Monte Carlo Based Numerical Pricing of Multiple Strike-Reset Options

Stavros Christodoulou

Linacre College
University of Oxford



MSc Thesis
Trinity 2011

Contents

List of figures **ii**

Introduction **2**

1 Strike Reset Options **4**
1.1 Financial Market 4
1.2 Description 4
1.3 Strike Reset Put Option 6
1.3.1 Strike Reset Put Option Valuation Algorithm 10
1.3.2 Strike Reset Put Option Forward Valuation Algorithm 12
1.4 Convergence and Errors 12
1.5 Estimating the Greeks 13
1.5.1 Likelihood Ratio Method 14
1.5.2 Pathwise Sensitivities Approach 17

2 Implementation and Numerical Results **22**
2.1 Strike Reset Put Option Valuation Algorithm and Results 22
2.2 Strike Reset Put Option Forward Valuation Algorithm and Results 25
2.2.1 Greeks 27
2.3 Strike Reset Put Option General Properties 31

3 Conclusion and Final Remarks **35**

4 Appendix A **36**

5 Appendix B **40**

List of Figures

2.1 Convergence of the Forward Valuation Algorithm 26
2.2 Value of the Strike Reset Put option with different number of strike resets . 27
2.3 Convergence for Delta using Likelihood Ratio Method for a fixed $M = 30$. 28
2.4 Convergence for Delta using Pathwise Sensitivities Approach for a fixed $M = 30$ 28
2.5 Delta as a Function of S_0 29
2.6 Convergence for Gamma using Likelihood Ratio Method for a fixed $M = 30$ 29
2.7 Convergence for Vega using Pathwise Sensitivities Approach for a fixed $M = 30$ 30
2.8 Convergence for Rho using Pathwise Sensitivities Approach for a fixed $M = 30$ 31
2.9 Value of the Strike Reset Put option with different number of shout opportunities 33
2.10 Value of the Strike Reset Put option with different maturities 34
2.11 Value of the Strike Reset Put option with different initial strike values . . . 34

.

Acknowledgements

I would like to thank my supervisor Greg Gyurkò for all the help, support and guidance that he provided in the duration of this dissertation, which enabled me to develop an understanding of this subject.

Introduction

In the last few years the complexity of some contracts offered by many financial markets has increased a lot. A common but complex financial contract is the Shout option, which grants the allowance to the holder to alter certain features of this contract, according to some specific rules. An example of a Shout option, and the one analysed in this thesis, is the Strike Reset option, which grands the option holder the right to reset the strike of the option a predetermined number of times before maturity.

The pricing and hedging of these contacts is much more complex than the simple European options, and for most of them there does not exist a closed form formulae. As a result the valuation of these options is mainly done using multiple layer binomial trees, using Finite Difference methods or using Monte-Carlo techniques. [6] and [5] analyse the Strike Reset Put option on a single underlying asset and consider optimal reset policies using a PDE approach. Moreover [19] uses Finite Difference methods to price Strike Reset Put options with more complex restrictions in the reset features.

In this thesis we use Least Squares Regression method, which was introduced by Longstaff and Schwartz, to price the Strike Reset Put option. Moreover we use Pathwise Sensitivities Approach and Likelihood Ratio Method to obtain the Greeks of this Strike Reset option within the Monte Carlo framework. Note that this thesis is an extension of a previous thesis, [20], submitted by Yudaken, where in this thesis we handle the increased complexity due to the multiple strike resets that the holder of the option has.

Note that we present all theoretical and numerical results for the specific case of Strike Reset Put option. These results can be very easily extended to similar Strike Reset options, for example the Strike Reset Call option and the Strike Reset Asian Call and Put options.

This thesis is structured as follows. In chapter 1, we present a description of the Strike Reset option. Then we extend the Least Squares approach to price the Strike Reset Put option. Moreover we present theoretical results for estimating the sensitivities of this option using Likelihood Ratio Method and Pathwise Sensitivities Approach. In chapter 2, we

present numerical results for modelling the Strike Reset Put option. This thesis ends by a conclusion and a suggestion for possible extensions.

Chapter 1

Strike Reset Options

1.1 Financial Market

The general set-up involves a finite, discrete-time horizon $\mathcal{T} = \{t_1, ..., t_M\}$ and a Markov stochastic process S on a complete filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$, where $\mathbb{F} = (\mathcal{F}_t)_{t\in\mathcal{T}}$ is the augmented filtration generated by the price process $(S_t)_{t\in\mathcal{T}}$. The sample space Ω is the set of all possible realizations of the stochastic process S, over $\mathcal{T}$. Finally $\mathbb{P}$ is the natural probability measure. Note that I assume the existence of an equivalent martingale measure $\mathbb{Q}$, consistent with no-arbitrage pricing.

1.2 Description

A simple Strike Reset option enables the holder to alter the initial strike K_0, to K_1, once at any time during the life of the option. We consider contracts for which the value of the new strike K_1, is set to be equal to the value of the underlying asset S_t, where t is the time of the reset. In this case the holder of the contract has the right to decide when to take the ownership of an at the money option. Note that the exercise of a reset right is usually called shouting. More generally, for Strike Reset options, the holder has the right to alter the the initial strike K_0 a maximum of L times at any time before maturity. Hence the Strike Reset option gives the owner the right to exchange his current contract for another contract, which provides less flexibility but its probably better than the previous contract. Note that the value of the option is determined by the optimal shouting policy, where any deviation from that policy decreases the value of this option.

A Strike Reset option is a contract defined by the following:

- Given that $t = 0$ is the time when the contract is written, then $[0, T]$ is the time interval on which the contract takes effect.
- S_t, which is the price of the underlying asset at time t.
- K_0, which is the initial strike of the option.

- L, which is the initial number of strike reset rights that the holder of the contract has. Any right can be exercised during $[0, T]$, with at most one exercise right per day.
- The new strike, if an exercised right is used, is set to be the prevailing stock price at the moment of shouting.
- $g\left(S_T, ..., S_0, K_0\right)$, which is the payoff of the strike reset option.

A common Strike Reset option, and the one extensively used in this thesis, is the Strike Reset Put option. This option has a payoff given by

$$g(S_T, K_0) = \begin{cases} (K_0 - S_T)^+ & \text{if no reset right was used} \\ (S_t - S_T)^+ & \text{if t is the last time that a strike reset right was used} \end{cases}$$

Another common Strike Reset option is the Strike Reset Asian Put option option which has the following payoff

$$g(S_T, ..., S_0, K_0) = \begin{cases} \left(K_0 - \bar{S}\right)^+ & \text{if no reset right was used} \\ \left(S_t - \bar{S}\right)^+ & \text{if t is the last time that a strike reset right was used} \end{cases}$$

where

$$\bar{S} = \frac{1}{M} \sum_{i=1}^{M} S_{t_i}$$

and $0 = t_0 < t_1 < ... < t_M = T$.

It's worth noting here that the holder of the Strike Reset Put option will never shout and reset the strike at maturity, since this will result to a zero payoff, whereas the holder of the Strike Reset Asian Put option may choose to shout at maturity provided that the underlying price at expiry S_T is greater than $\bar{S}$. This although depends on availability of a strike reset right and the value of the strike at maturity as well.

An interesting limiting case of the Strike Reset Put option is when the number of strike reset rights L tends to infinity and any right can be exercised on $[0, T]$ without any further restrictions. For this particular case the Strike Reset Put option becomes a fixed strike look-back option with payoff

$$g(S_T, K_0) = M_T - S_T$$

where M_T denotes the maximum value that was attained by the underlying price process during the life of the option. This is the case because the holder of the Strike Reset Put option will reset the strike, whenever the current value of the price process is greater than the current value of the strike.

In this thesis the valuation of Strike Reset Options will be done using Monte Carlo methods. Monte Carlo methods are very popular in finance, mainly because they are fairly simple to implement and since they allow the treatment of problems, when the dimension of the problem increases. However the main disadvantage of Monte Carlo methods, compared to other methods like Finite Difference methods, is that they have larger errors due to the variance and hence usually for low dimensions Finite Difference methods are preferred over Monte Carlo methods, but since Monte Carlo methods scale better with the dimension of the problem, they are usually preferred as the dimension of the problem increases.

Comparing the Strike Reset option with the European and Bermudan options we can see that the payoff is received at maturity, but we are actually faced with a Bermudan style optionality, since the payoff is determined by when the holder of the option is going to reset the strike. The valuation of the Strike Reset option leads to an optimal stopping problem, where it is necessary to investigate the optimal shouting policy and determine when it is optimal to shout and reset the strike of the option.

The early exercise feature is a great challenge for Monte Carlo methods. For Bermudan options several approaches have been discussed in the literature. Some authors approximate the transition density function and others focus directly on the approximation of the conditional expectation which is involved in the Dynamic Programming Formulation. In this thesis we approximate the conditional expectation using least squares regression methods for valuing Strike Reset Put option.

1.3 Strike Reset Put Option

We will now extend the least squares regression method to value the Strike Reset Put option. Consider a Strike Reset Put option with L strike reset rights, which can be used at any time during $[0, T]$, but at most one per day. In order to approximate the value of this contract we assume that the holder of the option can only exercise a strike reset right on a finite set of times $\{t_1, ..., t_M\} = \mathcal{T}$, but as M increases this value should approach the value of the Strike Reset Put option. Let $V_i^l(s, k)$ denote the price of the Strike Reset Put option at time t_i with l reset rights left, where $t_i \in \mathcal{T}$ and $l \in \{0, ..., L\}$. Moreover s denotes the underlying price and k the most recent value of the strike, just before any possible update at time t_i, which is maybe different than the initial strike K_0. Finally for simplicity we assume that $t_{i+1} - t_i = \Delta t$.

At the expiration date the holder of the Strike Reset Put option will not exercise a strike reset right, as exercising one will force the strike to become equal with the underlying asset price, and hence the payoff of the Put option will be zero and the option will expire worthless. On the other hand if the holder of the option does not reset the strike at maturity, the payoff will simply be the payoff of the Put option with the most recent strike value. At any exercise time t_i before expiration, the holder of the option must decide weather to exercise immediately, and use one of his rights to reset the strike, or continue to hold the option until the next exercise date, where he will face the same decision problem.

Similarly with the Bermudan option and according to [3], the value of Strike Reset Put option is maximised if the holder of the option resets the strike when the exercise value, the value of the option with one reset right less, is greater or equal than the continuation value, the value of the option with the same number of reset rights. Hence in order to value the L-Strike Reset Put option, where L indicates that initially the holder has L rights to reset the strike, we consider the following Dynamic Programming Formulation:

$$\begin{aligned} V_M^L(s,k) &= V_M^{L-1}(s,k) = ... = V_M^0(s,k) = g_M(s,k) \\ V_{i-1}^l(s,k) &= \max\left\{ e^{-r\Delta t}\mathbb{E}\left[V_i^l(S_i,k)\,|S_{i-1}=s \right], e^{-r\Delta t}\mathbb{E}\left[V_i^{l-1}(S_i,S_{i-1})\,|S_{i-1}=s \right] \right\} \end{aligned} \tag{1.1}$$

where we assumed a constant interest rate $r_t = r$. The additional complexity of this contract compared to the Bermudan option, is due to the fact that the value of the Strike Reset Put option at any time t, depends not only on S_t but on the value of the strike K as well, which is not necessarily determined by the value of S_t as a strike reset right may have been used before t. As a result, to overcome this complexity we extend the state space from the set of all possible realizations of the stochastic process $(S_t)_{t\in\mathcal{T}}$, to the the set of all possible realizations of the stochastic process $(S_t, K_t)_{t\in\mathcal{T}}$. Now the value of the Strike Reset option can be determined by the current values of S_t and K_t.

We are interested in $V_0^L(S_0, K_0)$, hence in order to proceed with the Dynamic Programming Formulation, we need to calculate the conditional expectations that arise in equation 1.1. We denote the continuation value of the Strike Reset option by

$$V_{i-1}^{l,C}(s,k) = e^{-r\Delta t}\mathbb{E}\left[V_i^l(S_i,k)\,|S_{i-1}=s \right] \tag{1.2}$$

and the shouting value by

$$V_{i-1}^{l,SH}(s,k) = e^{-r\Delta t}\mathbb{E}\left[V_i^{l-1}(S_i,S_{i-1})\,|S_{i-1}=s \right] \tag{1.3}$$

Now since the value of the Strike Reset option depends only on (S_t, K_t) is hence a Markov process we can extend the least squares regression framework to estimate the continuation value and the shouting value of this option. Using two sets of basis functions, one univariate

and one bivariate set of functions, we assume that the conditional expectations that arise in the Dynamic Programming Formulation can be approximated using a linear combination of a set of basis functions as follows:

$$\begin{aligned}\mathbb{E}\left[V_i^l\left(S_i,k\right)|S_{i-1}=s\right] &\approx \sum_{r=1}^{R_2}\beta_{ir}^l\phi_r\left(s,k\right) \text{ for } l\in\{1,...,L-1\}\\ \mathbb{E}\left[V_i^L\left(S_i,k\right)|S_{i-1}=s\right] &\approx \sum_{r=1}^{R_1}\beta_{ir}^L\varphi_r\left(s\right) \text{ for } l=L\end{aligned} \tag{1.4}$$

where $\varphi_1,\varphi_2,...,\varphi_{R_1}$ is a set of R_1 univariate basis functions, with the corresponding regression coefficients $\beta_{i1}^L,\beta_{i2}^L,...,\beta_{iR_1}^L$. Similarly $\phi_1,\phi_2,...,\phi_{R_2}$ is a set of R_2 bivariate basis functions, with the corresponding regression coefficients $\beta_{i1}^l,\beta_{i2}^l,...,\beta_{iR_2}^l$. Note that we have two sets of basis functions since for the case that no right has been used, the current strike of the option will be the initial strike, and hence there is no need to use a bivariate function in order to approximate the conditional expectation.

An example of possible set of basis functions is the Lauguere polynomials:

$$\begin{aligned}\varphi_1\left(s\right) &= e^{-s/2}\\ \varphi_2\left(s\right) &= e^{-s/2}\left(1-s\right)\\ \varphi_3\left(s\right) &= e^{-s/2}\left(1-2s+s^2/2\right)\\ &\vdots\\ \varphi_{n+1}\left(s\right) &= e^{-s/2}\frac{e^s}{n!}\frac{d^n}{ds^n}\left(s^ne^{-s}\right)\end{aligned} \tag{1.5}$$

As mentioned in [16], other possible sets of basis functions include the Chebyshev, Legendre, Hermite polynomials and much more. Numerical results although show that even trigonometric series and simple powers of the state variable give accurate results.

