Politecnico di Torino

POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Regression Monte Carlo Methods for Pricing American Put Options

Relatore
Prof. Tommaso Vanzan

Candidata
Emanuela Girlando

Anno Accademico 2025-2026

Contents

List of Figures

Abstract

This thesis studies regression-based Monte Carlo methods for pricing American-style options, with a focus on the Least Squares Monte Carlo (LSM) algorithm of Longstaff and Schwartz (2001). After reviewing the theoretical foundations, the LSM algorithm is implemented and validated on single-asset American put options, with particular attention to basis function choice, antithetic variates and convergence diagnostics with respect to the number of paths and exercise dates.

The method is then extended to American basket put options written on multiple correlated underlying assets. Two regression specifications for the continuation value are compared: a *univariate* specification, which regresses on the scalar basket value only, and a *multivariate* specification, which regresses on the full vector of constituent asset prices. Across a range of moneyness levels, correlation structures and basket dimensions, the two specifications yield statistically indistinguishable prices, indicating that the scalar basket value already captures the information relevant to the exercise decision. The multivariate specification therefore offers no measurable pricing benefit while incurring a higher computational and estimation cost.

The analysis is complemented by an empirical application to a basket of four Big Tech equities (Amazon, Apple, Microsoft, NVIDIA), using historical price data to calibrate volatilities, correlations and basket weights under both equal-weight and value-weight schemes. The results obtained on real market data are consistent with those of the synthetic basket experiments, confirming that the univariate regression specification provides an adequate and computationally efficient basis for pricing American basket options via LSM.

1 Options

An option is a financial derivative contract that grants the holder the right, but not the obligation, to buy or sell an underlying asset $S(t)$ at a predetermined future date T, known as the *maturity*, and at a predetermined price K, called the *strike price*.
Exercising an option means making use of this contractual right, converting the option position into a position in the underlying asset.
The exercise right belongs exclusively to the holder (or buyer) of the option, whereas the writer (or seller) of the option is obliged to accept the holder's decision and take the opposite position in the underlying market. In compensation for this asymmetry of rights and obligations, the holder pays an *option premium* at the time of purchase; which represents the price of the option regardless of whether the contract is exercised or expires worthless.

Options differ in their exercise style. The two most common exercise styles are European and American options.
A **European option** may be exercised only at maturity T, while an **American option** may be exercised at any time $t \leq T$. In the American case, a central problem is to determine the optimal stopping time τ^* at which the holder should exercise in order to maximize the expected discounted payoff $\mathbb{E}[e^{-r\tau^*}\phi(S(\tau^*))]$.
A **Bermudan option** allows early exercise as well, but only at a discrete set of predetermined dates $0 < t_1 < t_2 < \cdots < t_m = T$, thus lying between European and American options in terms of exercise flexibility.

Options can additionally be classified according to the holder's right. In the case of a European option:

- A **call option** gives the right to buy the underlying asset $S(t)$ at price K at time T,
- A **put option** gives the right to sell the underlying asset $S(t)$ at price K at time T.

As previously discussed, for European options the exercise time coincides with the maturity T.
By combining the investor's position in the contract (long or short) and the payoff type (call or put), one obtains four basic positions: long call, short call, long put and short put.

1.1 Payoff and Profit

From the holder's perspective, a call option is exercised at maturity only if the market price of the underlying asset $S(T)$, exceeds the strike price K. In this case, the holder acquires the asset at price K and may immediately sell it at the higher market price, generating a positive payoff. Conversely, if $S(T) \leq K$, it is optimal to let the call expire without exercise, since purchasing the asset on the market would be more convenient than exercising the option.

Formally, let $\phi(S(T))$ denote the payoff at time T, P the profit and let c and p be, respectively, the call and put premium paid by the buyer. In the following, payoff and

profit functions are derived for all four basic positions (long/short call and long/short put).

Long call: buy the right to buy $S(T)$ at time T and at price K, thus the payoff at maturity is:

$$\phi(S(T)) = (S(T) - K)^+ = \max(0, S(T) - K)$$

and the profit is:

$$P = (S(T) - K)^+ - c = \begin{cases} S(T) - K - c, & \text{if } S(T) > K \\ -c, & \text{otherwise} \end{cases}$$

Short call: sell the right to buy $S(T)$ at time T and at price K, thus the payoff at maturity is:

$$\phi(S(T)) = -(S(T) - K)^+ = -\max(0, S(T) - K)$$

and the profit is:

$$P = c - (S(T) - K)^+ = \begin{cases} c - S(T) + K, & \text{if } S(T) > K \\ c, & \text{otherwise} \end{cases}$$

A put option is exercised when the market price of the underlying at maturity falls below the strike price, $S(T) < K$. The holder benefits from selling the asset at strike price while its market value is lower. If $S(T) \geq K$ the option expires worthless.

