical problem called *'Free Boundary Problem'*, that makes pricing an American option an optimal stopping problem. Then, equation as the Black-Scholes cannot be used, we need advanced numerical techniques.

3.2 American Call Option

When we talk about American call option we have to make a clear distinction between those who pay dividends and those who don't. In fact, if the underlying stock pays dividends, a call option holder can receive them only by exercising the contract before the dividends are paid, i.e. by early exercising it. Since we are studying only options whose underlyings don't pay dividends, we won't take this problem in consideration.

Let's consider an American call option on a non-dividend paying stock. In this case, it is never optimal to early exercise the option. Exercising an American call option before maturity destroys its time value, which is the possibility that the stock goes much higher before expiry. Instead, if we keep it, the losses are only limited to the premium we paid to enter the contract but we remain open to unlimited gain. For this reason, if we have an American call option the best choice is to not early exercise it. Then, we can say that an American call option that doesn't pay dividends is practically equal to a European option with the same strike price and maturity.

3.2.1 Benchmark for American Call option

From what we said before, in this case we can use the Black-Scholes-Merton price of the corresponding European option, to compare the efficiency of the different binomial models, and see which one is more fast and accurate. In order to motivate this choice, in our code we computed the Black-Scholes value of the European option with same strike and maturity of our American call, and we also calculate the value of CRR binomial model with a very high number of steps ($N = 10^4$). In fact, we know from theory behind binomial models, that CRR converges to the Black-Scholes as the number of steps increases. The results we obtained are the following:

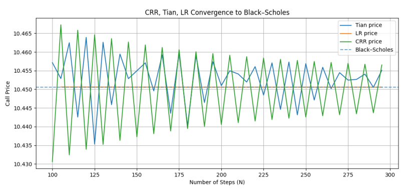
$$\textbf{Black-Scholes-Merton European call price} = 10.450583572185565,$$

$$\textbf{High steps CRR for American call price} = 10.450383602854469,$$

they are practically the same.

3.2.2 Binomial models, Smoothing techniques and Numerical methods

From the previous result, we understood American call options, in the case where no dividends are paid, behave in the same way as European options, so below we will show briefly the graphs of pure binomial models, and then of smoothing techniques and numerical methods applied to them. We don't expect any remarkable difference from what we saw for European options.



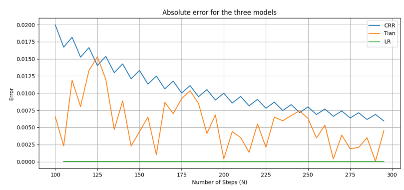


Figure 3.1: Convergence and Errors of CRR, Tian and Leisen-Reimer models with respect to BSM price for an American Call option

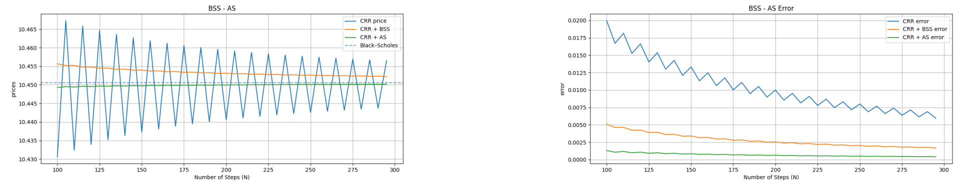


Figure 3.2: Convergence and Errors of CRR with Black-Scholes smoothing and Averaging smoothing with respect to BSM price for an American Call option

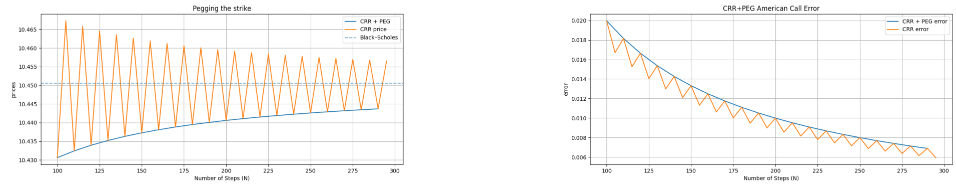


Figure 3.3: Convergence and Errors of CRR with Pegging the strike with respect to BSM price for an American Call option

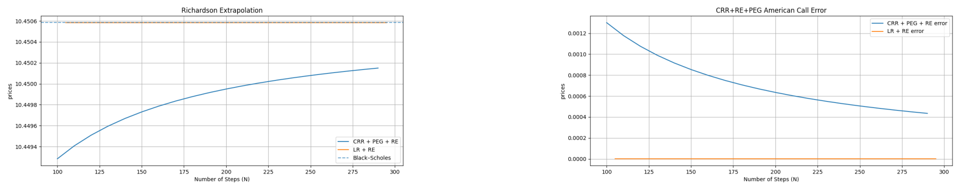


Figure 3.4: Convergence and Errors of CRR with Pegging the strike and Richardson Extrapolazione, and Leisen-Reimer model with RE, to BSM price for an American Call option