An example of a possible set of bivariate basis functions is the following:

$$\begin{aligned}\phi_1\left(s,k\right) &= 1\\ \phi_2\left(s,k\right) &= s\\ \phi_3\left(s,k\right) &= k\\ \phi_4\left(s,k\right) &= s^2\\ \phi_5\left(s,k\right) &= k^2\\ \phi_6\left(s,k\right) &= sk\\ &\vdots\end{aligned} \tag{1.6}$$

Coefficients β_{ir}^l can be estimated by orthogonal projection, which is the least squares minimization of

$$\mathbb{E}\left[\left(\mathbb{E}\left[V_i^l\left(S_i,K\right)|S_{i-1},K\right]-\sum_{r=1}^{R_2}\beta_{ir}^l\phi_r\left(S_{i-1},K\right)\right)^2\right]$$

Differentiating with respect to β_{ir}^l for $r=1,...,R_2$, and setting the result equal to zero gives

$$\mathbb{E}\left[\Phi\left(S_{i-1},K\right)^T\left(\Phi\left(S_{i-1},K\right)\beta_i^l-\mathbb{E}\left[V_i^l\left(S_i,K\right)|S_{i-1},K\right]\right)\right]=0$$

where $\Phi(x,y)=(\phi_1(x,y),...,\phi_{R_2}(x,y))$ and $\beta_i^l=\left(\beta_{i1}^l,...,\beta_{iR_2}^l\right)^T$.

Hence

$$\begin{aligned}\mathbb{E}\left[\Phi(S_{i-1},K)^T\Phi(S_{i-1},K)\right]\beta_i^l&=\mathbb{E}\left[\Phi(S_{i-1},K)^T\mathbb{E}\left[V_i^l\left(S_i,K\right)|S_{i-1},K\right]\right]\\&=\mathbb{E}\left[\mathbb{E}\left[\Phi(S_{i-1},K)^TV_i^l(S_i,K)|S_{i-1},K\right]\right]\end{aligned}$$

where the second equality arises since $\Phi(S_{i-1},K)^T$ is measurable with respect to S_{i-1} and K.

Now using the tower property of the conditional expectation we obtain that

$$\begin{aligned}\beta_i^l&=\mathbb{E}\left[\Phi(S_{i-1},K)^T\Phi(S_{i-1},K)\right]^{-1}\mathbb{E}\left[\Phi(S_{i-1},K)^TV_i^l\left(S_i,K\right)\right]\\&=:\left[B_{\Phi\Phi,i}^l\right]^{-1}\left[B_{V\Phi,i}^l\right]\end{aligned}\tag{1.7}$$

Note that the theoretical value of β_i^l is given by equation 1.7, but in practice it has to be estimated. Using Monte Carlo simulation we obtain that

$$\hat{B}_{\Phi\Phi,i}^l=\frac{1}{N\tilde{K}}\sum_{j=1}^{N\tilde{K}}\Phi\left[(S_{i-1},K)^j\right]^T\Phi\left[(S_{i-1},K)^j\right]$$

$$\hat{B}_{V\Phi,i}^l=\frac{1}{N\tilde{K}}\sum_{j=1}^{N\tilde{K}}\Phi\left[(S_{i-1},K)^j\right]^T\hat{V}_i^l\left[(S_i,K)^j\right]$$

which will be used to estimate β_i^l for $l=0,1,...,L-1$. Note that N is the number of trajectories that we have simulated and $\tilde{K}$ is the number of possible strike values that are used in this approximation. So for every possible asset price $S_i^j\in\Sigma_i:=\left\{S_i^1,...,S_i^N\right\}$ there are $\tilde{K}$ possible strike values, $K\in\Upsilon:=\{K_0,...,K_{max}\}$[1].

[1]Note that one can choose a grid for possible strike values with a constant strike step $K_{i+1}-K_i$ although in this thesis we use a grid of variable strike step. Moreover for K_{max} we use the maximum realised simulated price, although one can choose K_{max} using a confidence interval or even choose a large enough value of K_{max} and reduce it up to the point that the value of the approximation is still accurate enough. Finally another way of choosing a set of possible strikes $K_0,...,K_{max}$ is suggested in the conclusion of this thesis as a possible extension of this method.

As a result we have a grid of values $\Sigma_i \times \Upsilon$, where $(S_i, K)^j \in \Sigma_i \times \Upsilon$, $\forall j = \left\{1, ...N\tilde{K}\right\}$. Finally $\hat{V}_i^l$ is is the approximation for the value of the Strike Reset option with l available strike reset rights at time t_i.

Following exactly the same procedure, using least squares minimization for the univariate basis functions, we obtain that the coefficients β_i^L are equal to

$$\begin{aligned} \beta_i^L &= \mathbb{E}\left[\Phi\left(S_{i-1}\right)^T \Phi\left(S_{i-1}\right)\right]^{-1} \mathbb{E}\left[\Phi\left(S_{i-1}\right)^T V_i^L\left(S_i\right)\right] \\ &=: \left[B_{\Phi\Phi,i}^L\right]^{-1}\left[B_{V\Phi,i}^L\right] \end{aligned} \tag{1.8}$$

where $\Phi(x) = (\varphi_1(x), ..., \varphi_{R_1}(x))$ and $\beta_i^L = \left(\beta_{i1}^L, ..., \beta_{iR_1}^L\right)^T$.

Note that equation 1.8 gives the theoretical values of β_i^L which in practice has to be estimated. Using Monte Carlo simulation we obtain

$$\hat{B}_{\Phi\Phi,i}^L = \frac{1}{N}\sum_{j=1}^{N} \Phi\left(S_{i-1}^j\right)^T \Phi\left(S_{i-1}^j\right)$$

$$\hat{B}_{V\Phi,i}^L = \frac{1}{N}\sum_{j=1}^{N} \Phi\left(S_{i-1}^j\right)^T \hat{V}_i^L\left(S_i^j\right)$$

which will be used to estimate β_i^L. Now we are in a position to implement our first algorithm for pricing the Strike Reset Put option.

1.3.1 Strike Reset Put Option Valuation Algorithm

1. Simulate N independent trajectories, each consisting of M time steps.
2. Determine a set of possible strike values Υ, in order to create a grid of values $\Sigma_i \times \Upsilon$
3. For each of the $N\tilde{K}$ terminal nodes set
   - $\hat{V}_M^L\left(S_M, K_0\right) = g_M\left(S_M, K_0\right)$
   - $\hat{V}_M^l\left(S_M, K\right) = g_M\left(S_M, K\right)$ for $0 \le l \le L-1$
4. For each time step $i = M, M-1, ..., 2$ perform the following steps
   (a) Estimate β_i^L and β_i^l using
      - $\hat{\beta}_i^L = \left[\hat{B}_{\Phi\Phi,i}^L\right]^{-1}\left[\hat{B}_{V\Phi,i}^L\right]$
      - $\hat{\beta}_i^l = \left[\hat{B}_{\Phi\Phi,i}^l\right]^{-1}\left[\hat{B}_{V\Phi,i}^l\right]$, $0 \le l \le L-1$.
   (b) Evaluate the continuation value for each grid point $(S_{i-1}, K)^j \in \Sigma_{i-1} \times \Upsilon$, using

- $\hat{V}_{i-1}^{l,C}\left[(S_{i-1},K)^j\right] = e^{-r\Delta t}\Phi(S_{i-1}^j,K)^T\hat{\beta}_i^l$ where $0 \leq l \leq L-1$
- $\hat{V}_{i-1}^{L,C}\left(S_{i-1}^j\right) = e^{-r\Delta t}\Phi(S_{i-1}^j)^T\hat{\beta}_i^L$

(c) Evaluate the shouting value[2] for each grid point $(S_{i-1},K)^j \in \Sigma_{i-1}\times\Upsilon$, using

- $\hat{V}_{i-1}^{l,SH}\left[(S_{i-1},K)^j\right] = e^{-r\Delta t}\Phi(S_{i-1}^j,S_{i-1}^j)^T\hat{\beta}_i^{l-1}$ where $1 \leq l \leq L$

(d) Calculate the value function for each grid point

- $\hat{V}_{i-1}^{l}\left[(S_{i-1},K)^j\right] = \max\left\{\hat{V}_{i-1}^{l,C}\left[(S_{i-1},K)^j\right], \hat{V}_{i-1}^{l,SH}\left[(S_{i-1},K)^j\right]\right\}$, $0 \leq l \leq L-1$
- $\hat{V}_{i-1}^{L}\left(S_{i-1}^j\right) = \max\left\{\hat{V}_{i-1}^{L,C}\left(S_{i-1}^j\right), \hat{V}_{i-1}^{L,SH}\left(S_{i-1}^j\right)\right\}$, for $l = L$

5. Average over the N trajectories $\hat{V}_1^L\left(S_1^j\right)$ and discount back to zero in order to obtain an estimate for V_0^L using

$$\hat{V}_0^L = \frac{e^{-r\Delta t}}{N}\sum_{j=1}^{N}\hat{V}_1^L\left(S_1^j\right)$$

where $\hat{V}_i^L\left(S_i^j\right)$ is our estimate for the price of the Strike Reset Option at time t_i for the j^{th} trajectory and $V_i^{0,SH}\left[(S_i,K)^j\right] := 0 \ \forall j = 1,...,N$ and $i = 2,...,M$.

Note that this implementation yields an estimate for the price of the option and for the regression coefficients β_i^l. This Valuation Algorithm for the Strike Reset Option, similarly with the Bermudan option and according to [8], since we use identical paths in order to calculate both, the regression coefficients and the option price, and we look at the whole trajectories, this may yield to option values which are high biased. As a result in order to avoid this we can use two independent set of trajectories, one to estimate the regression coefficients and another to estimate the option price. If we make use of this approach we are actually estimating the optimal shouting policy and since any approximation is suboptimal, this approach yields a value for the Strike Reset option which is low biased.

If we use this approach its possible to work forwards in time and identify the optimal shouting policy. This approach can be implemented as follows

[2]Note that the shouting value can be approximated by a set of univariate basis functions as well, since it does not depend on the current value of the strike, but just on the current value of the underlying asset. However we implemented both methods and we could not observe a notable difference in the obtained results, hence due to the extra computational cost that arises we use the bivariate functions to approximate the shouting value of this option.

1.3.2 Strike Reset Put Option Forward Valuation Algorithm

1. Use the Strike Reset Put Option Valuation Algorithm described previously, to estimate the regression coefficients β_i^l for all $i \in \{2, ..., M\}$ and $l \in \{0, ..., L\}$.
2. Simulate another N independent trajectories, consisting of M time steps.
3. For every path $j = 1, ..., N$ perform the following steps
   (a) For each time step $i = 1, ..., M$ check if there exists any available exercise right left and if it does proceed to (b), otherwise proceed with the next trajectory (return to 3.)
   (b) Calculate the continuation value $V_i^{l,C}\left(S_i^j, K_i^j\right)$ and the shouting value $V_i^{l,SH}\left(S_i^j, K_i^j\right)$ using the regression coefficients and equation 1.4
   (c) If $V_i^{l,C}\left(S_i^j, K_i^j\right) \leq V_i^{l,SH}\left(S_i^j, K_i^j\right)$ reset the strike of the option to the current underlying value, S_i^j.
   (d) Return to (a)
4. Calculate the value of the Strike Reset Put option using

$$V_0^L(s) = \frac{e^{-rT}}{N} \sum_{j=1}^{N} \left(K_T^j - S_T^j\right)^+ \tag{1.9}$$

   where K_T^j is the strike value of the j^{th} trajectory at maturity, which yields from the last strike reset, if a reset strike is used, otherwise it will be the value of the initial strike.

The above analysis for both, the Strike Reset Put Option Valuation Algorithm and the Strike Reset Put Option Forward Valuation Algorithm are presented for the specific case of the Strike Reset Put option. Both can easily be extended to value any similar Strike Reset option.

1.4 Convergence and Errors

For the Strike Reset options the approximation errors are similar to the Bermudan option errors. Firstly we have the error which yields by approximating the Strike Reset option using a discrete version of this option, which allows exercise only on a finite set of times. Although if we include a reasonably large number of exercise opportunities then the value of our approximation should converge to the real value of the Strike Reset option. This suggest that our estimate for the value of Strike Reset Put option should be low-biased.

Moreover an additional source of error comes from the fact that in order to determine the price of the L Strike Reset option, using the Dynamic Programming Formulation, we need to approximate the values of all the Strike Reset contracts with l strike reset rights, where $0 \leq l \leq L-1$. As a result we have the error of our approximations for the values of all the Strike Reset options with less strike reset rights to propagate to the approximation for the value of the L Strike Reset option. This suggests than in order to get a accurate approximation for the L Strike Reset option, we need to obtain a much more accurate approximation for all the options with less strike reset rights as the errors made in those approximations may not cancel each other out, and hence we may have a huge unexpected error for the L Strike Reset option approximation.

Furthermore, similarly with the Bermudan option case and following [3], we have an additional error which comes from two parts. The first part of the error comes from approximating the conditional expectations, which arise in the Dynamic Programming Formulation, using a finite set of basis functions. The second part of the error comes when we estimate the actual conditional expectations using Monte Carlo techniques and least squares methods, in order to compute the value function of the Dynamic Programming Principle.

As a result in order to get an accurate estimate for the price of the Strike Reset option we need to take into account all these errors, since just minimizing one of those maybe has a limited effect in the accuracy of our algorithm.

For the convergence of this method we do not provide any proof and there does not exist any proof in the literature. We believe although that the convergence of this method can be proved using similar techniques as in [3]. Moreover the numerical results that we obtain in chapter 2 provide some justification for the convergence of this method, since are very close to results obtained at [19]. This although, may subject to further research and investigation.

1.5 Estimating the Greeks

Option prices to some extend can be derived from market prices, but this is not the case with Greeks. As a result in this section we use simulation techniques in order to estimate the Greeks of the Strike Reset Put option. In order to determine the Greeks we use Likelihood Ratio Method and Pathwise Sensitivities Approach. These ideas follow [20] and [12].

1.5.1 Likelihood Ratio Method

For the Likelihood Ratio Method we assume that the underlying process $S = (S_1, ..., S_M)$ has a probability density function p and that θ is a parameter of this density. In order to emphasize this dependence we write p_θ from now on.

The value of the Strike Reset option can be given by

$$V_0^L(S_0, K_0) = \mathbb{E}\left[e^{-r\Delta t} V_1^L(S_1, K_0) \,|S_0\right] \tag{1.10}$$

since there is no exercise right at $t = 0$. Hence

$$V_0^L(S_0, K_0) = \int e^{-r\Delta t} V_1^L(S_1, K_0)\, p_\theta(S_0, S_1)\, \mathrm{d}S_1$$

where $p_\theta(S_0, S_1)$ is the transition density of S_t from time t_0 to time t_1. As a result

$$\begin{aligned}
\frac{\partial V_0^L}{\partial \theta}(S_0, K_0) &= \frac{\partial}{\partial \theta} \int e^{-r\Delta t} V_1^L(S_1, K_0)\, p_\theta(S_0, S_1)\, \mathrm{d}S_1 \\
&= \int e^{-r\Delta t} V_1^L(S_1, K_0) \frac{\partial p_\theta}{\partial \theta}(S_0, S_1)\, \mathrm{d}S_1 \\
&= \int e^{-r\Delta t} V_1^L(S_1, K_0) \frac{\partial \log p_\theta}{\partial \theta}(S_0, S_1) p_\theta(S_0, S_1)\, \mathrm{d}S_1 \\
&= \mathbb{E}\left[e^{-r\Delta t} V_1^L(S_1, K_0) \frac{\partial \log p_\theta}{\partial \theta}(S_0, S_1) \,|S_0\right]
\end{aligned} \tag{1.11}$$

Note that this method relies on the interchange of differentiation and integration. We provide sufficient conditions for the interchange of differentiation and integration, according to [11], and we prove that these conditions hold for the Delta of this option at Appendix A of this thesis.