Long put: buy the right to sell $S(T)$ at time T and at price K, thus the payoff at maturity is:

$$\phi(S(T)) = (K - S(T))^+ = \max(0, K - S(T))$$

and the profit is:

$$P = (K - S(T))^+ - p = \begin{cases} K - S(T) - p, & \text{if } S(T) < K \\ -p, & \text{otherwise} \end{cases}$$

Short put : sell the right to sell $S(T)$ at time T and at price K, thus the payoff at maturity is:

$$\phi(S(T)) = -(K - S(T))^+ = -\max(0, K - S(T))$$

and the profit is:

$$P = p - (K - S(T))^+ = \begin{cases} p - K + S(T), & \text{if } S(T) < K \\ p, & \text{otherwise} \end{cases}$$

These relationships highlight the intrinsic asymmetry between buyers and writers of options. For long positions, the maximum loss is limited to the premium paid, while the potential gain is unbounded for a long call and bounded above by $(K - S(T) - p)$ for a long put.
For short positions, the maximum profit equals the premium received, whereas the potential losses can be substantial and, in the case of a short call, theoretically unbounded.

In Figure 1 the profit at maturity is represented as a function of the underlying price $S(T)$ for the four basic option positions. The Break-Even Price (BEP) is defined as the value of $S(T)$ for which the profit function equals zero. For a standard call option with strike K and premium c, the BEP satisfies $P(S(T)) = 0$ and is therefore given by $S(T) = K + c$; for a standard put option with strike K and premium p, the BEP is $S(T) = K - p$. In other words, the BEP is the point at which the intrinsic value of the option exactly offsets the premium paid (by the buyer) or received (by the writer).

