Applied Mathematics
&
Statistics

514 Report - Fall 2022

An Investigation into the Longstaff-Schwartz Method for Pricing American Options

Jason Bohne

originally authored by William Gustafsson

Contents

1 Introduction

Determining the fair value of derivative contracts is a common problem faced by traders and portfolio managers in the financial industry. Known throughout the applied mathematics and quantitative finance community, the solution to the Black Scholes Equation provides an analytical expression for the valuation of European options. While significant, the theory behind the Black-Scholes can not be applied to the valuation of exotic contracts in which early exercise is permitted. The most well-known style of the above structure is the American option with it's value function $u(S,t)$ for the put following the boundary conditions below

$$u(S,t) \geq max(K - S, 0)$$

$$u(S,T) = max(K - S, 0)$$

The vast majority of literature on pricing exotic derivatives focuses on numerical and simulation methods. In the latter, Monte-Carlo-style simulations are quite prevalent with research focusing on the statistical and computational design of the algorithm. This report will focus on the academic paper **Evaluating the Longstaff-Schwartz Method for Pricing American Options** by [Gustafsson, 2015]. Here Gustafsson presents an extension to the Longstaff-Schwartz Method, which falls within the class of Least-Squares Monte-Carlo methods.

The report will proceed as follows. In section 2 the LSM algorithm will be discussed with an emphasis on the underlying assumptions. From there, numerical results will be presented and benchmarked against the original results which the author included, respectively in section 3. In section 4 a thorough analysis of the exercise boundary will be provided as insight into uncertainty quantification. Finally, key takeaways and thoughts will be shared in section 5, the conclusion. The appendix includes access to an implementation of the method in python along with references.

2 Longstaff-Schwartz Method

First introduced by [Longstaff and Schwartz, 2001] in **Valuing American Options by Simulation: A Simple Least-Squares Approach** the Longstaff-Schwartz Method utilizes least squares to estimate the continuation value of an American option across sampled realizations of the underlying. From here the expected continuation value is compared against the known exercise value to determine whether it is optimal for early exercise. As standard in Monte-Carlo simulations, a large number of samples are generated allowing one to utilize the central limit theorem to use the sample mean as an estimate for the expected value.

2.1 Original LSM Algorithm

One of the critical assumptions of the algorithm is the dynamics of the path which the underlying can take. Both in the original implementation and the extension the underlying is assumed to follow a Geometric Brownian Motion of the following form:

$$S(T) = S(0)e^{(\mu - \frac{1}{2}\sigma^2)t + (t)}$$

Where $\mu = r$ is the drift term and σ is the volatility parameter. The original algorithm begins with the generation of M realizations for which the underlying can traverse. Starting at expiry, a dynamic programming approach is then applied to iteratively step backward; bookkeeping realizations in which it is optimal to exercise the option early.

At each timestep t_i the LSM relies on least squares to estimate the continuation value as a linear model of basis functions, L, applied on the discounted underlying price, X_i. One set of basis functions can be the Laguerre polynomials, which are solutions to the Laguerre equations, and can be generated following initial conditions and recursive relation: (CITE)

$$L_0(x) = 1, \ L_1(x) = 1 - x$$

$$L_{k+1}(x) = \frac{(2k + 1 - x)L_k(x) - kL_{k-1}(x)}{k + 1}$$

While [Gustafsson, 2015] only tested basis functions generated from Laguerre polynomials, [Longstaff and Schwartz, 2001] also suggested the use of Hermite, Legendre, and Chebyshev polynomials as well. Following suit of ordinary least squares, an optimal coefficient vector $\hat{\beta}$ is solved which minimizes the mean squared error of the linear model relating discounted payoffs Y and current price X.

$$Y = \beta L(X)$$

For each realized path $S_{j=1...M}$, a regression is performed at $t_{i=1...N}$. From here the estimator can be applied to compute $\hat{Y}_i = \hat{\beta}_i L(X_i)$. Note $\hat{Y}_i$ is the expected continuation value of the option given the underlying price X_i. For each path realization, S_j, the expected continuation value at t_i can be compared against the known immediate exercise value at t_i to determine whether early exercise is optimal. This procedure is repeated backward in time for all $t_{i=N-1...1}$ to solve for the optimal exercise of the contract on each sampled path. Once determined the sample mean of the discounted option value can be calculated, which can be interpreted as the current valuation.