Now given that the underlying price process follows a geometric Brownian motion,

$$dS_t = rS_t dt + \sigma S_t dB_t$$

we have that

$$S_1 = S_0 e^{\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t} Z_1} \tag{1.12}$$

The transition density from S_0 to S_1 is given by

$$p_\theta(S_0, S_1) = \frac{1}{S_1 \sigma \sqrt{2\pi\Delta t}} e^{-\frac{1}{2}\left(\frac{\log \frac{S_1}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t}{\sigma\sqrt{\Delta t}}\right)^2} \tag{1.13}$$

and hence

$$\log p_\theta(S_0, S_1) = -\log S_1 - \log \sigma - \frac{1}{2}\log(2\pi\Delta t) - \frac{1}{2}\left(\frac{\log \frac{S_1}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t}{\sigma\sqrt{\Delta t}}\right)^2$$

As a result for the Delta of the option

$$\frac{\partial \log p_\theta}{\partial S_0}(S_0, S_1) = \frac{1}{S_0 \sigma^2 \Delta t}\left(\log \frac{S_1}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t\right) \tag{1.14}$$

and if S_1 is generated from S_0 using a standard normal random variable Z_1, as given by equation 1.12, then

$$\log \frac{S_1}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t = \sigma Z_1 \sqrt{\Delta t}$$

and consequently the Delta of the Strike Reset Put option is given by

$$\Delta(S_0) = \frac{\partial V_0^L}{\partial S_0}(S_0, K_0) = \mathbb{E}\left[e^{-r\Delta t} V_1^L(S_1, K_0) \frac{Z_1}{S_0 \sigma \sqrt{\Delta t}} | S_0\right] \tag{1.15}$$

The Delta of the Strike Reset Put option is hence very easy to implement and we can calculate this at the same time as estimating the value of the option $V_0^L(S_0, K_0)$.

Using the Strike Reset Put Option Valuation Algorithm, which moves backwards in time at the last time step we will obtain our approximation for $V_1^L(S_1, K_0)$ for every trajectory, and hence our estimator for the Delta of the option, using the Likelihood Ratio Method will be

$$\Delta(S_0) = \frac{e^{-r\Delta t}}{N S_0 \sigma \sqrt{\Delta t}} \sum_{i=1}^{N} V_1^L\left(S_1^j, K_0\right) Z_1^j \tag{1.16}$$

Now in order to approximate the Delta of this option using the Forward Valuation Algorithm, we have to rewrite equation 1.15 as

$$\begin{aligned}
\Delta(S_0) &= \mathbb{E}\left[e^{-r\Delta t} V_1^L(S_1, K_0) \frac{Z_1}{S_0 \sigma \sqrt{\Delta t}} | S_0\right] \\
&= \mathbb{E}\left[e^{-r\Delta t} \mathbb{E}\left[e^{-r(T-t_1)} (K_T - S_T)^+ | S_1\right] \frac{Z_1}{S_0 \sigma \sqrt{\Delta t}} | S_0\right] \\
&= \mathbb{E}\left[e^{-rT} (K_T - S_T)^+ \frac{Z_1}{S_0 \sigma \sqrt{\Delta t}} | S_0\right]
\end{aligned} \tag{1.17}$$

where the last equality arises using the fact that Z_1 is $\mathcal{F}_1$ measurable and using the tower property of the conditional expectation. Note that K_T is the value of the strike at maturity. Hence our estimator for the Delta of the Strike Reset Put option using the Likelihood Ratio Method and the Forward Valuation Algorithm will be

$$\Delta(S_0) = \frac{e^{-rT}}{N S_0 \sigma \sqrt{\Delta t}} \sum_{i=1}^{N} \left(K_T^j - S_T^j\right)^+ Z_1^j$$

In order to get second order sensitivities note that

$$\begin{aligned}
\frac{\partial^2 \log p_\theta}{\partial \theta^2} &= \frac{\partial}{\partial \theta}\left(\frac{1}{p_\theta}\frac{\partial p_\theta}{\partial \theta}\right) = \frac{1}{p}\frac{\partial^2 p_\theta}{\partial \theta^2} - \frac{1}{p_\theta^2}\left(\frac{\partial p_\theta}{\partial \theta}\right)^2 \\
\Rightarrow \frac{1}{p}\frac{\partial^2 p_\theta}{\partial \theta^2} &= \frac{\partial^2 \log p_\theta}{\partial \theta^2} + \left(\frac{\partial \log p_\theta}{\partial \theta}\right)^2
\end{aligned}$$

where to simplify notation we have used p_θ for $p_\theta(S_0, S_1)$. As a result

$$\begin{aligned}\frac{\partial^2 V_0^L}{\partial\theta^2}(S_0,K_0) &= \frac{\partial^2}{\partial\theta^2}\mathbb{E}\left[e^{-r\Delta t}V_1^L(S_1,K_0)\right]\\ &= \int e^{-r\Delta t}V_1^L(S_1,K_0)\frac{\partial^2 p_\theta}{\partial\theta^2}\,\mathrm{d}S_1\\ &= \mathbb{E}\left[e^{-r\Delta t}V_1^L(S_1,K_0)\left(\frac{\partial^2\log p_\theta}{\partial\theta^2}+\left(\frac{\partial\log p_\theta}{\partial\theta}\right)^2\right)|S_0\right]\end{aligned} \tag{1.18}$$

where we assumed again that the interchange of differentiation and integration is allowed, see Appendix A for further details about this.

Assuming again a geometric Brownian motion for the underlying process and using the transition density given by equation 1.13 we obtain that

$$\begin{aligned}\frac{\partial^2\log p_\theta}{\partial S_0^2} &= \frac{-1-\left(\log\frac{S_1}{S_0}-\left(r-\frac{1}{2}\sigma^2\right)\Delta t\right)}{\sigma^2 S_0^2\Delta t}\\ &= \frac{-1-\sigma Z_1\sqrt{\Delta t}}{\sigma^2 S_0^2\Delta t}\end{aligned}$$

where the second equality arises since S_1 is generated from S_0 using a standard normal random variable Z_1, as in equation 1.12. As a result we obtain that the Gamma of the Strike Reset Put option will be

$$\Gamma(S_0) = \mathbb{E}\left[e^{-r\Delta t}V_1^L(S_1,K_0)\left(\frac{Z_1^2-1}{\sigma^2 S_0^2\Delta t}-\frac{Z_1}{\sigma S_0^2\sqrt{\Delta t}}\right)|S_0\right] \tag{1.19}$$

which, similarly to Delta is very easy to implement and can be calculated at the same time as we estimate the value of the option $V_0^L(S_0,K_0)$.

Using the Strike Reset Put Option Valuation Algorithm which moves backwards in time our estimator for the Gamma of the option using the Likelihood Ratio Method will be

$$\Gamma(S_0) = \frac{e^{-r\Delta t}}{N}\sum_{j=1}^{N}V_1^L\left(S_1^j,K_0\right)\left(\frac{\left(Z_1^j\right)^2-1}{\sigma^2 S_0^2\Delta t}-\frac{Z_1^j}{\sigma S_0^2\sqrt{\Delta t}}\right) \tag{1.20}$$

For the Gamma of this option, using the Forward Valuation Algorithm and the Likelihood Ratio Method we need to rewrite equation 1.19 as

$$\begin{aligned}\Gamma(S_0) &= \mathbb{E}\left[e^{-r\Delta t}V_1^L(S_1,K_0)\left(\frac{Z_1^2-1}{\sigma^2 S_0^2\Delta t}-\frac{Z_1}{\sigma S_0^2\sqrt{\Delta t}}\right)|S_0\right]\\ &= \mathbb{E}\left[e^{-rT}(K_T-S_T)^+\left(\frac{Z_1^2-1}{\sigma^2 S_0^2\Delta t}-\frac{Z_1}{\sigma S_0^2\sqrt{\Delta t}}\right)|S_0\right]\end{aligned} \tag{1.21}$$

using identical arguments as for the Delta of the option and hence our approximation for the Gamma will be

$$\Gamma\left(S_0\right)=\frac{e^{-rT}}{N}\sum_{i=1}^{N}\left(K_T^j-S_T^j\right)^+\left(\frac{\left(Z_1^j\right)^2-1}{\sigma^2S_0^2\Delta t}-\frac{Z_1^j}{\sigma S_0^2\sqrt{\Delta t}}\right) \tag{1.22}$$

Rho and Vega of the Strike Reset Put option cannot be calculated using the Likelihood Ratio Method. This is the case since $V_1^L\left(S_1,K_0\right)$ does depend on r and σ, whereas for the Gamma and Delta of the Strike Reset put option $V_1^L\left(S_1,K_0\right)$ does not depend on S_0 due to the Markov property of S_t.

Likelihood Ration Method may be a simple and fairly easy way to obtain the sensitivities of options, but it has a major disadvantage. As it can be observed by both the Delta and the Gamma of the Strike Reset Put option, for a small σ or for a small time step Δt the variance of the estimators will explode. This indicates that for the Strike Reset Put option, as the number of possible exercise dates increases, in order to get a better approximation for the value of this option, the variance of the Likelihood Ratio Method estimators for both Delta and Gamma will be really large. Generally Likelihood Ratio Method is an approach with a large variance compared to other techniques which are used to obtain option sensitivities.

1.5.2 Pathwise Sensitivities Approach

Pathwise Sensitivities Approach is an alternative method for estimating the Greeks, which has a smaller variance compared to Likelihood Ratio Method. On the other hand it has some restrictions as we will see further on.

For Pathwise Sensitivities according to [8], we assume that the dependence of parameter θ comes from the underlying price process, and not from the probability density function of the underlying price process, which was the case for Likelihood Ratio Method. As a result in order to emphasise this dependence we denote the underlying price process by $S_i(\theta)$ from now on. This means that for a fixed outcome $\omega\in\Omega$, $\theta\to S(\theta,\omega)$ is a random function of θ. The value of the Strike Reset Put option is given by

$$V_0^L\left(S_0\right)=e^{-rT}\mathbb{E}\left[\left(K_T-S_T\right)^+|S_0\right] \tag{1.23}$$

where K_T is the value of the strike at maturity. Assuming the dependence of S on θ

$$\begin{aligned}V_0^L\left(S_0\left(\theta\right)\right)&=e^{-rT}\mathbb{E}\left[\left(K_T\left(\theta\right)-S_T\left(\theta\right)\right)^+|S_0\right]\\&=e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(X_i\left(\theta\right)-S_T\left(\theta\right)\right)^+\mathbf{1}_{F_i}|S_0\right]\end{aligned} \tag{1.24}$$

where

$$X_i(\theta) = \begin{cases} K_0 & \text{if } i = 0 \\ S_i(\theta) & \text{otherwise} \end{cases}$$

and F_i is the event that last time that a reset right has been exercised is t_i, for $i > 0$, and F_0 is the event that no strike reset has been exercised.

Hence

$$F_i = \left\{ V_i^{l,SH} \geq V_i^{l,C} \text{ and } V_j^{l-1,SH} < V_j^{l-1,C}, j = i+1, ..., M \right\} \text{ for } i > 0 \text{ and } 1 \leq l \leq L-1$$
$$F_0 = \left\{ V_j^{L,SH} < V_j^{L,C}, j = 1, ..., M \right\}$$

where we set $V_i^{0,SH} = 0 \ \ \forall i \in \{1, ..., M\}$.

As a result

$$\begin{aligned} \frac{\partial}{\partial\theta} V_0^L(S_0(\theta)) &= \frac{\partial}{\partial\theta} e^{-rT} \mathbb{E}\left[\sum_{i=0}^{M} (X_i(\theta) - S_T(\theta))^+ \mathbf{1}_{F_i} | S_0 \right] \\ &= e^{-rT} \mathbb{E}\left[\sum_{i=0}^{M} \frac{\partial}{\partial\theta} \left[(X_i(\theta) - S_T(\theta))^+ \mathbf{1}_{F_i} \right] | S_0 \right] \\ &= e^{-rT} \mathbb{E}\left[\sum_{i=0}^{M} \frac{\partial}{\partial\theta} (X_i(\theta) - S_T(\theta))^+ \mathbf{1}_{F_i} + (X_i(\theta) - S_T(\theta))^+ \frac{\partial}{\partial\theta} \mathbf{1}_{F_i} | S_0 \right] \end{aligned} \tag{1.25}$$

where we assumed that the interchange of differentiation and integration is valid. We do not provide any proof for this assumption. However the numerical results that are obtained in chapter 2, suggest that the interchange of integration and differentiation holds, but this requires further research and investigation.
Let

$$R_i^l = \left\{ V_i^{l,SH}(S_i(\theta)) < V_i^{l,C}(S_i(\theta)) \right\} \tag{1.26}$$

Then we can write F_i as

$$\begin{aligned} F_i &= \bigcup_{l=1}^{L} \left\{ V_i^{l,SH} \geq V_i^{l,C} \right\} \cap \left\{ V_{i+1}^{l-1,SH} < V_{i+1}^{l-1,C} \right\} \cap ... \cap \left\{ V_M^{l-1,SH} < V_M^{l-1,C} \right\} \\ &= \bigcup_{l=1}^{L} \bar{R}_i^l \cap R_{i+1}^{l-1} \cap ... \cap R_M^{l-1} \end{aligned} \tag{1.27}$$

As a result we have that

$$\mathbf{1}_{F_i} = \sum_{l=1}^{L} \mathbf{1}_{\bar{R}_i^l} \mathbf{1}_{R_{i+1}^{l-1}} \mathbf{1}_{R_{i+2}^{l-1}} ... \mathbf{1}_{R_M^{l-1}} \tag{1.28}$$

Hence the product rule of differentiation allows as to write the derivative of $\mathbf{1}_{F_i}$ as sum of derivatives of $\mathbf{1}_{\bar{R}_i^l}$ or $\mathbf{1}_{R_j^{l-1}}$ multiplied with some other indicator functions. If the process $S_i(\theta)$ is almost surely continuous with respect to parameter θ for each i, then on each path

$$\mathbf{1}_{\left\{V_i^{l,SH}(S_i(\theta))<V_i^{l,C}(S_i(\theta))\right\}} = \mathbf{1}_{\left\{V_i^{l,SH}(S_i(x))<V_i^{l,C}(S_i(x))\right\}} \tag{1.29}$$

given that $V_i^{l,C}$ is continuous and $|x-\theta| < \epsilon_\omega$, where ϵ depends on the given path ω. $V_i^{l,C}(S(\theta))$ is clearly continuous when $\theta = S_0$ due to the Markov property of S_t. For the rest of the sensitivities this continuity is not so obvious and we do not provide a proof for this. However this may subject to further research and investigation.