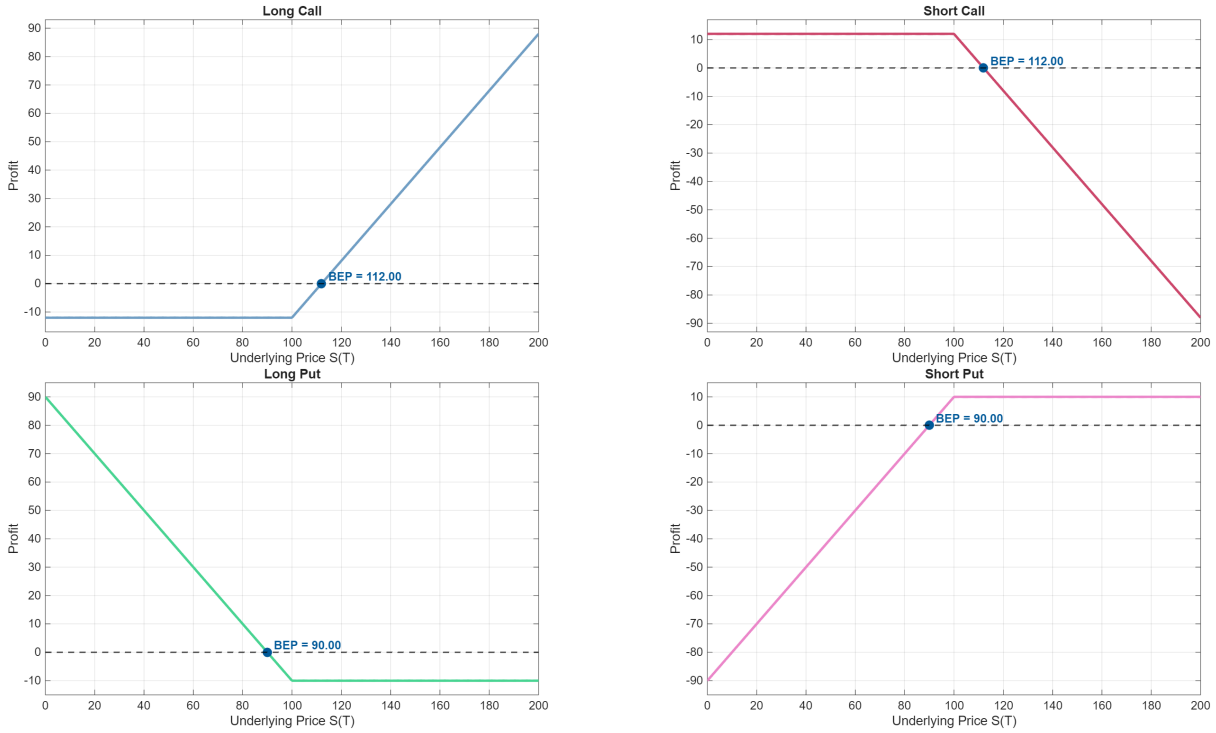


Figure 1: Profit diagrams for the four basic option positions: long/short call and long/short put. Consider, for example, K=100, c=12, p=10.

1.2 Upper and lower bound for option pricing

Before discussing in detail the differences between European and American options, it is necessary to introduce the **no-arbitrage principle** and derive its basic implications for the admissible range of option prices.

No Arbitrage principle: A financial market is said to be free of arbitrage if there exist no self-financing trading strategy that:

- starts from zero initial wealth,
- never produces a negative portfolio value at any time,
- has a strictly positive probability of yielding a strictly positive final payoff.

Any price that violated this principle would allow arbitrage opportunities, and therefore cannot represent a fair market price for a derivative security.

1.2.1 Put–Call Parity

An important relationship between European put and call options is the **Put–Call Parity**. Consider a portfolio constructed at time 0 by buying one European call option (long call) and simultaneously selling one European put option (short put), both written on the same underlying asset, with maturity T and strike price K. Let $S(T)$ denote the stock price at maturity, the total payoff ϕ of the portfolio at maturity can be obtained by summing the payoff of call and put options, respectively $\phi_c = (S(T) - K)^+$ and $\phi_p = (K - S(T))^+$.

We distinguish the following cases:

- **Case 1:** $S(T) \geq K$**.** The call finishes in the money and is exercised, generating a positive payoff $S(T) - K$, while the put expires worthless. In this case, the portfolio payoff equals $\phi = S(T) - K$.
- **Case 2:** $S(T) < K$**.** The call expires worthless and the put is exercised by the holder, so the writer must buy the underlying at K getting a loss. The portfolio payoff in this case is $\phi = -(K - S(T)) = S(T) - K$.

Hence, independently of the value of $S(T)$, the portfolio payoff coincides with that of a **long forward contract** with delivery price K and maturity T. By the no-arbitrage principle [Hull, 2021], two portfolios with identical payoffs at maturity must have the same value at time 0. Denoting by c and p the prices of the European call and put at time 0 and by $V(0)$ the value of the forward contract, we obtain:

$$c - p = V(0).$$

For a non-dividend-paying stock and a constant risk-free interest rate r, the value at time 0 of a forward contract with maturity T and strike K is given by:

$$V(0) = S(0) - Ke^{-rT},$$

which yields the standard form of the put–call parity:

$$c - p = S(0) - Ke^{-rT}.$$

More generally, if the underlying stock pays dividends between times 0 and T, the forward value must be adjusted by subtracting the present value of the dividend payments from the spot price. Let $D(0)$ denote the present value of all dividends paid over $(0, T]$, then the time-0 value of the forward is:

$$V(0) = S(0) - D(0) - Ke^{-rT},$$

and the put-call parity becomes:

$$c - p = S(0) - D(0) - Ke^{-rT}.$$

The observation above can be summarized in the following proposition.

Proposition 1 (Put–Call Parity)**.** *Let $S(0)$ denote the current price of a non-dividend-paying stock, with strike price K, maturity T and r the constant risk-free interest rate. Let c and p denote the prices at time 0 of a European call and a European put option, respectively, both written on the same underlying asset. Then the following relation holds:*

$$c - p = S(0) - Ke^{-rT}.$$

An alternative proof can be derived using the following arbitrage strategy:

Proof. Assume temporarily that the parity relation does not hold. To demonstrate this formula, let's construct an arbitrage strategy and suppose that: $c - p > S(0) - Ke^{-rT}$. In this case is convenient to buy put and to sell the call. A convenient arbitrage strategy is the following:

t=0	Cashflow	T	Cashflow
buy asset	$-S(0)$	If S(T)<K : exercise put, not exercise call	$+K$
buy put	$-p$	If S(T)>K : holder exercise call, not exercise put	$+K$
sell call	$+c$		
invest (> 0) or borrow (< 0)	$c - p - S(0)$	cash or repay	$(c - p - S(0))e^{rT}$

Total cashflow at T is: $K+(c-p-S(0))e^{rT} > 0$ under the assumption, this expression is strictly positive, violating the no-arbitrage principle.