2.2 Algorithm Extension

The original LSM algorithm is extended within [Gustafsson, 2015] by utilizing a well-known Brownian excursion when sampling realizations; a Brownian bridge. Achieved by conditioning a Brownian motion on it's endpoints, the distribution of a Brownian bridge can be derived as the following:

$$(X(T_i)|X(T_{i+1}) = x_{t_{i+1}} X(0) = 0) \sim \ N(\frac{t_i x_{t_{i+1}}}{t_{i+1}}, \sigma^2 \frac{t_i(t_{i+1} - t_i)}{t_{i+1}})$$

Note as for all Brownian motion the property $X(0) = 0$ holds, the second condition can be dropped without changing the distribution. The implication of this transformation is that each realization can be sampled backwards in time with the following recursion starting with the expiry price, $X(T)$.

$$X(t_i) = \frac{t_i X(t_{i+1})}{t_{i+1}} + \sigma \sqrt{\frac{t_i \triangle t}{t_{i+1}}} Z$$

This extension drastically reduces the amount of memory required by the algorithm as at a given time t_i only two observations of a realization need to be stored; X_{t_i} and $X_{t_{i-1}}$

3 Numerical Results

3.1 Parameter Set

Simulations to price the American put were included within [Gustafsson, 2015] across a parameter set of three strike values (ITM, ATM, OTM), 1-4 Laguerre basis functions, and a range of 10-200 discretization steps in time. Given the sample mean μ_{θ_i} for each simulation parameterized by θ_i, the value of the option is taken to be the sample mean of the K simulations.

$$\mu_0 = \frac{1}{K}\sum_{i=1}^{K}\mu_{\theta_i}$$

Below is the full parameter set used within simulations.

Parameter	**Value**
K	100
σ	0.15
r	0.03
T	1
N	$10 \ldots 190$
$S0$	90, 100, 110
L	$L_{i=1\ldots4}$

3.2 Simulations of the American Put

Original simulations were performed to price the American put option. Each row in the table corresponds to a time discretization N_i with the columns referring to the American put value when the current price of the underlying is across a range of prices. Overall the sample mean of the simulations was within $4 * 10e^{-2}$ of what was reported within [Gustafsson, 2015], with the largest discrepancy occurring at the at-the-money contract. Interestingly enough the empirical distribution of the sample means across the set of time discretizations is heavy-tailed. Below are histograms portraying such.

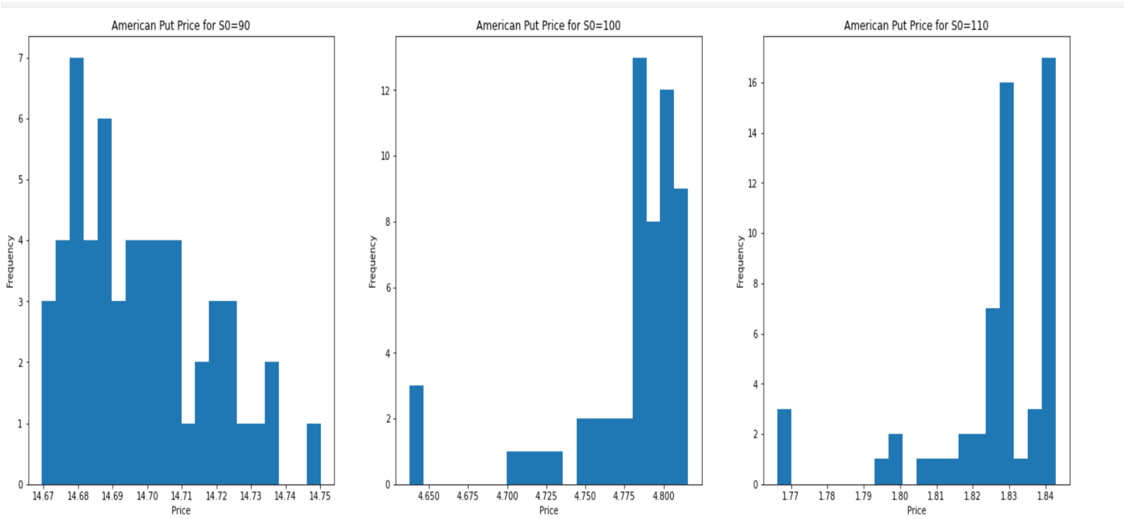

Figure 1: Empirical Distributions of Sample Means of Varying Time Discretizations