As a result $\frac{\partial}{\partial\theta}\mathbf{1}_{R_j^l} = 0$ which yields that $\frac{\partial}{\partial\theta}\mathbf{1}_{F_i} = 0$ and consequently equation 1.25 becomes

$$\frac{\partial}{\partial\theta}V_0^L(S_0(\theta)) = e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\frac{\partial}{\partial\theta}\left(X_i(\theta) - S_T(\theta)\right)^+\mathbf{1}_{F_i}|S_0\right] \tag{1.30}$$

Subsequently, for the Delta of the Strike Reset Put option we obtain that

$$\begin{aligned}
\Delta(S_0) &= e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(\frac{\partial}{\partial X_i}(X_i - S_T)\frac{\partial X_i}{\partial S_0}\mathbf{1}_{X_i>S_T} + \frac{\partial}{\partial S_T}(X_i - S_T)\frac{\partial S_T}{\partial S_0}\mathbf{1}_{X_i>S_T}\right)\mathbf{1}_{F_i}|S_0\right] \\
&= e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(\frac{\partial X_i}{\partial S_0}\mathbf{1}_{X_i>S_T} - \frac{\partial S_T}{\partial S_0}\mathbf{1}_{X_i>S_T}\right)\mathbf{1}_{F_i}|S_0\right] \\
&= e^{-rT}\mathbb{E}\left[\left(\sum_{i=1}^{M}\left(\frac{\partial S_i}{\partial S_0}\mathbf{1}_{S_i>S_T} - \frac{\partial S_T}{\partial S_0}\mathbf{1}_{S_i>S_T}\right)\mathbf{1}_{F_i}\right) - \frac{\partial S_T}{\partial S_0}\mathbf{1}_{K_0>S_T}\mathbf{1}_{F_0}|S_0\right]
\end{aligned} \tag{1.31}$$

Assuming that the underlying process is a geometric Brownian motion then

$$S_i = S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)t_i+\sigma B_i} \tag{1.32}$$

where $B_i \sim \mathcal{N}(0, t_i)$. As a result we have that

$$\frac{\partial S_t}{\partial S_0} = e^{\left(r-\frac{1}{2}\sigma^2\right)t+\sigma B_t} = \frac{S_t}{S_0} \tag{1.33}$$

which implies that the Delta of the Strike Reset Option is given by

$$\Delta(S_0) = e^{-rT}\mathbb{E}\left[\left(\sum_{i=1}^{M}\frac{S_i - S_T}{S_0}\mathbf{1}_{S_i>S_T}\mathbf{1}_{F_i}\right) - \frac{S_T}{S_0}\mathbf{1}_{K_0>S_T}\mathbf{1}_{F_0}|S_0\right] \tag{1.34}$$

Our approximation for the Delta of this option using the Forward Valuation Algorithm is given by

$$\Delta(S_0) = \frac{e^{-rT}}{N}\sum_{j=1}^{N}\left[\left(\sum_{i=1}^{M}\frac{S_i^j - S_T^j}{S_0}\mathbf{1}_{S_i^j>S_T^j}\mathbf{1}_{F_i^j}\right) - \frac{S_T^j}{S_0}\mathbf{1}_{K_0^j>S_T^j}\mathbf{1}_{F_0^j}\right]$$

where F_i^j is the event previously described but is now used for the j^{th} trajectory.

For the Vega of the Strike Reset Put option we obtain that

$$\begin{aligned}
Vega\left(S_0\right) &= e^{-rT}\frac{\partial V_0^L}{\partial \sigma}\mathbb{E}\left[\sum_{i=0}^{M}\left(X_i - S_T\right)^+ \mathbf{1}_{F_i}|S_0\right] \\
&= e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(\frac{\partial}{\partial X_i}\left(X_i - S_T\right)\frac{\partial X_i}{\partial \sigma}\mathbf{1}_{X_i>S_T} + \frac{\partial}{\partial S_T}\left(X_i - S_T\right)\frac{\partial S_T}{\partial \sigma}\mathbf{1}_{X_i>S_T}\right)\mathbf{1}_{F_i}|S_0\right] \\
&= e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(\frac{\partial X_i}{\partial \sigma}\mathbf{1}_{X_i>S_T} - \frac{\partial S_T}{\partial \sigma}\mathbf{1}_{X_i>S_T}\right)\mathbf{1}_{F_i}|S_0\right] \\
&= e^{-rT}\mathbb{E}\left[\left(\sum_{i=1}^{M}\left[\left(B_{t_i} - \sigma t_i\right)S_i - \left(B_T - \sigma T\right)S_T\right]\mathbf{1}_{S_i>S_T}\mathbf{1}_{F_i}\right) - \left(B_T - \sigma T\right)S_T\mathbf{1}_{K_0>S_T}\mathbf{1}_{F_0}|S_0\right]
\end{aligned} \tag{1.35}$$

For the Rho of the Strike Reset Put option we need to do a minor modification of this method, due to the discounting term that appears in equation 1.24.

$$\begin{aligned}
\rho\left(S_0\right) &= \frac{\partial}{\partial r}\left(e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(X_i\left(\theta\right) - S_T\left(\theta\right)\right)^+ \mathbf{1}_{F_i}|S_0\right]\right) \\
&= -Te^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(X_i\left(\theta\right) - S_T\left(\theta\right)\right)^+ \mathbf{1}_{F_i}|S_0\right] + e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\frac{\partial}{\partial r}\left(X_i\left(\theta\right) - S_T\left(\theta\right)\right)^+ \mathbf{1}_{F_i}|S_0\right] \\
&= e^{-rT}\mathbb{E}\left[\sum_{i=0}^{M}\left(-T\left(X_i - S_T\right) + \frac{\partial}{\partial X_i}\left(X_i - S_T\right)\frac{\partial X_i}{\partial r} + \frac{\partial}{\partial S_T}\left(X_i - S_T\right)\frac{\partial S_T}{\partial r}\right)\mathbf{1}_{X_i>S_T}\mathbf{1}_{F_i}|S_0\right] \\
&= e^{-rT}\mathbb{E}\left[\left(\sum_{i=1}^{M}S_i\left(t_i - T\right)\mathbf{1}_{S_i>S_T}\mathbf{1}_{F_i}\right) - TK_0\mathbf{1}_{K_0>S_T}\mathbf{1}_{F_0}|S_0\right]
\end{aligned} \tag{1.36}$$

Our estimators for the Vega and the Rho of the Strike Reset Put option using Pathwise Sensitivities Approach and the Forward Valuation Algorithm, are very easily obtained from equations 1.35 and 1.36 following a similar approach as in equation 1.5.2.

Pathwise Sensitivities Approach has the limitation that it requires the payoff to be differentiable. As a result when we consider the Gamma of this option, the payoff is not twice differentiable and hence this method cannot be used to obtain the Gamma of the Strike Reset Put option. However this method can be modified to handle cases where payoff does not have the necessary continuity. One can use smoothness techniques like piecewise linear approximation of the discontinuity to avoid this limitation.

Finally comparing the two approaches for estimating sensitivities, we can see that the Likelihood Ratio Method can handle discontinuous payoffs, although it has a large variance and

its not applicable to all the Greeks. On the other hand Pathwise Sensitivities Approach has a lower variance compared the Likelihhod Ratio Method but it requires continuous payoffs for first derivatives and similarly continuous first derivatives in order to obtain second order sensitivities.

Chapter 2

Implementation and Numerical Results

In this chapter we present the numerical results we obtained for the Strike Reset Put option using the techniques described in chapter 1. We did the coding using Matlab, which is provided in the Appendix B section of this thesis.

For the underlying price process we assumed a geometric Brownian motion

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t \mathrm{d}B_t$$

where we assumed that interest rate r and volatility σ are both constant. S_t denotes the underlying price process and B_t is a Brownian motion. As a result in order to construct sample trajectories we were able to use the explicit formulae

$$S_{t_i} = S_{t_{i-1}} e^{\left(r-\frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z_i}$$

where $\Delta t = t_i - t_{i-1}$ is constant and $Z_i \sim \mathcal{N}(0,1)$.

For the Error of the Monte Carlo method we use the Monte Carlo Error, denoted by MCE, which is given by

$$MCE = \sqrt{\frac{\hat{\sigma}^2}{N}} \tag{2.1}$$

where $\hat{\sigma}^2$ is the variance of the sample option values, and is given by

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{j=1}^{N}\left(\hat{V}_0^L - \frac{1}{N}\sum_{j=0}^{N}\hat{V}_0^L(S_0^j, K_0)\right)^2 \tag{2.2}$$

2.1 Strike Reset Put Option Valuation Algorithm and Results

For the Strike Reset Put option in order to compare the results that we obtain, with the ones provided by [19], we consider an option specified by

- $S_0 = 10$

- $K = 10$
- $\sigma = 0.25$
- $T = 5$
- $r = 0.06$
- $L = 5$

where T is the number of years until the expiration of the option and L is the number of exercise rights that the holder of the Strike Reset Put option has.

Table 2.1: Multiple Strike Reset Put Option Values For a Fixed M and 4 Independent Runs

N	Approximation 1	Approximation 2	Approximation 3	Approximation 4
1500	2.191	3.265	3.463	2.216
2000	3.251	2.226	3.103	2.975
2500	2.105	2.261	2.270	2.44
3000	2.156	2.177	2.195	2.144
3500	2.124	2.225	2.118	2.136
4000	2.469	2.181	2.243	2.123
4500	2.133	2.126	2.200	2.211
5000	2.203	2.155	2.202	2.194
5500	2.116	2.164	2.188	2.208
6000	2.151	2.166	2.207	2.175
6500	2.197	2.142	2.187	2.165
7000	2.147	2.188	2.178	2.193

Tables 2.1 and 2.2 present values of the Strike Reset Put option described above, for different number of trajectories and exercise dates. In order to obtain these results we use simple powers of the price variable s, up to order six as a set of univariate basis functions:

$$\varphi_1(s) = 1, \ \varphi_2(s) = s, \ \ldots, \ \varphi_7(s) = s^6$$

As a set of bivariate basis functions we use again simple powers of the price variable s and the strike variable k, but this time we include cross variational terms as well. We use all the combinations of the two variables up to order five:

$$\phi_i(s,k) = s^n k^m, \text{ where } 0 \leq n+m \leq 5 \text{ and } 0 \leq n, m$$

For the results presented at Table 2.1, we assume that the holder of the option has the right to reset the strike of this contract at $M = 30$ dates during the life of the option. This table presents the values for the Strike Reset Put option we obtained using the Strike Reset Put option Valuation Algorithm four distinct times. From this table we can observe that for a low number of simulated trajectories the regression coefficients are not stable, and hence the estimated price is too volatile. On the other hand as the number of simulated trajectories increases, it can be observed that the obtained prices converge to a value close to 2.2.

Using [19] as a reference for the Strike Reset Put option value, one can see that the value of this option should be 2.4374. As a result, this suggest that the value obtained using the Strike Reset Put option Valuation Algorithm, for a large enough value of N, underestimates the true value of the option. This recommends that increasing the number of exercise dates, will probably increase the value of the approximation and yield a more accurate value for the Strike Reset Put option. Consequently we repeat the same calculations, but now for a fixed number of trajectories N, with an increasing number of exercise opportunities M.

Table 2.2: Multiple Strike Reset Put Option Values For a Fixed N

M	30	35	45	55	65	75	85	90
Value	2.142	2.155	2.214	2.247	2.340	2.424	2.428	2.439

Table 2.2 shows that the estimated option price increases with an increasing number of exercises dates, and converges to the reference value of the Strike Reset option that is provided in [19]. Its worth noting that the difference of the two estimates is less than 1%, which implies that the Strike Reset Option Valuation Algorithm described at the previous chapter gives accurate results. We should mention here that Windcliff, Forsyth and Vetzal in [19], are using a Finite Difference Approach to estimate the value of the this contract. As discussed earlier, the main disadvantage of Monte Carlo methods, compared to Finite Difference methods, is that they have larger errors due to the variance. The results that we obtain verify this fact since the computational cost for obtaining the value 2.439 as an approximation for the value of this contract, seem to be considerably higher than than the computational cost of the Finite Difference method. It should be mentioned although that since Monte Carlo methods scale better with the dimension of the problem, this method may be preferred as the dimension of the problem increases. This although requires further investigation and research.

2.2 Strike Reset Put Option Forward Valuation Algorithm and Results

As described in the previous chapter the Backward Valuation Algorithm gives an estimate for both, the price of the option and the regression coefficients. However as mentioned earlier this algorithm may yield option values which are high-biased compared to the actual price of a contract. Note that this phenomenon was observed for different sets of basis functions, for which a smaller number of basis functions was used. As a result we will now use the Backward Valuation Algorithm to obtain the regression coefficients and then using the Forward Valuation algorithm we estimate the price of this contract. For this section we will use the same option as before but now with $S_0 = 8$.

Table 2.3: Forward Valuation Algorithm

M	5	10	15	20	25	30	40
Value	1.868	2.105	2.164	2.203	2.2119	2.262	2.257

Table 2.3 shows the values of the Strike Reset Put option using the Forward Valuation Algorithm for different number of exercise dates. Initially we calculated the regression coefficients using the $N = 10000$ simulated paths and then we used the Forward Valuation Algorithm with $N = 15000$ to obtain the value for this option. In order to obtain these results, similarly with before, we have used univariate basis functions up to order six and bivariate basis functions up to order five. A reference value for this contract as given in [19] is 2.359. As it can be observed from this table the Forward Valuation algorithm clearly underestimates the value of this contract approximately by 7%. This approximation does not improve even if we include more exercise dates or if we use another set of basis functions. These two modifications make the obtained Strike Reset Put option value to fluctuate around 2.26. As a result to avoid extra computational cost none of these two modification is used further on and for the Forward Valuation Algorithm and we restrict the possible exercise dates to $M = 30$. This is the case since using this method we are actually approximating the shouting policy, and since any approximation of the shouting policy is suboptimal, we are guaranteed a value for the option which is low-biased.

An interesting sequel of this low biased estimator would be to obtain a high biased estimate, similarly with the one obtained in [8] for Bermudan options, using martingales. By doing this we can obtain an interval in which the value of the Strike Reset Put option should lie in and check how accurate is the Backward Valuation Algorithm. This although requires further research and investigation.

Table 2.4: Convergence of the Forward Valuation Algorithm for M = 30

$N/10^3$	1	2	3	4	5	10	15	25	50	75
Value	2.351	2.193	2.281	2.241	2.254	2.282	2.261	2.262	2.264	2.261
MCE	0.585	0.387	0.314	0.285	0.254	0.176	0.140	0.108	0.076	0.062

Table 2.4 considers the impact of increasing the number of simulated paths N, while holding the number of exercise dates fixed at $M = 30$. We observe that initially MCE was too large but, as the number of simulated paths increases, MCE decreases and the value of the Strike Reset Put option converges to 2.26.