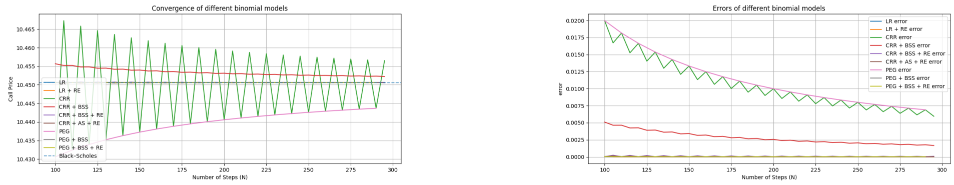


Figure 3.5: Convergence and Errors of different binomial models with respect to BSM price for an American Call option

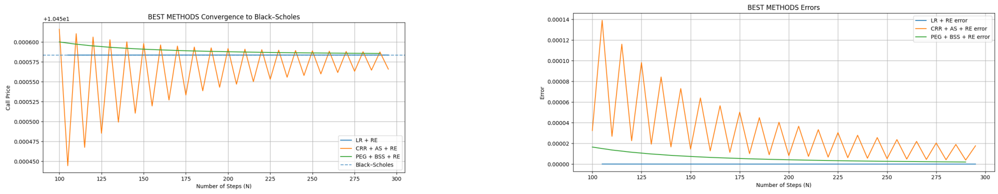


Figure 3.6: Convergence and Errors of the best binomial models with respect to BSM price for an American Call option

By comparing these images with the graphs of European call options, we can see the models behave more or less in the same way, accordingly to what we said earlier about the non-early exercise convenience. In figures 3.5 and 2.5, we confirm that, also for American call options, the most efficient models, both in terms of speed of convergence and error, are the Leisen-Reimer model with Richardson extrapolation and the Pegging the strike with Black-Scholes smoothing and Richardson Extrapolation.

3.3 American Put Option

With regard to American put options, the results are slightly different. Firstly, with this type of options we cannot use as benchmark the price obtained by computing Black-Scholes-Merton formula for European put options.

3.3.1 Benchmark for American Put Option

As stated above, the Black-Scholes-Merton formula for the European put option produces a value we can't use as benchmark. This becomes clear if we compute the CRR model with a high number of steps ($N = 10^4$), and we compare it with BSM value:

$$\textbf{Black-Scholes-Merton European put price} = 5.573526022256971,$$

$$\textbf{High steps CRR for American put price} = 6.090295412872714.$$

The reason of this remarkable difference between the two prices is a consequence of the fact that the BSM formula does not consider the *'early exercise premium'* and therefore undervalues the American put. Therefore,for American put options we will use as benchmark the price computed by CRR with high steps.

3.3.2 American Put Option Binomial models

In this section we are going to compute the prices and analyze the errors of the Cox-Ross-Rubestein, Tian and Leisen-Reimer method.

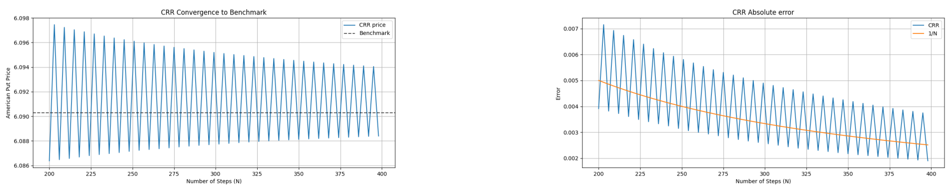


Figure 3.7: Convergence and error of CRR to Benchmark price for an European Put option

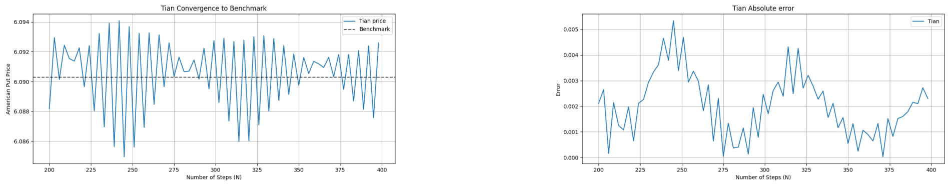


Figure 3.8: Convergence and error of Tian to Benchmark price for an European Put option

By looking at figure 3.8 we notice that for CRR and Tian the convergence to the benchmark is essentially the same as for European options.