N	S0		
	S0=90	S0=100	S0=110
10	10.46	4.64	1.77
20	10.65	4.72	1.80
30	10.63	4.75	1.81
40	10.68	4.76	1.82
50	10.75	4.77	1.82
60	10.78	4.78	1.83
70	10.79	4.79	1.83
80	10.79	4.79	1.83
90	10.77	4.79	1.83
100	10.73	4.80	1.83
110	10.74	4.79	1.84
120	10.76	4.80	1.84
130	10.77	4.80	1.84
140	10.78	4.80	1.84
150	10.74	4.80	1.84
160	10.75	4.80	1.84
170	10.74	4.81	1.84
180	10.76	4.80	1.84
190	10.76	4.81	1.84
Sample Mean fixed S0	10.728	4.778	1.826
[Gustafsson, 2015] value	10.726	4.820	1.828
Error from[Gustafsson, 2015]	-0.002	0.042	0.002

Table 1: Sample Mean of Discounted American Put Value over Parameter Set

3.3 Simulations of the American Call

It is natural to explore the algorithm's performance when pricing the American call, as it is never optimal to exercise an American call on a non-dividend-paying stock before expiry. The implications of this allow the valuation of an American call as a European call using the Black-Scholes Equation. Moreover from this argument, the exercise boundary for the American call can be derived as the positive domain for $0 \leq t_i < T$ with $S_T = K$ at $t_i = T$. Focusing on the

Black-Scholes partial differential equation, given S_t, risk free rate r, underlying volatility σ, and time to expiration t the equation is of the following form:

$$\frac{\partial u}{\partial t} + rS\frac{\partial u}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 u}{\partial S^2} = 0$$

.

The solution of this differential equation for the European call is:

$$S\phi(d_1) - Ke^{-rt}\phi(d_2)$$

where $d_1 = \frac{\ln \frac{S}{K} - T(r+\frac{1}{2}\sigma^2)}{\sigma\sqrt{T}}$ and $d_2 = d_1 - \sigma\sqrt{T}$

The LSM extension approximates the true value of the American call within accuracy of $2 * 10e^{-2}$. Across the simulations, the estimates tend to over-approximate the true value of the call, which is worth further investigation.

N	S0		
	S0=90	S0=100	S0=110
10	2.73	7.46	14.70
20	2.73	7.45	14.68
30	2.73	7.45	14.69
40	2.73	7.45	14.69
50	2.74	7.46	14.69
60	2.74	7.47	14.69
70	2.75	7.46	14.69
80	2.74	7.47	14.70
90	2.73	7.47	14.70
100	2.74	7.48	14.70
110	2.74	7.49	14.72
120	2.75	7.49	14.72
130	2.73	7.46	14.71
140	2.73	7.47	14.71
150	2.73	7.48	14.71
160	2.73	7.45	14.71
170	2.73	7.47	14.70
180	2.73	7.47	14.68
190	2.72	7.46	14.70
Sample Mean Fixed S0	2.733	7.465	14.698
Black Scholes Value	2.758	7.485	14.702
Error from Black Scholes	0.025	0.02	0.004

Table 2: Sample Mean of Discounted American Call Value over Parameter Set

4 Further Analysis

4.1 Exercise Boundary on Local Extrema

In [Gustafsson, 2015], a qualitative graph of the exercise boundary is provided for the American put. While informative, this is not generated from the results of the algorithm. In order to derive an exercise boundary from the LSM extension for the American put and call, the underlying prices must first be stored at each timestep working backward in time from $t_i | i = \{N-1 \ldots 1\}$.