Now suppose that: $c - p < S(0) - Ke^{-rT}$

t=0	Cashflow	T	Cashflow
short sell asset	$+S(0)$		
buy call	$-c$	If S(T)<K : not exercise call, holder exercise put	$-K$
sell put	$+p$	If S(T)>K : exercise call, not exercise put	$-K$
invest (> 0) or borrow (< 0)	$S(0) - c + p$	cash or repay	$(S(0) - c + p)e^{rT}$

Total cashflow at T is: $(S(0)-c+p)e^{rT}-K > 0$ under the assumption, this expression is strictly positive, again violating the no-arbitrage principle. □

1.2.2 Put–Call parity for American options

For American options, an exact put–call parity relation does not hold due to the possibility of early exercise. However, the no-arbitrage principle implies that the difference between the prices of an American call and an American put is bounded [Hull, 2021].

Let c^A and p^A denote the prices at time $t = 0$ of an American call and an American put option, respectively, written on a non-dividend-paying stock with spot price $S(0)$, strike price K, and maturity T. Then the following inequalities hold:

$$S(0) - K \;\leq\; c^A - p^A \;\leq\; S(0) - Ke^{-rT}.$$

The upper bound follows from the European put–call parity and the dominance relations between American and European options. Indeed, since an American put is always at

least as valuable as its European counterpart ($p^A \geq p$), and an American call written on a non-dividend-paying stock is never exercised early (so that $c^A = c$), one obtains

$$c^A - p^A \leq c - p = S(0) - Ke^{-rT}.$$

The lower bound reflects the possibility of early exercise of the American put option. Since the American put can be exercised immediately, its value is bounded below by its intrinsic value $K - S(0)$, which implies

$$p^A \geq K - S(0).$$

Combining this inequality with the trivial bound $c^A \geq 0$ yields

$$c^A - p^A \geq S(0) - K.$$

More generally, if the underlying asset pays dividends over the interval $(0, T]$, let $D(0)$ denote the present value of all dividends. In this case, the bounds become

$$S(0) - D(0) - K \;\leq\; c^A - p^A \;\leq\; S(0) - D(0) - Ke^{-rT}.$$

From the put–call parity relation, several important properties concerning the relationship between European and American option prices can be derived.

Property 1: The price of an American option is always greater than or equal to the price of the corresponding European option: $c \leq c^A$ and $p \leq p^A$. This follows from the fact that the American holder has a larger exercise opportunity, making the contract more valuable.

Theorem 2. *The price of an American call option written on a non-dividend-paying stock coincides with the price of an European call option with the same maturity T, strike price K and spot price S:*

$$c^A = c$$

Proof. Assume, by contradiction, that the price of the American call exceeds that of the European call, that is $c^A > c$. An arbitrage strategy can be constructed.

t=0	Cashflow	T	Cashflow
short American call	$+c^A$	Case 1	
long European call	$-c$	Case 2	
invest	$-(c^A - c)$		

We distinguish two cases:

Case 1: the American call is exercised early at some time $t < T$. If the holder exercises the American call at time t, the writer must deliver one share and receives the strike price K. To deliver the share without purchasing it the writer has to borrow one share, sell it to the option holder and receive K, which is invested from t to T.

The borrowed share must be repurchased at time T, at a cost of $S(t)e^{r(T-t)}$. The investment of K grows to $Ke^{r(T-t)}$. Meanwhile, the initial investment grows to $(c^A - c)e^{rT}$. At maturity T, exercise the European call to buy one share for K and use it to close the short position in the stock.

The total payoff at time T is therefore:

$$(c^A - c)e^{rT} + \big(K - S(t)\big)e^{r(T-t)}.$$

Since early exercise of the call option requires $S(t) > K$, the second term is negative but strictly dominated by the strictly positive first term. Thus

$$(c^A - c)e^{rT} + (K - S(t))e^{r(T-t)} > 0,$$

which yields a risk-free profit and so an arbitrage.