Figure 2.1: Convergence of the Forward Valuation Algorithm

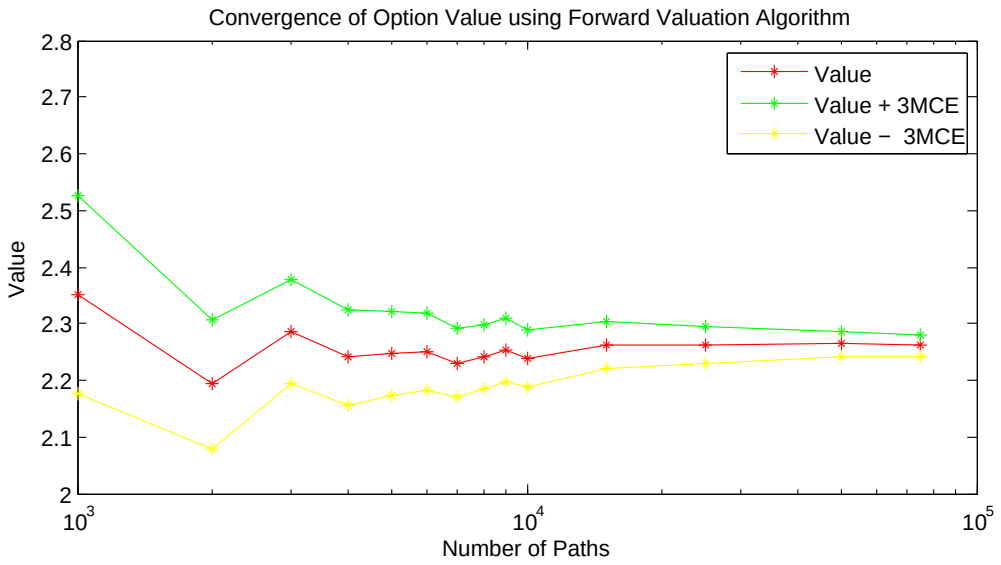


Figure 2.1 plots the value of this option as the number of paths increases. In addition it plots the value of the option $\pm 3MCE$. As it can be observed from the graph, for a low number of simulated paths the MCE is high, whereas as the number of simulated paths increases the error decreases. This figure appears to support the fact that the Forward Valuation Algorithm leads to a sub optimal estimate for the value of the option, since as the number of paths increases the value of the option plus $3MCE$ is not greater than the 2.359, which is the reference value for this contract.

Figure 2.2: Value of the Strike Reset Put option with different number of strike resets



Figure 2.2 plots the value of the Strike Reset Put option, which is described above with $S_0 = 8$, against the strike reset rights that the holder of the option initially has. As it can be observed from the graph as the number of strike resets increase the value of the option increases as well. Note that when no strike reset is available the value of the option is simply the value of a put option.

2.2.1 Greeks

As mentioned to the previous chapter we estimate the sensitivities of the Strike Reset Put option using two approaches, Likelihood Ratio Method (LRM) and Pathwise Sensitivities Approach (PSA). We will now present our estimates for the Greeks for the same Strike Reset Put option that was used previously in this section, with $S_0 = 8$.

Table 2.5: Convergence for Delta using Likelihood Ratio Method and Pathwise Sensitivities Approach for a fixed $M = 30$

$N/10^3$	1	5	10	12.5	15	25	50	75
Delta(LRM)	-0.122	-0.171	-0.113	-0.141	-0.150	-0.144	-0.120	-0.121
MCE	0.1246	0.0564	0.0411	0.0366	0.0327	0.0254	0.0173	0.0142
Delta(PSA)	-0.114	-0.121	-0.125	-0.122	-0.124	-0.124	-0.125	-0.124
MCE	0.0063	0.0034	0.0052	0.0022	0.0017	0.0013	0.0010	0.0009

Figure 2.3: Convergence for Delta using Likelihood Ratio Method for a fixed $M = 30$

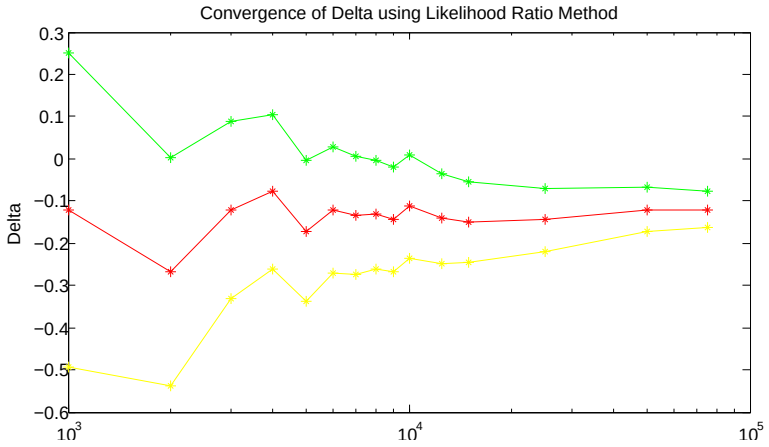


Table 2.5 presents the convergence results for the Delta of the Strike Reset Put option using both Likelihood Ratio Method and Pathwise Sensitivities Approach. Figures 2.3 and 2.4 present this convergence for each method. For both of these methods we can see that the Monte Carlo Errors decrease with the increasing number of simulated trajectories. Moreover, as it can be observed, the Delta of the Strike Reset Put option converges to the same value for both of these methods, which is -0.12. Its notable although that the Monte Carlo Errors of the Likelihood Ratio Method appears to be 14 times greater, compared to the Monte Carlo Errors of the Pathwise Sensitivities Approach, as expected. Note that for the Greeks of the Strike Reset Put option we don't have any reference numbers to compare our results with.

Figure 2.4: Convergence for Delta using Pathwise Sensitivities Approach for a fixed $M = 30$

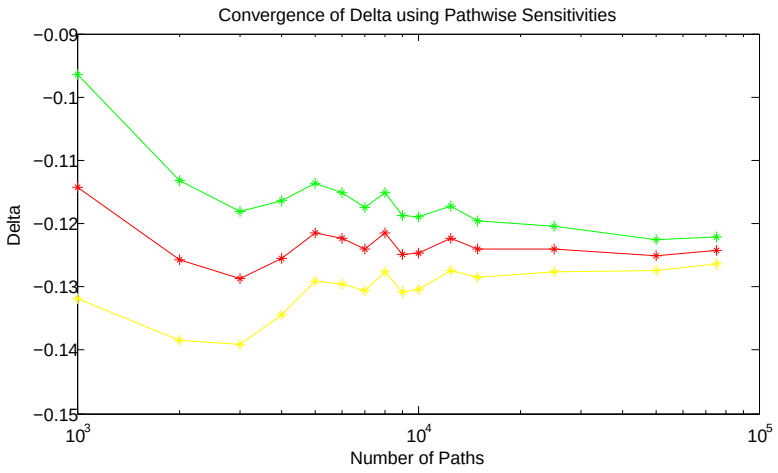


Figure 2.5: Delta as a Function of S_0

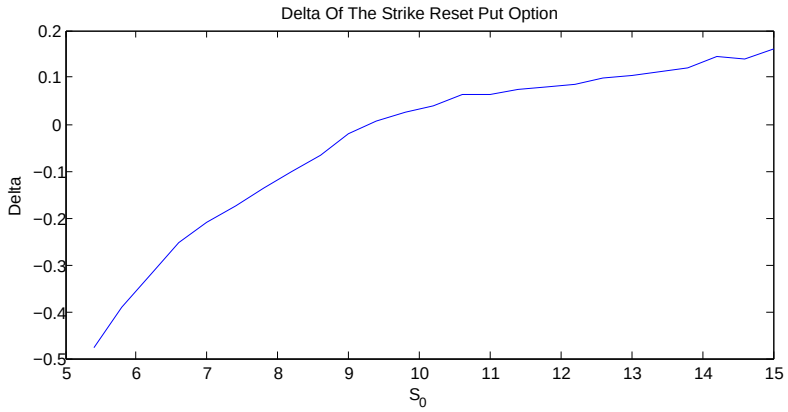


Figure 2.5 presents how the Delta of the Reset Strike Put option evolves as a function of the initial underlying price S_0. It can be observed from this figure that the value for the Delta of this option is approximately -0.12 as presented table 2.5. In the next section more figures which describe the behaviour of the Strike Reset Put option will be presented.

Table 2.6: Convergence for Gamma of the Strike Reset Put option using Likelihood Ratio Method for fixed $M = 30$

$N/10^3$	1	5	10	12.5	15	25	50	75
Gamma(LRM)	0.3047	0.2635	0.2297	0.2062	0.1454	0.1357	0.1176	0.1075
MCE	0.2570	0.1106	0.0854	0.0723	0.0632	0.0497	0.0352	0.0281

Figure 2.6: Convergence for Gamma using Likelihood Ratio Method for a fixed $M = 30$

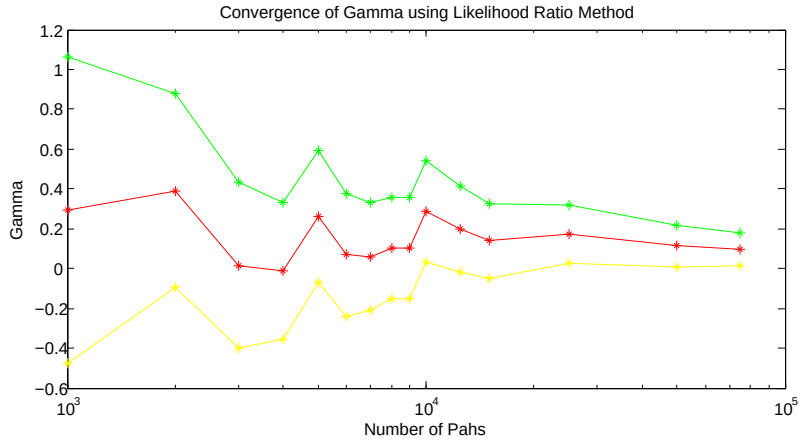


Table 2.6 presents the convergence results for the Gamma of the Strike Reset Put option, for a fixed $M = 30$, using Likelihood Ratio Method. Figure 2.6 illustrates this convergence and similarly with the Delta of this option we can see that as the number of simulated paths increase the value for the Gamma of this contract converges to 0.1075 and the Monte Carlo error decreases.

Table 2.7: Convergence for Vega of the Strike Reset Put option using Pathwise Sensitivities Approach for fixed $M = 30$

$N/10^3$	1	5	10	12.5	15	25	50	75
Vega(PSA)	4.7214	4.4761	4.4617	4.4966	4.4773	4.4929	4.4557	4.4341
MCE	0.1815	0.0807	0.0571	0.0511	0.0467	0.0361	0.0255	0.0208

Figure 2.7: Convergence for Vega using Pathwise Sensitivities Approach for a fixed $M = 30$

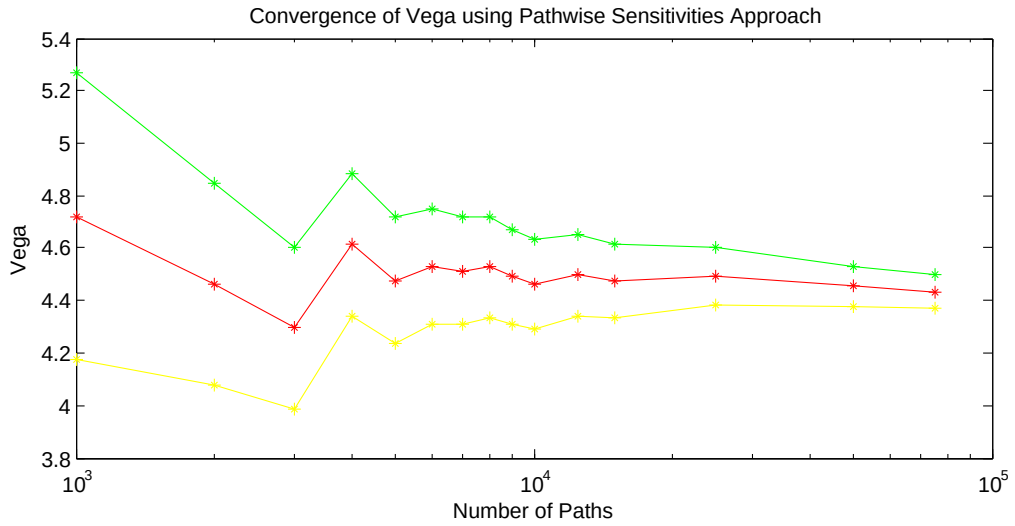


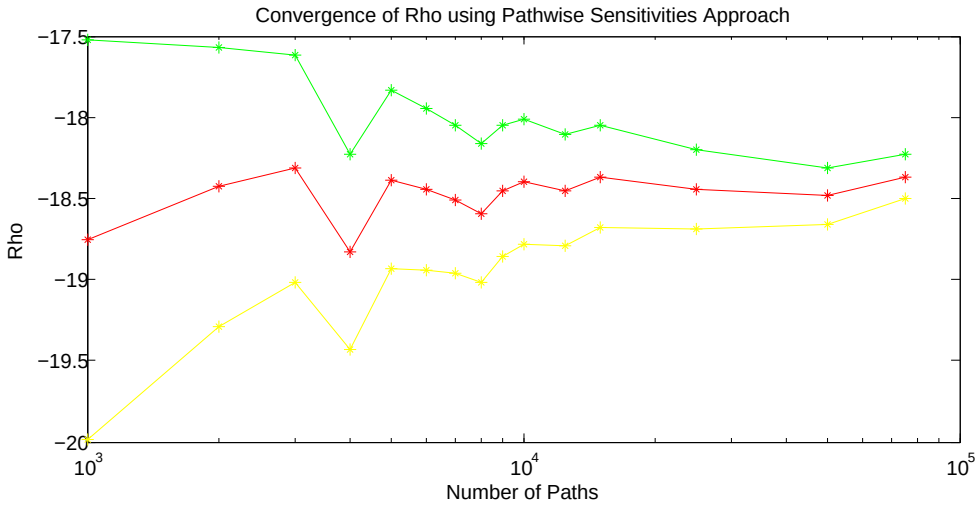
Table 2.7 presents the convergence results for the Vega of the Strike Reset Put option, for a fixed $M = 30$, using Pathwise Sensitivities Approach. Figure 2.7 illustrates this convergence. It can be observed that as the number of simulated paths increase the value for the Vega of this contract converges to 4.43 and the Monte Carlo Error decreases.

Finally Table 2.8 presents the convergence results for the Rho of the Strike Reset Put option, for a fixed $M = 30$, using Pathwise Sensitivities Approach. Figure 2.8 illustrates this convergence. It can be observed that as the number of simulated paths increase the value for the Rho of this contract converges to -18.36 and the Monte Carlo Error decreases.