Given the underlying realizations X_i and the discounted payoffs Y_i the LSM extension computes the exercise value E_i and the continuation value C_i of the American option at each timestep t_i. From here the subset of realizations that are desirable to exercise the option early can be defined as

$$Exp = \{i = 1 \ldots M | E_i > C_i\}$$

To form the exercise boundary for the American put let's define lower and upper bounds from the local extrema at t_i

$$U = \inf\{X_i | i \notin Exp\}$$

$$L = \sup\{X_i | i \in Exp\}$$

Then intuitively the exercise boundary EB_t should satisfy the following condition.

$$L_{t_i} \leq EB_{t_i} \leq U_{t_i}$$

.

Using the local extrema, let's plot the exercise boundary for the American put, and the adjusted boundary for the American call. Note the box constraints defined above are not met. Moreover the exercise boundary results are primarily composed of noise, where at many points in time the exercise boundary is nonexistent ($U_{t_i} < EB_t < L_{t_i}$).

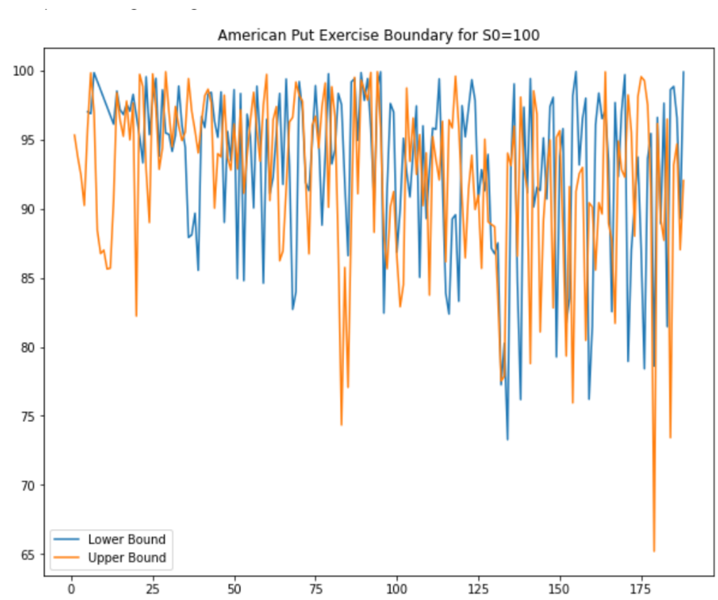

Figure 2: Exercise Boundary on Extrema for American put

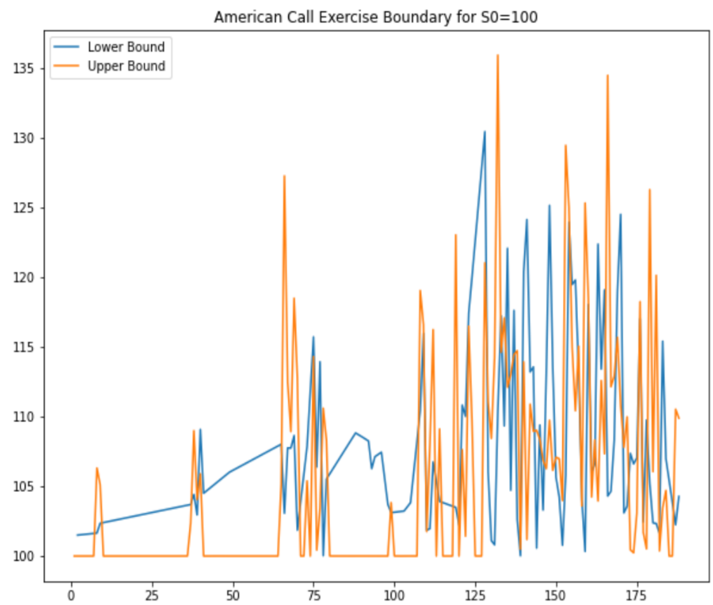

Figure 3: Exercise Boundary on Extrema for American call

4.2 Exercise Boundary on Conditional Sample Mean

However recall that at each timestep t_i, the LSM algorithm utilizes conditional expectation on the underlying value of realization X_i for determining the continuation value of the option. Instead of taking the extrema to formulate the exercise boundary, let's take the sample mean of the underlying values conditional on whether the option is exercised on the realizations at the timestep t_i. Redefining the lower and upper bounds, as below, the resulting exercise boundary on expectation satisfies the prespecified box conditions.