Case 2: the American call is never exercised before T. At maturity, the European call and the American call have identical payoffs, so their values cancel out. The investor keeps the accumulated value of the initial investment:

$$(c^A - c)e^{rT} > 0.$$

This is again an arbitrage.

Conclusion. Both cases lead to a strictly positive payoff with no initial cost, which constitutes an arbitrage. Since arbitrage is impossible in a no-arbitrage market, our assumption must be false. Hence,

$$c^A = c.$$

□

Property 2: The price of a European call option satisfies: $c \leq S(0)$ otherwise it is possible to find an arbitrage opportunity.

Property 3: For European options written on a non-dividend-paying stock, the following bounds hold:

$$c \geq S(0) - Ke^{-rT},$$
$$Ke^{-rT} - S(0) \leq p \leq Ke^{-rT}.$$

Property 4: The prices of European call and put options are monotone with respect to the strike price K:

$$K' < K \;\Rightarrow\; c(K') \geq c(K), \qquad p(K') \leq p(K).$$

2 Technical Background on Option Pricing

2.1 Market Assumptions

The following hypotheses follow the standard formulations adopted in Hull [2021], Capinski and Zastawniak [2011] and Glasserman [2004] within the standard set of assumptions commonly used in arbitrage pricing theory. In particular, assume that financial markets satisfy the following properties:

1. **Divisibility of assets.** All traded assets are perfectly divisible, meaning that fractional positions can be freely taken.

2. **No bid-ask spread.** The buying price equals the selling price for every asset. Consequently, investors can enter and exit positions at the same price without incurring losses due to price differentials.

3. **No transaction costs.** Trading is frictionless: there are no brokerage fees, taxes, or any fixed transaction costs associated with buying or selling financial assets.

4. **Perfect liquidity.** Markets are assumed to be perfectly liquid, implying that agents can buy or sell any quantity of the asset at any time without affecting its price. Trading opportunities are continuous and unrestricted.

5. **Absence of arbitrage.** The market is arbitrage-free i.e. there exist no self-financing trading strategies that generate a sure profit with zero initial investment and no risk. This assumption is fundamental for the consistency of pricing models and, in continuous-time settings, for the existence of an equivalent risk-neutral probability measure.

Option pricing consists in determining the fair value of an option contract by modeling the behavior of the underlying asset under standard market assumptions. The price of an option depends on a set of key market factors:

- the maturity of the contract T,
- the volatility of the underlying asset σ under the risk-neutral measure,
- the current market price S_0 of the underlying asset,
- the strike price K,
- the risk-free interest rate r,
- the dividends paid by the underlying asset, if any.

Table 1, adapted from Hull [2021], summarizes the effect of a marginal change in each of these variables on the prices of standard European and American options, holding all other factors constant.

Table 1: Effect of underlying variables on option prices

Variable	**EU call**	**EU put**	**American call**	**American put**
Current stock price	+	−	+	−
Strike price	−	+	−	+
Time to expiration	?	?	+	+
Volatility	+	+	+	+
Risk-free rate	+	−	+	−
Amount of future dividends	−	+	−	+

+ indicates that an increase in the variable causes the option price to increase or remain unchanged;

− indicates that an increase in the variable causes the option price to decrease or remain unchanged;
? indicates that the relationship is ambiguous.

Under standard market assumptions, an American option with a longer time to expiration cannot be worth less than an otherwise identical American option with a shorter maturity. The possibility of early exercise implies that a longer maturity expands the decision set available to the holder and consequently the option's value increases.

For European options, which can only be exercised at maturity, the argument is different. In general, a longer time to expiration increases the option value since additional time introduces greater uncertainty regarding the future price of the underlying asset. Higher uncertainty raises the probability that the option finishes ITM, thereby increasing its expected payoff. As a result, European option prices are typically increasing in maturity.

However, once dividends or other cash flows on the underlying asset are introduced, the monotonicity of European option prices with respect to maturity may fail.
Consider, for instance, two European call options written on the same stock, with maturities $T_1 = 1$ month and $T_2 = 2$ months, respectively. Suppose that a large, known dividend is expected to be paid in 6 weeks. On the dividend payment date, the stock price is expected to drop by approximately the amount of the dividend, reflecting the fact that new buyers of the stock are no longer entitled to receive that payment. The call with maturity T_1 expires before the dividend date and is therefore unaffected by this anticipated price drop, while the call with maturity T_2 is exposed to the downward jump in the underlying price. If the dividend is sufficiently large relative to volatility and interest rates, the expected terminal payoff of the longer-maturity call can be lower than that of the shorter-maturity call, so that the longer-maturity call may be strictly less valuable today than the otherwise identical shorter-maturity call.