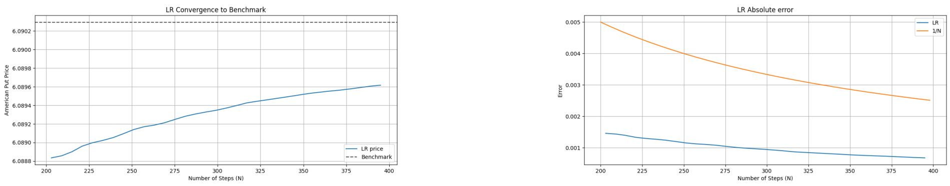


Figure 3.9: Convergence and error of LR to Benchmark price for an European Put option

Figure 3.9 shows that Leisen-Reimer model is less accurate for American put options with respect to the European options that we have mentioned before. Indeed, the error is in the order of 10^{-3}, while for European call was in the order of 10^{-6}, accordingly to the fact that for American options the error of LR model is around $\frac{1}{n}$, due to the early exercise possibility.

3.3.3 Smoothing and Numerical methods

Black-Scholes smoothing for American Put options

To implement the Black–Scholes smoothing, a problem has been encountered. In fact, since the Black–Scholes–Merton formula does not produce the correct price for an American put option, it is not possible to construct the Black–Scholes smoothing in the same way as for other types of options. However, we observe that it is possible to use binomial models with a high number of steps instead of the BSM formula to modify the penultimate layer of the binomial tree. Nevertheless, this approach makes the smoothing computationally significantly expensive. So we tried to use faster methods such as LR with Richardson extrapolation, which converge faster to the real option value, but despite this the computational cost remains very high. Therefore, we have not implemented the Black–Scholes smoothing method for American put options.

Comparison of different techniques to improve convergence

Without Black-Scholes smoothing, we used the other techniques available in order to reduce the sawtooth effect and to improve the convergence. In the following figures, are presented the results we obtained by combining the fundamental binomial models with the different smoothing and numerical techniques.

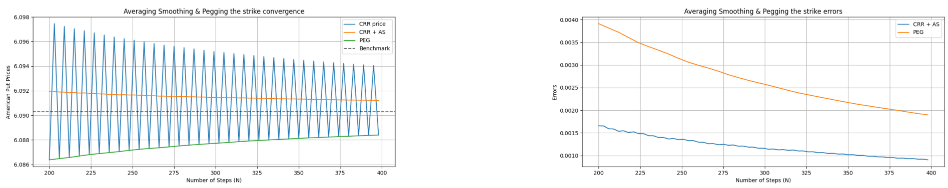


Figure 3.10: Convergence and error of Pegging the strike with averaging smoothing price for an European Put option

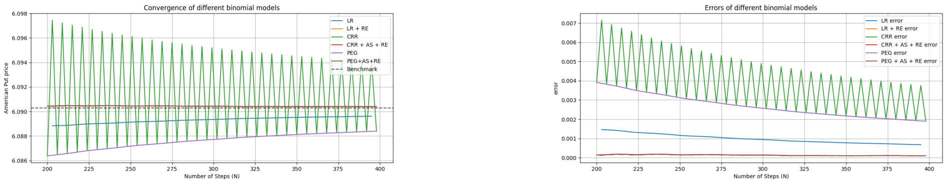


Figure 3.11: Convergence and error of different Binomial Models for an European Put option

From the last figure 4.2, we notice that the best binomial models to price American put options are the Leisen-Reimer with Richardson Extrapolation, the CRR with Averaging smoothing and Richardson extrapolation and the Pegging the strike with Averaging Smoothing and Richardson Extrapolation.

Chapter 4

A special case

In this chapter, we will test our application on an option with different parameters,obtaining some slightly different results. Then, we will comment our outcomes and give a possible answer to the differences found between the graphs of the two options.

Parameters

We operate under the assumptions of the Black-Scholes model, including constant volatility and a risk-free environment. In the following lines we will show the set of financial parameters we will use along this chapter to price the options, and compare them. The parameters used are:

- **Underlying Price** (S): $S = 100$
- **Strike Price** (K): $K = 110$
- **Risk-Free Interest Rate** (r): $r = 2.0\%$ (or 0.02)
- **Time to Maturity** (T): $T = 0.5$ years (or 6 months)
- **Volatility** (σ): $\sigma = 40.0\%$ (or 0.40)

The time step (Δt) for the binomial lattice is defined by the number of time steps (N) used:

$$\Delta t = \frac{T}{N}.$$

4.1 Binomial models, Smoothing techniques and Numerical methods

In this section, we will visualize the behavior of binomial models implemented with smoothing techniques and numerical methods. To obtain this graphs we used exactly the same Python code we used in the previous chapter only changing the parameters. Despite this, as we will see in the following images, the models seem to have different trends with respect to what we analyzed in the previous chapters.