$$L = E\{X_i | i \in Exp\}$$

$$U = E\{X_i | i \notin Exp\}$$

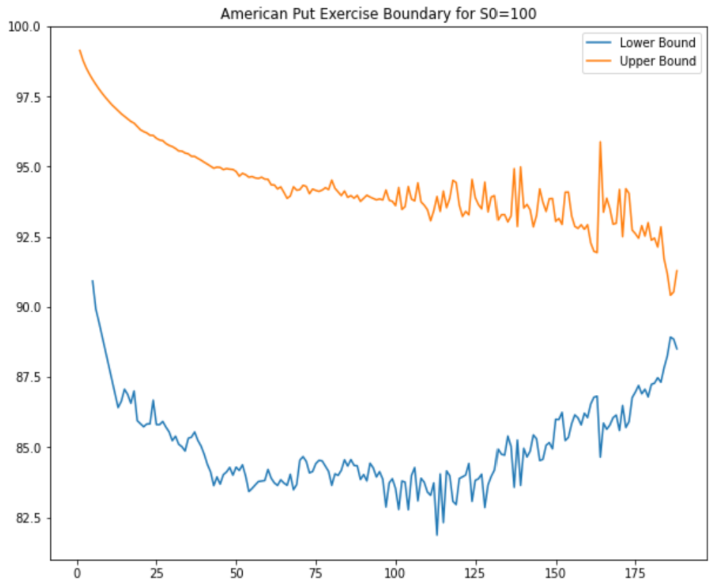

Figure 4: Exercise Boundary on Conditional Expectation for American put

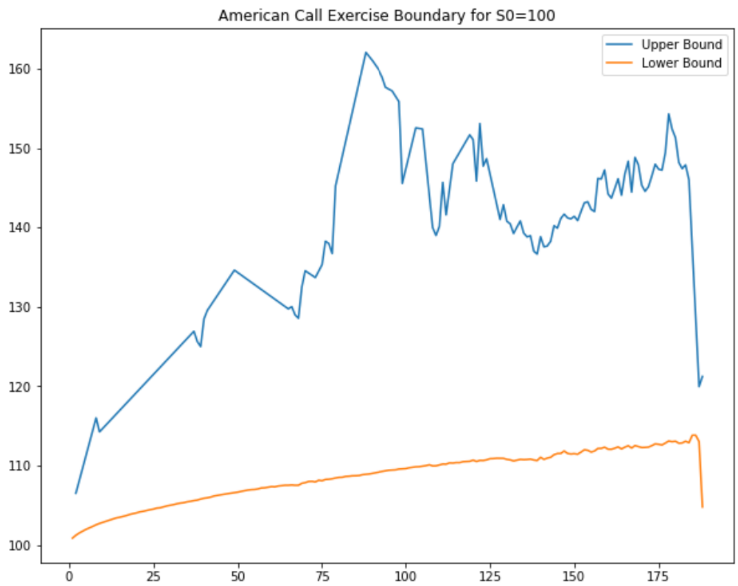

Figure 5: Exercise Boundary on Conditional Expectation for American call

As previously mentioned, the exercise boundary for the put and call cannot be determined uniquely using only the LSM algorithm as any monotonically increasing or decreasing function between the box constraints will be a solution. For the American put and call, respectively the upper and lower bounds are more stable and respect smoothness and monotonicity more so than the lower and upper bounds.

One particular reason for this observation can be due to the fact the set of realizations that result in optimal exercise values E_i relative to the continuation values C_i are much smaller in magnitude than the alternative. Given fewer available samples, it is intuitive that the resulting conditional sample mean is less robust to noise and outliers.