A related type of non-monotonicity may arise for European put options. Although a European put cannot be exercised early, its value is always bounded above by the value of the corresponding American put written on the same underlying with the same strike and maturity, i.e. $Ke^{-rT} - S_0 \leq p \leq Ke^{-rT}$.

For a deep ITM American put $S_0 \ll K$ with short maturity in an environment with high interest rates and negligible dividends, early exercise can become optimal: by exercising early, the holder receives the strike price K immediately and can invest it at

higher risk-free rate r, rather than waiting until maturity. This possibility generates an early-exercise premium that pushes the American put price upward near maturity, where early exercise is more likely to be optimal.

Since the European put price cannot exceed the value of the corresponding American put, a short-maturity European put may also be relatively expensive in situations where early exercise of the American put is likely. In contrast, for longer maturities early exercise is typically suboptimal and the early-exercise premium becomes negligible. Under suitable combinations of high interest rates, deep in-the-money conditions, and limited dividend payments, it is therefore possible for a European put with shorter maturity to be more valuable than an otherwise identical European put with longer maturity.

2.2 Wiener Process and Stochastic Calculus

In finance, the price of the underlying asset is modeled as a stochastic process whose future path is uncertain and evolves either in discrete or continuous time. Discrete-time models, such as binomial trees, assume that the underlying price changes only at a finite set of predetermined dates and remains constant between successive time steps. In contrast, in continuous-time models the price evolves continuously providing a more realistic characterization of market dynamics.

The fundamental driving process in many continuous-time models is the **Wiener process** also known as **Standard Brownian Motion** which is a continuous-time stochastic process with the Markov property that describes the behaviour of a random variable over time.
Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $\{\mathcal{F}_t\}_{t\geq 0}$ denote the natural filtration generated by the process W, i.e. the $\sigma-$algebra [1] generated by the process:

$$\mathcal{F}_t = \sigma\{W_s : 0 \leq s \leq t\}.$$

For financial application, $\mathcal{F}_t$ represents the information available to market participants up to time t.
A stochastic process $W = \{W_t\}_{t\geq 0}$ defined on this space is called a *Wiener process* if the following properties hold:

1. $W_0 = 0$ almost surely.
2. W_t has almost surely continuous sample paths.
3. For every choice $0 \leq t_0 < t_1 < \cdots < t_n$, the increments $W_{t_1} - W_{t_0}$, $W_{t_2} - W_{t_1}$, $\ldots$, $W_{t_n} - W_{t_{n-1}}$ are independent.
4. The increments are stationary and Gaussian: for $0 \leq s < t$, the increment $W_t - W_s$ is normally distributed with mean 0 and variance $t - s$, i.e.

$$W_t - W_s \sim \mathcal{N}(0, t - s).$$

[1] Following Billingsley [1995], a σ-algebra (or σ-field) on a set Ω is a collection $\mathcal{F}$ of subsets of Ω satisfying: (i) $\Omega \in \mathcal{F}$; (ii) if $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$ (closed under complementation); and (iii) if $A_n \in \mathcal{F}$ for $n = 1, 2, \ldots$, then $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$ (closed under countable unions).

Equivalently, since the Wiener process is a Gaussian process, it is completely characterized by its mean and covariance functions.

Proposition 3. *Let $W = \{W(t)\}_{t\geq 0}$ be a Wiener process. Then, the following relations hold:*

- $\mu_t = \mathbb{E}[W(t)] = 0$,
- $\mathrm{Var}(W(t)) = t$,
- $\mathrm{Cov}(W(s), W(t)) = \min\{s, t\}, \qquad 0 \leq s \leq t.$

Proof.

$$\begin{aligned}
\mathrm{Cov}(W(s), W(t)) &= \mathbb{E}[W(t)W(s)] - \mathbb{E}[W(t)]\mathbb{E}[W(s)] \\
&= \mathbb{E}[W(t)W(s)] \\
&= \mathbb{E}[W(s)[W(t) - W(s) + W(s)]] \\
&= \mathbb{E}[W(s)[W(t) - W(s)]] + \mathbb{E}[W^2(s)] \quad \text{(independent increments)} \\
&= \mathbb{E}[W(s)]\mathbb{E}[[W(t) - W(s)]] + \mathbb{E}[W^2(s)] \\
&= 0 + \mathrm{Var}(W(s)) \\
&= s = \min(s, t)
\end{aligned}$$

□

The following properties describe the probabilistic structure and temporal evolution of the Wiener process W, highlighting how future increments are related to past information.