Table 2.8: Convergence for Rho of the Strike Reset Put option using Pathwise Sensitivities Approach for fixed $M = 30$

$N/10^3$	1	5	10	12.5	15	25	50	75
Rho(PSA)	-18.755	-18.384	-18.399	-18.450	-18.368	-18.447	-18.486	-18.365
MCE	0.4111	0.1823	0.1277	0.1148	0.1046	0.0811	0.0571	0.0468

Figure 2.8: Convergence for Rho using Pathwise Sensitivities Approach for a fixed $M = 30$



2.3 Strike Reset Put Option General Properties

In this Section we consider a few Strike Reset Put options examples and present how their values change with different parameters. For this Section the parameters of the Strike Reset Put option, unless otherwise stated, will be

- $S_0 = 100$
- $K = 100$
- $\sigma = 0.2$
- $T = 1$
- $r = 0.1$
- $L = 5$
- $M = 30$

Figure 2.9 compares various Strike Reset Put options. It presents how the value of this option evolves for different initial value S_0 and for different number of strike reset rights. Clearly the holder of a Strike Reset Put option can choose not to exercise any strike reset right and arrive at maturity with the initial strike K_0. As a result the value of any Strike Reset Put option must be worth at least as much as the European Put option which is specified by the same parameters. Indeed this is shown in Figure 2.9. Similarly any holder of a Strike Reset Put option with more strike reset rights than a second Strike Reset Put option, but with all the other parameters identical, can replicate the payoff of that contract and hence the value of the Strike Reset Put option with the most shout opportunities has to be at least as valuable as the value of the second contract. Indeed this is shown in Figure 2.9, since as the number of strike reset rights increase the value of the options increases as well. Moreover, for low initial prices S_0, the values of these contracts are much more closer to each other, since the probability that the underlying price process exceeds the initial strike $K_0 = 100$ and a Strike Reset right is used is very low. As a result for low initial price S_0 the values of these Strike Reset Put options are almost identical to the value of the equivalent European Put option, as a Strike Reset right is unlikely to be used. Finally for large values of the initial price S_0, Strike Resets Put options with more exercise rights are much more expensive than others with fewer exercise rights because its highly likely that an exercise right will be used immediately. This is the case since the Put option with initial strike K_0 is unlikely to be in the money at maturity.

The straight line that is shown in this figure represents the value of the Strike Reset Put option with five shout opportunities and initial strike $K_0 = 0$, or equivalently the value of an at the money Strike Reset Put option with four shout opportunities. Note that the value of this contract changes linearly with respect to the initial price of the underlying S_0, since the value of an at the money European Put option is linear with respect to the initial price S_0. As a result proceeding inductively we can observe that the value of any at the money Strike Reset Put option will change linearly with S_0. From this plot we can observe that there is a point at which the graph of the Strike Reset Put option with five shout opportunities becomes tangential with this straight line. According to [6], this is the optimal point to reset the strike of this contract and switch to an at the money Strike Reset Put option with four shout opportunities.

Figure 2.9: Value of the Strike Reset Put option with different number of shout opportunities

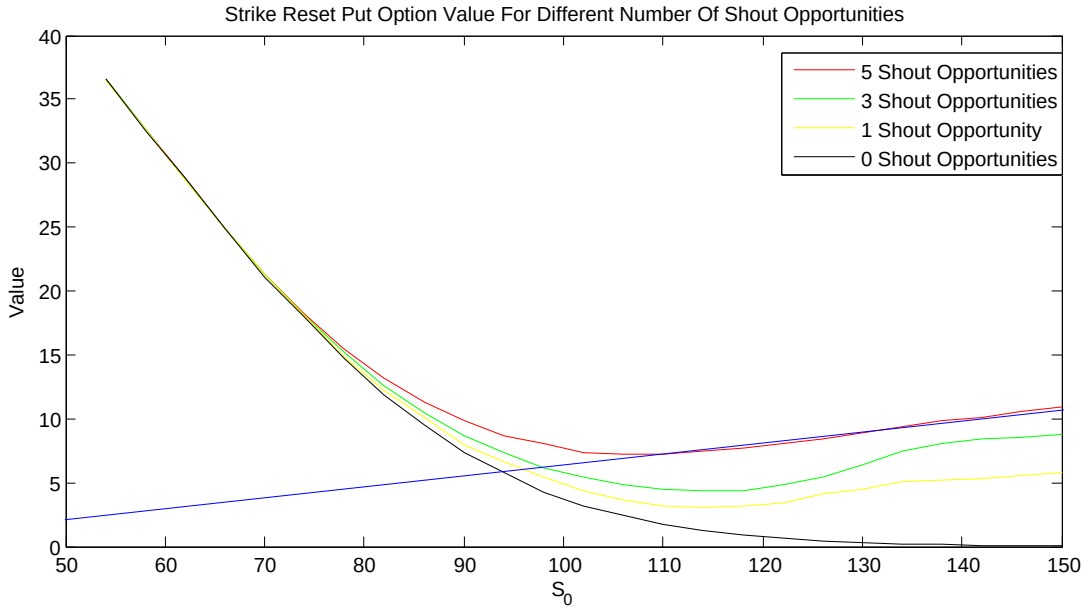


Figure 2.10 plots the value of various Strike Reset Put options against the initial value of the underlying asset, S_0, for different maturities T. As it can be observed from this plot the value of the Strike Reset Put option does not admit a monotonicity property in T. As discussed in [6], this behaviour is in contrast to the Bermudan options, since the Bermudan option values are always monotonically increasing with maturity T. This lack of monotonicity with maturity can be explained by the fact than when there is one shout opportunity available and the holder uses this right, then he receives a European Put option, which does not exhibit monotonicity in T. Similarly since the Simple Strike Reset Put option does not exhibit any monotonicity in T, this implies than the general Strike Reset Put option will not exhibit monotonicity in T as well.

Figure 2.10: Value of the Strike Reset Put option with different maturities

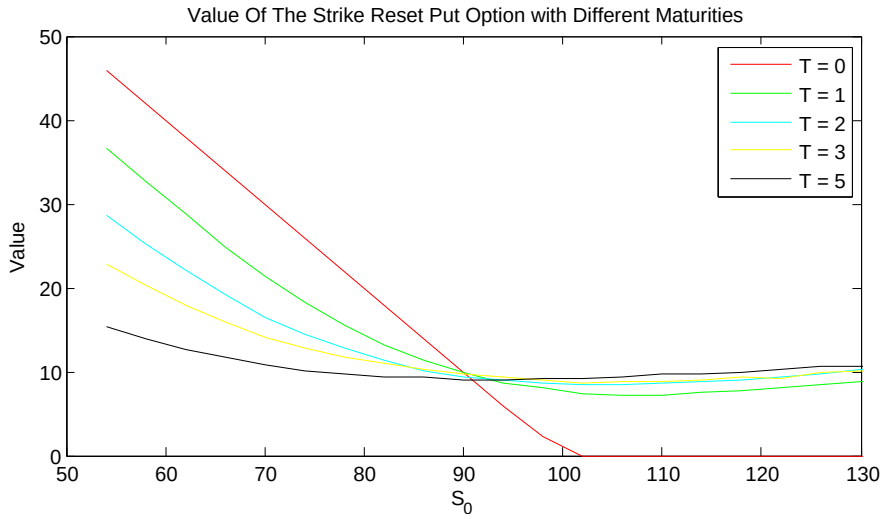
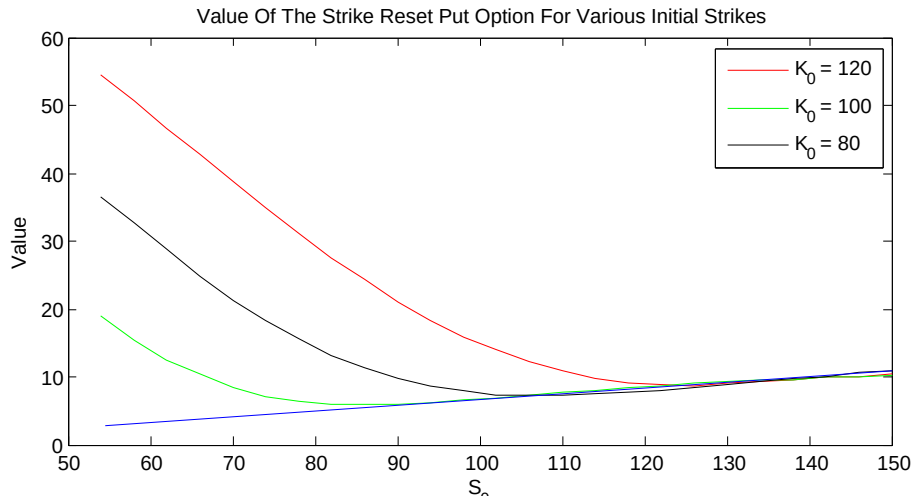


Figure 2.11 plots the value of the Strike Reset Put option for various initial strikes K_0. Clearly the higher the initial strike K_0, the higher the value of this option will be, but as discussed in [19] as $S_0 \to \infty$ the values of these contracts will be almost the same. This is the case since as $S_0 \to \infty$ the holders of these contracts are more likely to shout immediately and hence they will receive an identical contract with one strike reset opportunity less. As a result as $S_0 \to \infty$ the value of the Strike Reset Put option does not depend on the initial strike K_0. Its clearly shown from this figure that if the initial asset price is approximately 140 its not more valuable to have an initial strike $K_0 = 120$ than $K_0 = 100$ or $K_0 = 80$.

Figure 2.11: Value of the Strike Reset Put option with different initial strike values



Chapter 3

Conclusion and Final Remarks

In this thesis, we extended the least squares method and developed an approach for pricing the Strike Reset Put option using Monte Carlo methods. We tested the convergence of the algorithm for pricing the Strike Reset Put option, by comparing our results with the results that are presented in [6] using Finite Difference methods. Moreover we presented an easy way to obtain the Greeks with this method using both Likelihood Ratio Method and Pathwise Sensitivities Approach. The results presented in chapter 2 verify the convergence of these estimates.

Pricing Strike Reset options using Monte Carlo methods is an alternative to Finite Difference methods that are mostly used for the pricing of these contracts. Compared to Finite Difference Methods this approach has a slower rate of convergence, and hence Finite Deference methods may be preferred, however we believe that as the dimension of the problem increases this method may be preferred. Although this requires further research.

As mentioned in chapter 2 the Strike Reset option Forward Valuation Algorithm underestimates the price of this option. As a result, a possible extension of this work would be to obtain an upper bound for the Strike Reset option using a martingale approach and compare how close are the results that are obtained using these two methods. Another possible extension of this work would be to use a different approach for selecting the possible strike values, which are used to obtain the regression coefficients. Instead of a fixed set of strike values $K_0, ..., K_{max}$ that was used in our approximation for all timesteps, which formed a grid of values $\Sigma_i \times \Upsilon$, one can instead sample possible strike values form a uniform distribution $U(K_0, K_{max})$ for each timestep and each realised price S_i^j. We expect that this modification, using less but sampled possible strike values, will decrease the computational cost of this method and probably maintain or even increase the accuracy of the algorithm. This although requires further research and investigation.

Chapter 4

Appendix A

In order to obtain the Greeks for the Strike Reset Put option using Likelihood Ratio Method we interchange the integration and the differentiation that arises at equation 1.11 on chapter 1. On this Appendix we provide sufficient conditions for this interchange for this approach and prove that these conditions are satisfied for the Delta of the Strike Reset option. In a similar way one can prove that these conditions hold for the Gamma of this option as well.

For Likelihood Ratio Method we used

$$\begin{aligned}\frac{\partial V_0^L}{\partial \theta}(\theta) &= \frac{\partial}{\partial \theta}\,\mathbb{E}_\theta\left[e^{-r\Delta t}V_1^L(S_1,K_0)\right]\big|_{\theta=\theta_0} \\ &= \mathbb{E}_{\theta_0}\left[e^{-r\Delta t}V_1^L(S_1,K_0)\,\frac{\partial \log p_\theta}{\partial \theta}(S_1)\Big|_{\theta=\theta_0}\right]\end{aligned} \tag{4.1}$$

A sufficient condition for this interchange according to [11] is given by the following theorem.

Theorem 1. Let $h(s)$ be a function of s where $s \in S \subseteq \mathbb{R}$, Θ a neighbourhood of θ_0, with $(p_\theta(s))_{\theta\in\Theta}$ a family of densities such that

- $p_\theta(s) > 0 \quad$ for $s \in S \quad$ and $\quad \forall\,\theta \in \Theta \;\; \mathbb{P}_\theta$-almost surely.
- $p_\theta(s)$ is differentiable with respect to θ for $s \in S$ and $\forall \theta \in \Theta \;\; \mathbb{P}_\theta$-almost surely.
- $\exists \nu$, μ such that $\frac{1}{\mu} + \frac{1}{\nu} = 1$, and a function $N(s)$ such that $\forall \theta \in \Theta$ we have
  1. $\mathbb{E}_\theta\left[|h(s)|^\mu\right] < \infty$
  2. $\mathbb{E}_\theta\left[|N(s)|^\nu\right] < \infty$
  3. $|\frac{\partial log p_\theta(s)}{\partial \theta}\frac{p_\theta(s)}{p_{\theta_0}(s)}| < N(s) \;\; \forall s \in S$

Then,

$$\frac{\partial}{\partial \theta}\mathbb{E}_\theta\left[h(s)\right]|_{\theta=\theta_0} = \mathbb{E}_{\theta_0}\left[h(s)\,\frac{\partial \log p_\theta(s)}{\partial \theta}\Big|_{\theta=\theta_0}\right]$$

In our case $p_\theta(s)$ is the transition density of S_t from time t_0 to t_1 and since we assume that the underlying process is a geometric Brownian motion this transition density is given by

$$p_\theta(s) = \frac{1}{s\sigma\sqrt{2\pi\Delta t}} e^{-\frac{1}{2}\left(\frac{\log\frac{s}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t}{\sigma\sqrt{\Delta t}}\right)^2} \tag{4.2}$$

In order to prove that these conditions hold for the Delta of the Strike Reset Put option let $h(s) := e^{-r\Delta t} V_1^L(s, K_0)$, $\theta = S_0$ and finally $\theta_0 = \tilde{S}_0$ which is fixed.

Proof. The density $p_{S_0}(s)$ clearly satisfies the first two conditions of Theorem 1 so we only need to prove the last condition.