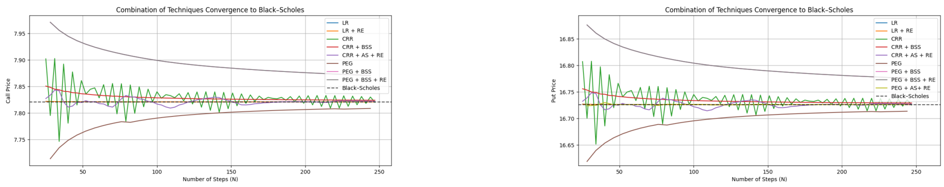


Figure 4.1: Convergence of different Binomial models for a European Call and Put option

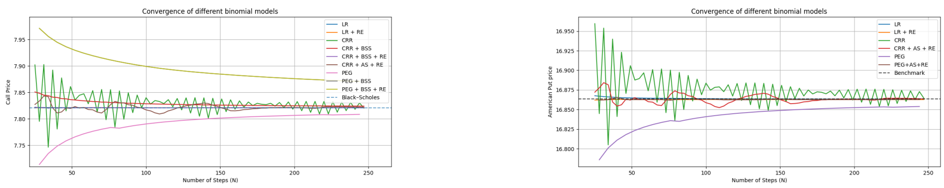


Figure 4.2: Convergence of different Binomial Models for an American Call and Put option

First of all, we notice a different behavior of the CRR binomial method, this is probably due to the fact that in the previous option the initial price of the stock and the strike price were the same, instead in this case they are different, so the probability that the strike price falls in the central node or in the centre between two nodes is very low. For this reason, the CRR looses a big part of its symmetry. This problem generates consequences in other models as well.

- **Averaging smoothing:** Since this smoothing technique does the average between the prices of the option computed by using a tree with N steps and another one with $N+1$ steps, it conserves the error due to the distance between the strike price and the nearest node.

In binomial models the volatility is the measure of the range between the *'up'* node and *'down'* node, in fact

$$u = e^{\sigma\sqrt{\Delta t}}.$$

So, for the option we considered at the beginning since the underlying stock had 20% of volatility the grid of the tree was fine, conversely in the option we are looking at, the volatility is doubled, so the nodes are far one from the other. A more spaced tree means that the discretizing error of the binomial model is bigger for the same number of step.

- **Pegging the strike:** To peg the strike means changing the parameters for the strike price K to be exactly a node. But in the case we are currently considering, since the nodes are away from each other, it's more difficult to force the tree in a symmetrical configuration and this creates a remarkable error. From figure 4.1, if we look at the brown line, we see that the PEG converges but at a slower rate and with worse accuracy than for the other option.

- **Richardson Extrapolation:** To apply Richardson Extrapolation we have to know exactly how the error behaves. Then, if the shifted grid generates an irregular oscillation the Richardson Extrapolation could fail. This is probably what happened to the Pegging the strike model with Black-Scholes Smoothing. In fact, if we look at figure 4.1, the black line, which refers to the model in question, is exactly around the convergence limit, but if we apply to it Richardson Extrapolation (purple line) its speed of convergence decreases considerably.

Conclusions

In the first three Chapters, we tested with our Python code the behavior of binomial models, smoothing techniques and numerical methods for an option with a certain set of parameters. At the beginning, we showed the pure binomial models, then we tried to implement some methods with the aim of improving the convergence, overcoming the limitations of those models and in general bettering their efficiency. We obtained clear results coherent with the theoretical aspects of binomial option pricing theory. We did this work first of all with a European-type option, then with an American one. For the American call, we saw that the results were the same as those of the European option; instead, we encountered some difficulties with the American put, in particular in implementing the Black–Scholes smoothing, whose computational cost was too high.

Then, in the last chapter, we used our Python code to price, with the same models used before, an option with different parameters, including a higher volatility. This change led to results that were a little bit different from what we saw before and from what we expected. We justified some of these differences in Chapter 4 using theoretical considerations. By the way, we have to say that several of these differences can be attributed to the fact that we implemented some of these models using complex and iterative structures, which propagate the error and alter our results. Maybe, a problem of this type can be solved by using more efficient coding structures to implement the models.