Property 1. (Distribution of small increments) For any t and sufficiently small $\Delta t > 0$,

$$W_{t+\Delta t} - W_t \sim \mathcal{N}(0, \Delta t).$$

which implies that random fluctuations scale with the square root of time.
For simulation purposes, a useful representation is:

$$W_{t+\Delta t} - W_t \overset{d}{=} \sqrt{\Delta t}\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 1).$$

where the symbol $\overset{d}{=}$ denotes equality in distribution sense.

Property 2. (Markov property) The Wiener process satisfies the Markov property: the distribution of future states depends only on the current state and not on the trajectory. Formally, all $0 \leq s \leq t$, the conditional distribution of $W(t)$ given the natural filtration $\mathcal{F}_s = \sigma\{W(u) : 0 \leq u \leq s\}$ coincides with the conditional distribution given only $W(s)$:

$$\mathbb{P}(W(t) \leq x \mid \mathcal{F}_s) = \mathbb{P}(W(t) \leq x \mid W(s))$$

Property 3 (Martingale property). The Wiener process belongs to the class of martingales, reflecting the fact that its conditional expectation remains constant at the last observed value. This property is mathematically expressed as follows:

Lemma 4. *Let $W = \{W(t)\}_{t\geq 0}$ be a Wiener process with natural filtration $\{\mathcal{F}_t\}_{t\geq 0}$. Then, for all $0 \leq s \leq t$,*

$$\mathbb{E}[W(t) \mid \mathcal{F}_s] = W(s).$$

Proof. For $0 \leq s \leq t$, we decompose $W(t)$ as

$$W(t) = W(s) + \big(W(t) - W(s)\big).$$

Taking conditional expectation with respect to $\mathcal{F}_s$ and using linearity, we obtain

$$\mathbb{E}[W(t)|\mathcal{F}_s] = \mathbb{E}[W(s) + (W(t) - W(s))|\mathcal{F}_s]$$

Since $W(s)$ is $\mathcal{F}_s$-measurable, it follows that

$$\mathbb{E}[W(s) \mid \mathcal{F}_s] = W(s).$$

Moreover, by the independent increments property of the Wiener process, the increment $W(t) - W(s)$ is independent of $\mathcal{F}_s$. Therefore,

$$\mathbb{E}[W(t) - W(s) \mid \mathcal{F}_s] = 0,$$

because $W(t) - W(s) \sim \mathcal{N}(0, t - s)$.

Combining the above results yields

$$\mathbb{E}[W(t) \mid \mathcal{F}_s] = W(s),$$

which proves that $W(t)$ is a martingale. □

Property 4 (Quadratic Variation). In classical calculus, the square of a differential $(dt)^2$ is of higher order than dt and is therefore neglected. The Wiener process, however, is characterized by a fundamentally higher degree of variability which requires a different treatment [Shreve, 2004b].
Recalling Property 1, for any t and sufficiently small $\Delta t > 0$, the increment

$$\Delta W_t = W_{t+\Delta t} - W_t \sim \mathcal{N}(0, \Delta t).$$

The expected value of its square follows from

$$\mathbb{E}\big[(\Delta W_t)^2\big] = \mathrm{Var}(\Delta W_t) + \big(\mathbb{E}[\Delta W_t]\big)^2 = \Delta t + 0 = \Delta t.$$

To assess how closely $(\Delta W_t)^2$ concentrates around this mean, compute its variance as:

$$\mathrm{Var}\big[(\Delta W_t)^2\big] = \mathbb{E}\big[(\Delta W_t)^4\big] - \big(\mathbb{E}\big[(\Delta W_t)^2\big]\big)^2.$$

Since $\Delta W_t \sim \mathcal{N}(0, \Delta t)$, one can write $\Delta W_t \overset{d}{=} \sqrt{\Delta t}\,\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$, so that

$$\mathbb{E}\big[(\Delta W_t)^4\big] = (\Delta t)^2\,\mathbb{E}[\varepsilon^4] = 3(\Delta t)^2,$$

where $\mathbb{E}[\varepsilon^4] = 3$ is the fourth moment of a standard normal random variable. Substituting yields:

$$\mathrm{Var}\big[(\Delta W_t)^2\big] = 3(\Delta t)^2 - (\Delta t)^2 = 2(\Delta t)^2.$$