Given the transition density one can derive that

$$\frac{\partial \log p_{S_0}}{\partial S_0}(s) = \frac{1}{\sigma^2 S_0 \Delta t}\left(\log\frac{s}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t\right) \tag{4.3}$$

and that

$$\frac{p_{S_0}(s)}{p_{\tilde{S}_0}(s)} = \frac{\frac{1}{s\sigma\sqrt{2\pi\Delta t}} e^{-\frac{\left(\log\frac{s}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t\right)^2}{\sigma^2\Delta t}}}{\frac{1}{s\sigma\sqrt{2\pi\Delta t}} e^{-\frac{\left(\log\frac{s}{\tilde{S}_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t\right)^2}{\sigma^2\Delta t}}} \tag{4.4}$$

which after doing some calculations can be written as

$$\frac{p_{S_0}(s)}{p_{\tilde{S}_0}(s)} = s^{-\frac{\zeta}{\sigma^2\Delta t}} e^{\frac{\zeta}{2\sigma^2\Delta t}\left(\log S_0 + \log\tilde{S}_0 + 2\left(r - \frac{1}{2}\sigma^2\right)\Delta t\right)} \tag{4.5}$$

where $\zeta = \log\tilde{S}_0 - \log S_0$. As a result we obtain that

$$\left|\frac{\partial \log p_{S_0}(s)}{\partial S_0}\frac{p_{S_0}(s)}{p_{\tilde{S}_0}(s)}\right| =$$

$$\left|s^{-\frac{\zeta}{\sigma^2\Delta t}} e^{\frac{\zeta}{2\sigma^2\Delta t}\left(\log(S_0) + \log(\tilde{S}_0) + 2\left(r - \frac{1}{2}\sigma^2\right)\Delta t\right)} \frac{1}{\sigma^2 S_0 \Delta t}\left(\log\frac{s}{S_0} - \left(r - \frac{1}{2}\sigma^2\right)\Delta t\right)\right| \tag{4.6}$$

Let

$$\hat{R}(S_0) = e^{\frac{\zeta}{2\sigma^2\Delta t}\left(\log(S_0) + \log_(\tilde{S}_0) + 2\left(r - \frac{1}{2}\sigma^2\right)\Delta t\right)} \frac{1}{\sigma^2 S_0 \Delta t}$$

and

$$\hat{C}(S_0) = -\log S_0 - \left(r - \frac{1}{2}\sigma^2\right)\Delta t$$

Hence equation 4.6 becomes

$$\begin{aligned}\left|\frac{\partial \log p_{S_0}(s)}{\partial S_0}\frac{p_{S_0}(s)}{p_{\tilde{S}_0}(s)}\right| &= \left|\hat{R}(S_0)\left(\log s + \hat{C}(S_0)\right)s^{-\frac{\zeta}{\sigma^2\Delta t}}\right| \\ &= \left|\hat{R}(S_0)\,s^{-\frac{\zeta}{\sigma^2\Delta t}}\log s + \hat{R}(S_0)\,\hat{C}(S_0)\,s^{-\frac{\zeta}{\sigma^2\Delta t}}\right| \\ &\leq \left|\hat{R}(S_0)\,s^{-\frac{\zeta}{\sigma^2\Delta t}}\log s\right| + \left|\hat{R}(S_0)\,\hat{C}(S_0)\,s^{-\frac{\zeta}{\sigma^2\Delta t}}\right| \\ &\leq \left|\hat{R}(S_0)\,s^{1-\frac{\zeta}{\sigma^2\Delta t}}\right| + \left|\hat{R}(S_0)\,\hat{C}(S_0)\,s^{-\frac{\zeta}{\sigma^2\Delta t}}\right| \\ &=: \left|\hat{R}(S_0)\,s^{\alpha}\right| + \left|\hat{R}(S_0)\,\hat{C}(S_0)\,s^{\gamma}\right|\end{aligned} \tag{4.7}$$

where $\alpha = 1 - \frac{\zeta}{\sigma\Delta t}$ and $\gamma = -\frac{\zeta}{\sigma\Delta t}$.

Since $\zeta = \log\tilde{S}_0 - \log S_0$ we can choose a neighbourhood of Θ of $\tilde{S}_0$ such that the following four conditions are satisfied

1. $\frac{1}{2} \leq \alpha \leq \frac{3}{2}$
2. $-\frac{1}{2} \leq \gamma \leq \frac{1}{2}$
3. $\left|\hat{R}(S_0)\right| < \bar{R} \;\; \forall S_0 \in \Theta$
4. $\left|\hat{C}(S_0)\right| < \bar{C} \;\; \forall S_0 \in \Theta$

By choosing such a neighbourhood of $\tilde{S}_0$ we hence have that

$$\left|\hat{R}(S_0)\,s^{\alpha}\right| \leq \left|\bar{R}s^{\frac{1}{2}}\right| + \left|\bar{R}s^{\frac{3}{2}}\right|$$

and

$$\left|\hat{R}(S_0)\,\hat{C}(S_0)\,s^{\gamma}\right| \leq \left|\bar{R}\bar{C}s^{-\frac{1}{2}}\right| + \left|\bar{R}\bar{C}s^{\frac{1}{2}}\right|$$

As a result we have now obtained a function $N(s)$, where

$$N(s) = \left|\bar{R}s^{\frac{1}{2}}\right| + \left|\bar{R}s^{\frac{3}{2}}\right| + \left|\bar{R}\bar{C}s^{-\frac{1}{2}}\right| + \left|\bar{R}\bar{C}s^{\frac{1}{2}}\right| \tag{4.8}$$

such that

$$\left|\frac{\partial \log p_{S_0}(s)}{\partial S_0}\frac{p_{S_0}(s)}{p_{\tilde{S}_0}(s)}\right| \leq N(s) \quad \forall s \in S \tag{4.9}$$

Now we have to check if $\exists\,\nu, \mu$ where $\frac{1}{\mu} + \frac{1}{\nu} = 1$ such that

$$\mathbb{E}\left[|h(S_1)|^{\mu}\right] < \infty \;\text{ and }\; \mathbb{E}\left[|N(S_1)|^{\nu}\right] < \infty$$

Since the underlying process S_t follows a geometric Brownian motion,

$$dS_t = rS_t dt + \sigma S_t dB_t \tag{4.10}$$

we can hence use Itôs lemma to obtain that

$$\begin{aligned} d\left(S_t\right)^{\lambda} &= \lambda S_t^{\lambda-1} dS_t + \frac{\lambda(\lambda-1)}{2} S_t^{\lambda-2} d\left\langle S_t \right\rangle_t \\ &= \lambda S_t^{\lambda-1}\left(rS_t dt + \sigma S_t dB_t\right) + \frac{\lambda(\lambda-1)}{2} S_t^{\lambda-2}\left(\sigma^2 S_t^2 dt\right) \\ &= \left(\lambda r + \frac{1}{2}\sigma^2\lambda\left(\lambda-1\right)\right) S_t^{\lambda} dt + \lambda\sigma S_t^{\lambda} dB_t \end{aligned} \tag{4.11}$$

This means that for any $\lambda \in \mathbb{R}$ the process S_t^{λ} is log-normally distributed and as a result

$$\mathbb{E}\left[S_t^{\lambda}\right] < \infty \ \ \forall t \geq 0 \tag{4.12}$$

Using this result and choosing $\nu = 2$ we now have that

$$\mathbb{E}\left[\left|N\left(S_1\right)\right|^2\right] < \infty \tag{4.13}$$

Finally for the function $h\left(s\right) = e^{-r\Delta t} V_1^L\left(s, K_0\right)$ we have that

$$\begin{aligned} V_1^L\left(s,K_0\right)^2 &\leq e^{r(T-\Delta t)} \mathbb{E}\left[\left(M_T - S_T\right)^+ | S_1 = s\right]^2 \\ &\leq \mathbb{E}\left[M_T | S_1 = s\right]^2 \\ &\leq \mathbb{E}\left[M_T^2 | S_1 = s\right] < \infty \end{aligned}$$

where M_T is the realised maximum value of the underlying price process. The first inequality arises since the value of the Strike Reset Put option should be less than the value of the corresponding look-back option. The third inequality is Jensen's inequality. The final inequality arises since S_t^2 is log normally distributed and the maximum of a log normal distribution has finite first moments. As a result if we choose $\mu = 2$ we obtain that $\mathbb{E}\left[\left|h\left(S_1\right)\right|^2\right] \leq \mathbb{E}\left[M_T^2\right] < \infty$. This finishes the proof as all the conditions of Theorem 1 are met. □

For the Gamma of the Strike Reset option we do not provide a proof for this interchange of integration and differentiation but similar arguments can be used to obtain the required result.

Chapter 5

Appendix B

In this part of the appendix we present the Matlab code used for the numerical results that we obtained at chapter 2.

Function PathGenerator takes as inputs the initial value of the underlying asset S_0 the risk free rate r, the dividends rate q, the volatility of the underlying asset *sigma*, the number of sample trajectories that will be constructed N and finally the number of exercise dates M. This function outputs are the sample trajectories as well as the Brownian increments that generate this trajectories.

```
% This function outputs the trajectories and the Brownian increments
% that are used to generate these trajectories
function [S,Z] = PathGenerator(S0,r,q,T,sigma,N,M)
% Assuming that the price process dynamics are of the form
% dSt = St*(r-q)dt + St*(sigma)dWt
% Where r is the risk free rate, q is the continuous divident rate and
% sigma is the standard deviation and all of them are constants

% dt is the time Step
dt = T/M;
% drift is the drift of the SDE of the logarith of the price
drift = ones(N,1)*(r-q-0.5*sigma^2)*dt;
% dVol is the volatility of the SDE of the logarithm of the price
dVol = sigma*sqrt(dt);
% Generates an Nx(M+1) matrix where the paths are stored
S = zeros(N,M+1);
%Initialize the asset price to be S0
S(:,1) = S0*ones(N,1);
% Generate an NxM matrix that consists of normal random variables with
% mean zero and variance 1
Z = randn(N,M);
% Generate the trajectories
for i=2:1:(M+1)
    S(:,i) = S(:,i-1).*exp(drift + dVol*Z(:,i-1));
```

```
25 end
```

The following two functions are used to evaluate the univariate and bivariate basis functions at the given points. Basis_functions_1D takes as an input a vector of sampled prices. Similarly Basis_functions_2D takes as an input a matrix that contains sampled prices and possible strike values.

```
1  % This function outputs the univariate basis functions evaluated at a given
2  % vector or point. A possible output of this function is is given below
3  function [basis_functions1D] = Basis_functions_1D(x)
4  basis_functions1D = [ones(size(x),1), x, x.^2, x.^3, x.^4,x.^5,x.^6];
5  %This function for an input of the form S(:,M), will return the
6  %following matrix
7
8  %   1   S(1,M)  S(1,M)^2 ...
9  %   1   S(2,M)  S(2,M)^2 ...
10 %   .      .       .
11 %   .      .       .
12 %   .      .       .
13 %   1   S(N,M)  S(N,M)^2 ...
```

```
1  % This function outputs the bivariate basis functions evaluated at a given
2  % grid or a grid point. A possible output of this function is is given
3  % below
4  function [basis_functions2D] = Basis_functions_2D(s,k)
5  % dimention gives the number of grid points
6  dimension = size(s,1)*size(s,2);
7  basis_functions2D = [ones(dimension,1), reshape(s,dimension,1),...
8      reshape(k,dimension,1),reshape(s.*s ,dimension,1),...
9      reshape(k.*k,dimension,1), reshape(s.*k,dimension,1),...
10     reshape(s.*s.*s,dimension,1), reshape(s.*s.*k,dimension,1),...
11     reshape(s.*k.*k,dimension,1), reshape(k.*k.*k,dimension,1),...
12     reshape(s.*s.*s.*s,dimension,1), reshape(s.*s.*s.*k,dimension,1),...
13     reshape(s.*s.*k.*k,dimension,1), reshape(s.*k.*k.*k,dimension,1),...
14     reshape(k.*k.*k.*k,dimension,1), reshape(s.*s.*s.*s.*s,dimension,1),...
15     reshape(s.*s.*s.*s.*k,dimension,1),reshape(s.*s.*s.*k.*k,dimension,1),...
16     reshape(s.*s.*k.*k.*k,dimension,1),reshape(s.*k.*k.*k.*k,dimension,1),...
17     reshape(k.*k.*k.*k.*k,dimension,1)];
18
19 %This function for an input of a grid (s,k),
20 %eg: S(:,M) with K will return the following matrix
21 %   1   S(1,M) -  K1   S(1,M)^2  K1^2   S(1,M)*K1    ...
22 %   1   S(1,M) -  K2   S(1,M)^2  K2^2   S(1,M)*K2    ...
23 %   .      .     .
24 %   .      .     .
```

```
%   1   S(1,M) - Kmxa   S(1,M)^2   Kmax^2 S(1,M)*Kmax   ...
%   1   S(2,M) -  K1    S(2,M)^2   K1^2    S(2,M)*K1    ...
%   1   S(2,M) -  K2    S(2,M)^2   K2^2    S(2,M)*K2    ...
%   1      .   - .
%   1      .      .
%   1      .      .
%   1   S(N,M)    Kmax   S(N,M)^2   Kmax^2 S(N,M)*Kmax   ...
```