5 Conclusion

In sum, this report provides an overview on **Evaluating the Longstaff-Schwartz Method for Pricing American Options** originally authored by [Gustafsson, 2015]. Designed as an extension to the original Longstaff-Schwartz approach, this method computes a present value for exotic derivatives with the ability for early exercise. Structurally, the algorithm iteratively samples observations backward in time using the theoretical results of Brownian bridges. Utilizing a dynamic programming framework, the continuation values are estimated at each point in time on the set of realizations using least squares and then compared with known immediate exercise values to determine early exercise. The original numerical simulations and results are reproducible with [Gustafsson, 2015] to an accuracy of $4 * 10e^{-2}$, however, the original exercise boundary figure is inaccurate. Careful consideration is required to generate the exercise boundary from the algorithm, as using local extrema arises in an inconsistent solution. Instead bounding the true exercise boundary using the conditional sample mean seems to yield better results from the simulations performed, Further investigation could be interesting in both the distribution of sample means conditioned across distinct parameter values, in addition to improvements in estimating the true exercise boundary.

6 Appendix

6.1 Code Access

The main function used in the above simulations can be found below. High-level the algorithm iterates backward in time estimating the discounted payoffs of the derivative accounting for early exercise via a dynamic programming framework. Specifically, the dynamic programming arises in how the indexing of payoffs is stored with least squares relating the underlying prices to the discounted payoffs at each time t_i.

```
def LSM(T, r, sigma, K, S0, N, M, k, right=1, seed=1234):
    """
    Longstaff-schwartz method: Gustafsson Implementation
    T: Expiration Time
    r: interest rate
    sigma: underlying vol
    K: strike
    S0: inital underlying
    N: # timesteps from t=0 to t=T
    M: # realizations (even)
    k: basis functions
    seed: random seed for reproducibility
    """

    np.random.seed(seed)
    t = np.arange(0, T, T / N)
    z = np.random.normal(size=(math.floor(M / 2), 1))
    w = (r - (sigma ** 2 / 2)) * T + sigma * np.sqrt(T) * np.vstack([z, -z])
    S = S0 * np.exp(w)
    sup_eb = {}
    inf_eb = {}

    if right == 1:  # put payoffs
        P = np.maximum.reduce([K - S, np.zeros(S.shape)]).reshape(1, -1)[0]
    elif right == 0:  # call payoffs
        P = np.maximum.reduce([S - K, np.zeros(S.shape)]).reshape(1, -1)[0]
    else:
        return

    for i in range(N - 2, 0, -1):
```

```
    z = np.random.normal(size=(math.floor(M / 2), 1))
    w = (t[i] * w) / t[i + 1] + sigma * np.sqrt(
        (T / N) * t[i] / t[i + 1]
    ) * np.vstack(
        [z, -z]
    )  # brownian bridge sampling
    S = S0 * np.exp(w)
    if right == 1:
        index = np.where(K > S)[0]
    elif right == 0:
        index = np.where(S < K)[0]
    X = S[index].reshape(1, -1)[0]  # prices and payoffs itm
    Y = P[index] * np.exp(-r * T / N).reshape(1, -1)[0]
    A = laguerre_basis(X, k)  # laguerre basis functions
    x, resid, rank, s = np.linalg.lstsq(A, Y, rcond=None)
    C = A.dot(x.reshape(1, -1)[0])  #  estimated continuation using ols

    if right == 1:
        E = K - X
    elif right == 0:
        E = X - K

    exP = np.where(C <= E)[0]  # indices better to exercise
    index_exP = index[exP]
    if len(exP) > 0:
        sup_price = np.mean(X[exP])
        sup_eb[i] = sup_price

    non_exP = np.where(C >= E)[0]
    if len(non_exP) > 0:
        inf_price = np.mean(X[non_exP])
        inf_eb[i] = inf_price

    rest = np.setdiff1d(
        np.arange(0, M), index_exP
    )  # new realizations optimal to exercise
    P[index_exP] = E[exP]  # update payoffs
    P[rest] = P[rest] * np.exp(-r * T / N)
u = np.mean(P * np.exp(-r * T / N))
```

```
    return u, sup_eb, inf_eb
```

The full repository of all scripts and data utilized in this project can be accessed on **GitHub**.

References

[Gustafsson, 2015] Gustafsson, W. (2015). Evaluating the longstaff-schwartz method for pricing of american options.

[Longstaff and Schwartz, 2001] Longstaff, F. and Schwartz, E. (2001). Valuing american options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14:113–47.