Function RegressionCoefficients is the implementation of the Strike Reset Put option Valuation Algorithm. It takes as inputs the initial underlying asset price S_0, the number of sampled trajectories N, the number of possible exercise dates M, the number of strike reset rights L, the volatility of the underlying asset $sigma$, the risk free rate r, the dividends rate q, the time to maturity T and finally the initial strike value $k0$. It calculates the regression coefficients, the option price and the Delta and Gamma of this option using Likelihood Ratio Method. Finally it outputs the regression coefficients which are used by the Strike Reset Put option Forward Valuation Algorithm.

```
% This function is the Strike Reset Put option Valuation algorithm.
% Given the contract parameters it calculates the Regression Coefficients
% and the price of the option. Moreover it calculates the MCE and the
% Delta and Gamma using Likelihood ratio method.
function [b_1D,b_2D] = RegressionCoefficients(S0,N,M,L,sigma,r,q,T,k0)
% Generates sample trajectories.
[S,Z] = PathGenerator(S0,r,q,T,sigma,N,M);
% This Z_1 will be used to calculate the greeks using Likelihood Ratio
% Method.
Z_1 = Z(:,1);
% dt is the timestep.
dt = T/M;
% Increase M by 1.
M=M+1;

% K is a vector of possible strikes .
K = [max(k0,S(1,1)):1.5:4*k0,425:25:(ceil(max(max(S))/S(1,1))*k0)];
% Dimension of K.
Kdim = length(K);

% Initialize the Matrix at which the Option Value will be stored for
% avaiilabel strike resets less than L.
V = zeros(Kdim*N,L,M);

% R_1D is the Number of univariate Basis_functions that I use.
R_1D = size(Basis_functions_1D(0),2);
% R_2D is the Number of bivariate Basis_functions that I use.
R_2D = size(Basis_functions_2D(0,0),2);
```

```
29
30 % Initialize the matrix at which the regression coefficients of the
31 % bivariate basis functions will be stored to be zero.
32 b_2D = zeros(R_2D,L,M);
33
34 % Initialize the matrix at which the regression coefficients of the
35 % univariate basis functions will be stored to be zero. Note that the
36 % Regression Coefficients b_L are stored at b(:,2,i) and the Regression
37 % coefficients for b_1 are b(:,1,i), These are the only two cases for which
38 % I use univariate basis functions.
39 b_1D = zeros(R_1D,2,M);
40
41 % Generates the grid at maturity.
42 [s,k] = meshgrid(S(:,M),K);
43
44 % Evaluate the Option Value at maturity when all the exercise rights are
45 % still available.
46 V_L(:,M) = max(k0-S(:,M),0);
47 % Evaluate the Option Value at maturity when at least one the exercise
48 % right has been used.
49 V(:,:,M) = repmat(reshape(max(k-s,0),N*Kdim,1),1,L);
50
51 % Loop throught all timesteps in order to determine the regression
52 % coefficients and the value of the option at each timestep.
53 for i = M:-1:3
54
55     % Generates the grid at t_{i-1}.
56     [s,k] = meshgrid(S(:,i-1),K);
57
58     % Calculate the Regression coefficients when all the rights are still
59     % available, Note that this can be done using the Backslash command.
60     b_1D(:,2,i) = Basis_functions_1D(S(:,i-1))\V_L(:,i);
61
62     % These will be the Regression coefficients for shouting for the case
63     % that only one right left and we are interested in shouting.
64     % The regression is performed with the values of the option at maturity
65     % and not at each time step so that the error due to approximation is
66     % minimised.
67     b_1D(:,1,i) = Basis_functions_1D(S(:,i-1))\max(S(:,i-1)-S(:,M),0);
68
69     % These will be the regression coefficients for the case that no
70     % exercise right is left. The difference from the rest is that
71     % the regression is performed with the values of the option at maturity
72     % and not at each time step so that the error due to approximation is
73     % minimised.
74
75     b_2D(:,1,i) = Basis_functions_2D(s,k)\V(:,1,M);
```

```
76
77      % Calculate all the rest Regression Coefficients in a simmilar way.
78      for l = 2:L
79          b_2D(:,l,i) = Basis_functions_2D(s,k)\V(:,l,i);
80      end
81
82      % Now I evaluate the option price for different available shout
83      % opportunities at this particular timestep.
84
85      % No exercise right left
86      V(:,1,i-1) = exp(-r*dt*(M-(i-1)))*Basis_functions_2D(s,k)*b_2D(:,1,i);
87
88      % 1 Exercise Right left
89      C = exp(-r*dt)*Basis_functions_2D(s,k)*b_2D(:,2,i);
90      VSh = reshape(repmat((exp(-r*(M-i+1)*dt)*...
91      Basis_functions_1D(S(:,i-1))*b_1D(:,1,i))',Kdim,1),Kdim*N,1);
92      V(:,2,i-1) = (C < VSh).*VSh + (C>=VSh).*C;
93
94      % No exercise right used
95      C = exp(-r*dt)*Basis_functions_1D(S(:,i-1))*b_1D(:,2,i);
96      VSh = exp(-r*dt)*Basis_functions_2D(S(:,i-1),S(:,i-1))...
97          *b_2D(:,L,i).*(S(:,i-1)>k0);
98      V_L(:,i-1) = (C < VSh).*VSh + (C>=VSh).*C;
99
100     % For all the other Cases
101     for l = 3:L
102             C = exp(-r*dt)*Basis_functions_2D(s,k)*b_2D(:,l,i);
103             VSh = exp(-r*dt)*Basis_functions_2D(s,s)*b_2D(:,l-1,i).*...
104                 reshape(s>k,N*Kdim,1);
105             V(:,l,i-1) = (C < VSh).*VSh + (C>=VSh).*C;
106     end
107 end
108 %Calculates the option Price.
109 Price = exp(-r*dt)*sum(V_L(:,2))/N
110
111 % Calculates the MCE
112 Std = sqrt((1/N^2)*sum((exp(-r*dt)*V_L(:,2)-Price).^2))
113
114 % Calculates the Delta of the option
115 Delta = 1/(N*S0*sigma*sqrt(dt)) * exp(-r*dt)*V_L(:,2)'*Z_1
116
117 % Calculates the Gamma of the option
118 Gamma = 1/N * exp(-r*dt) *V_L(:,2)'*((Z_1.^2 -1)/(sigma^2*S0^2*dt) -...
119     Z_1/(sigma*S0^2*sqrt(dt)))
```

Function ForwardValuation is the implementation of the Strike Reset Put option Forward valuation Algorithm. Simillarly with the previous functions it takes as inputs the initial underlying asset price S_0, the number of sampled trajectories N, the number of possible exercise dates M, the number of strike reset rights L, the volatility of the underlying asset $sigma$, the risk free rate r, the dividends rate q, the time to maturity T and finally the initial strike value $k0$. Moreover it takes as an input the regression coefficients β_i^l for $1 \leq l \leq L-1$ and β_i^L. It calculates all the sensitivities described in the first chapter as well as the price of the option using the Forward Valuation Algorithm and their Monte Carlo Errors.

```
% This function is the implementation of the Strike Reset Puto option
% Forward Valuation Algorithm. It takes as input the contract parameters
% and the regression coefficients and it outpouts the price with the MCE
% and all the Sensitivities with their MCE.
function [Price] = ForwardValuation(S0,N,M,L,sigma,r,q,T,k0,b_1D,b_2D)

% dt is the timestep.
dt = T/M;
% Generates sample trajectories.
[S,Z] = PathGenerator(S0,r,q,T,sigma,N,M);
% M is increased by 1.
M = M+1;

% Initialise Counter to a vector of ones. Counter is the variable that
% informs as when and if a strike reset takes place.
Counter = ones(N,1);

% Initialise all the strikes to the initial strike k0.
strikes = ones(N,1)*k0;

% Initialise Vega_Sum to a zero vector. Vega_Sum is used to obtain the vega
% of the option.
Vega_Sum = zeros(N,1);

% The payoff function.
payoff = @(x,k) max((k-x),0);

% Loop through all the trajectories.
for j = 1:N

    % Set the exercise rights left to be equal to L foe each trajectory.
    exercise_rights_left = L;

    % Loop through all the possible timesteps going forward.
    for i = 3:M
```

```
36          % Check if there is any available right.
37          if exercise_rights_left>0
38             % Check if no right has been used.
39             if exercise_rights_left == L
40                  % Evaluate the continuation value.
41                  C = exp(-r*dt)*Basis_functions_1D(S(j,i-1))*b_1D(:,2,i);
42                  % Evaluate the Shouting Value
43                  VSh = exp(-r*dt)*Basis_functions_2D(S(j,i-1),S(j,i-1))*...
44                      b_2D(:,L,i);
45                      % Check if the Shouting value is greater than the
46                      % continuation value and if the curent price is greater
47                      % then the current strike
48                      if C ≤ VSh && S(j,i-1)>strikes(j)
49                          % Set the new strike equal to the current asset
50                          % price. Set the counter equal to i-1 end reduce
51                          % the number of available exercise rights by 1
52                          strikes(j) = S(j,i-1);
53                          Counter(j) = i-1;
54                          exercise_rights_left = exercise_rights_left - 1;
55                      end
56              % Check if one reset right is available and proceed simillarly
57              % with the Similarly with the case that no right has been used
58              elseif exercise_rights_left == 1
59                  C = exp(-r*dt)*Basis_functions_2D(S(j,i-1),strikes(j))*...
60                      b_2D(:,2,i);
61                  VSh = exp(-r*(M-i+1)*dt)*Basis_functions_1D(S(j,i-1))*...
62                      b_1D(:,1,i);
63                       if C ≤ VSh && S(j,i-1)>strikes(j)
64                          strikes(j) = S(j,i-1);
65                          Counter(j) = i-1;
66                          exercise_rights_left = exercise_rights_left - 1;
67                       end
68              % For all other cases proceed simillarly
69              % with the Similarly with the case that no right has been used
70              else
71                  C = exp(-r*dt)*Basis_functions_2D(S(j,i-1),strikes(j))*...
72                      b_2D(:,exercise_rights_left+1,i);
73                  VSh = exp(-r*dt)*Basis_functions_2D(S(j,i-1),S(j,i-1))*...
74                      b_2D(:,exercise_rights_left,i);
75                      if C ≤ VSh && S(j,i-1)>strikes(j)
76                          strikes(j) = S(j,i-1);
77                          Counter(j) = i-1;
78                          exercise_rights_left = exercise_rights_left - 1;
79                      end
80              end
81          end
82      end
```

```
83
84      % Calculate Vega_sum for each path which is used to calculate the vega
85      % of the option
86      Vega_Sum(j) = ((sqrt(dt)*sum(Z(j,1:Counter(j)-1)) - sigma...
87       *(Counter(j)-1)*dt)*S(j,Counter(j))- (sum(Z(j,:))-sigma*T)*S(j,M))*...
88       (strikes(j)>S(j,Counter(j)))*(Counter(j)~=1)-(sqrt(dt)*sum(Z(j,:))-...
89       sigma*T)*S(j,M)*(k0>S(j,M))*(Counter(j)==1);
90  end
91
92  % Calculates the price of the option
93  Price = exp(-r*T)*sum(payoff(S(:,M),strikes))/N;
94  % Calculates the MCE
95  Std_Price = sqrt((1/N^2)*sum((exp(-r*T)*payoff(S(:,M),strikes)-Price).^2))
96  % Calculates the Delta of the option using Likelihood Ratio Method
97  Delta_likelihood =(exp(-r*T)/N)*sum((strikes-S(:,M)).*(strikes>S(:,M)).*...
98      (Z(:,1)))/(S(1,1)*sigma*sqrt(dt))
99  % Calculates the MCE of the Delta estimator
100 Std_Delta_likelihood = sqrt((1/N^2)*sum((exp(-r*T)*(strikes-S(:,M)).*...
101   (strikes>S(:,M)).*(Z(:,1))/(S(1,1)*sigma*sqrt(dt))-Delta_likelihood).^2))
102 % Calculates the Delta of the option using pathwise sensitivities approach
103 Delta_pathwise = (exp(-r*T)/N)*sum((strikes-S(:,M)).*(strikes~=k0).*...
104     (strikes>S(:,M)) - S(:,M).*(strikes>k0).*(strikes == k0))/S(1,1)
105 % Calculates the MCE for the Delta estimator
106 Std_Delta_pathwise = sqrt((1/N^2)*sum((exp(-r*T)*((strikes-S(:,M)).*...
107     (strikes~=k0).*(strikes>S(:,M)) - S(:,M).*(strikes>k0).*...
108     (strikes == k0))/S(1,1) - Delta_pathwise).^2))
109 % Calculates the Gamma of the option using Likelihood Ratio Method
110 Gamma_likelihood =(exp(-r*T)/N)*sum((strikes-S(:,M)).*(strikes>S(:,M)).*...
111     ((Z(:,1).^2-1)/(sigma^2*S(1,1)^2*dt)-Z(:,1)/(sigma*S(1,1)^2*sqrt(dt))))
112 % Calculates the MCE for the Gamma estimator
113 Std_Gamma_likelihood = sqrt((1/N^2)*sum((exp(-r*T)*(strikes-S(:,M)).*...
114     (strikes>S(:,M)).*((Z(:,1).^2-1)/(sigma^2*S(1,1)^2*dt)-Z(:,1)/...
115     (sigma*S(1,1)^2*sqrt(dt)))-Gamma_likelihood).^2))
116 % Calculates the Vega of the optin using Pathwise sensitivities approach
117 Vega_pathwise = exp(-r*T)*sum(Vega_Sum)/N
118 % Calculates the MCE of the Vega estimator
119 Std_Vega_Sum = sqrt((1/N^2)*sum((exp(-r*T)*Vega_Sum - Vega_pathwise).^2))
120 % Calculates the Rho of the otpion using Pathwise sensitivities approach
121 Rho_pathwise = 1/N*exp(-r*T)*sum(strikes.*(Counter~=1).*...
122   ((Counter-1)*dt-T).*(strikes>S(:,M))-T*k0*(Counter==1).*(strikes>S(:,M)))
123 % Calculates the MCE for the Rho estimator
124 Std_Rho_pathwise = sqrt((1/N^2)*sum((exp(-r*T)*(strikes.*(Counter~=1).*...
125     ((Counter-1)*dt-T).*(strikes>S(:,M)) - T*k0*(Counter==1).*...
126     (strikes>S(:,M))) - Rho_pathwise).^2))
```

Bibliography

[1] M.Broadie and P.Glasserman, (1997), *Monte Carlo Methods for Pricing High-Dimentional American Options*, Columbia University

[2] T.H.F Cheuk and T.C.F Vorst,(1997), *Shout Floors*, Net Exposure 2

[3] E.Clement, D.Lamberton, and P.Protter, (2002), *An analysis of a least squares regression algorithm for American option pricing*, Finance and Stochastics 6:449â471.

[4] J.C.Cox, S.Ross and M.Rubinstein, (1979), *Option Pricing: a simplified approach*, Journal of Financial Economics Vol.7, pp.229-263

[5] M.Dai, Y.K.Kwok and L.Wu, (2003), *Options with Multiple Reset Rights*,International Journal of Theoretical and Applied Finance, Vol.6, No.6, pp.637-653

[6] M.Dai, Y.K.Kwok and L.Wu, (2004), *Optimal Stopping Policies of options with Strike Reset Right*, Mathematical Finance, Vol.14, No.3, pp.383-401

[7] U.Dorr, (2003), *Valuation of Swing Options and Examination of Exercise Strategies by Monte Carlo Techniques*, University of Oxford, MSc Thesis.

[8] P.Glasserman, (2004), *Monte Carlo Methods in Financial Engineering*, Springer.

[9] P.Glasserman and B. Yu, (2003), *Number of Paths Versus Number of Basis Functions in American Option Pricing*, Columbia University

[10] S.Gray and R.Whaley, (1999), *Reset put options: valuation, risk characteristics, and an application*, Australian Journal of Management 24, pp.1-20

[11] L.G.Gyurkó, (2010), *Numerical Methods 2*, Lecture Notes, University of Oxford

[12] T.Howard, (2009), *Longstaff Schwartz Pricing of Bermudan Options and their Greeks*, University of Oxford, MSc Thesis.

[13] P.Jaillet, E.Ronn, and S.Tompadis, (2004), *Valuation of commodity-based swing options.*, Management Science 50, pp.909-921

[14] T.Kluge, (2006), *Pricing Swing Options and other Electricity Derivatives.*, PhD thesis, University of Oxford.

[15] Y.K.Kwok, Options, Financial Engineering and Risk Management, Routledge

[16] F.A.Longstaff and E.S.Schwartz, (2001), *Valuing American Options by Simulation: A simple least-squares approach*, Review of Fin. Studies, 14 , pp. 113-147.

[17] N.Meinhausen and B.M.Hambly, (2004), *Monte Carlo Methods for the Valuation of Multiple Exercise Opportunities*, Mathematical Finance, Vol.14, No.4, pp.557-583.

[18] J.N.Tsitsiklis and B. Van Roy, (2001), *Regression Methods for Pricing Complex American-Style Options*, IEEE Trans. on Neural Networks, Vol. 12, No. 4, pp. 694-703.

[19] H.Windcliff, P.A.Forsyth and K.R.Vetzal, (1999), *Shout Options: A Framework For Pricing Contracts Which Can Be Modified by the Investor*, Journal of Computational and Applied Mathematics 134, pp.213-241

[20] L.Yudaken, (2010), *Numerical Pricing of Shout Options*, University of Oxford, MSc Thesis.