As $\Delta t \to 0$, the variance of $(\Delta W_t)^2$ vanishes faster than its mean Δt, so $(\Delta W_t)^2$ concentrates around Δt with negligible fluctuations and behaves as a deterministic quantity. This leads to the fundamental identity of Itô calculus:

$$(dW_t)^2 = dt. \tag{1}$$

Together with the fact that $dW_t = \mathcal{O}(\sqrt{dt})$, this identity implies the following **Itô multiplication rules** Hull [2021], Shreve [2004b]:

$$(dW_t)^2 = dt, \qquad dW_t\, dt = \mathcal{O}(dt^{3/2}) = 0, \qquad (dt)^2 = \mathcal{O}(dt^2) = 0,$$

where the last two terms are of higher order in dt and therefore vanish as $dt \to 0$. In contrast, $(dW_t)^2$ retains order dt and contributes explicitly in stochastic differential calculus.

2.3 Drifted Brownian Motion and the Generalized Wiener Process

A natural extension of the standard Wiener process is the **Drifted Brownian Motion** (DBM), which is defined as

$$B_t = B_0 + \mu t + \sigma W_t,$$

where:

- B_0 is the initial value at time zero,
- $\mu \in \mathbb{R}$ denotes the constant drift parameter, representing the average rate of change per unit time,
- $\sigma > 0$ is the volatility coefficient scaling the random component,
- W_t is a standard Wiener process as defined previously.

The introduction of the drift term μt allows the expected value of the process to evolve deterministically over time, while the stochastic component σW_t captures random fluctuations around this trend.
In particular, the first two moments of B_t are given by

$$\mathbb{E}[B_t] = B_0 + \mu t, \qquad \mathrm{Var}(B_t) = \sigma^2 t,$$

so that B_t is normally distributed as

$$B_t \sim \mathcal{N}(B_0 + \mu t,\ \sigma^2 t).$$

In differential form, the drifted Brownian motion can be expressed as the solution of the Stochastic Differential Equation (SDE),

$$dB_t = \mu\, dt + \sigma\, dW_t.$$

This representation highlights the decomposition of the process into a deterministic component, $\mu\,dt$ and a stochastic component, $\sigma\,dW_t$.

More generally, a **generalized Wiener process** is defined by the SDE

$$dx_t = a\,dt + b\,dW_t,$$

where a and b are constant real coefficients representing, respectively, the drift and the volatility of the process. By identifying $a = \mu$ and $b = \sigma$, the Drifted Brownian Motion B_t is seen as a particular case of a generalized Wiener process.

Generalized Wiener processes play a fundamental role in financial modeling. In the Black and Scholes framework, they are employed to describe asset price dynamics, under the real-world probability measure, where the drift reflects the expected rate of return and the volatility quantifies uncertainty and risk. The Stochastic Differential Equation of the DBM:

$$dB_t = \mu\,dt + \sigma\,dW_t$$

can be written in integral form as:

$$B(t) = B(0) + \int_0^t \mu ds + \int_0^t \sigma dW(s)$$

While the first integral is a standard Riemann integral, the second term is a stochastic integral with respect to Brownian motion. Its rigorous definition requires the tools of stochastic calculus and motivates the introduction of Itô processes and Itô integration.

2.4 Itô processes and Itô's lemma

In many applications the drift and volatility of a stochastic process are not constant, but may depend both on time and on the current level of the process itself (e.g. the price of the underlying asset). This motivates the introduction of **Itô processes**, which generalize the constant–coefficient generalized Wiener processes previously described.

Itô process. Let $(\Omega, \{\mathcal{F}_s\}_{s\geq 0}, \mathbb{P})$ be a filtered probability space supporting a standard Brownian motion $W = \{W_t\}_{t\geq 0}$. A real-valued adapted process $X = \{X_t\}_{t\geq 0}$ is called an *Itô process* if it admits a decomposition of the form:

$$dX_t = a(X_t, t)\,dt + b(X_t, t)\,dW_t,$$

where

- $a : \mathbb{R} \times [0, \infty) \to \mathbb{R}$ is the *drift*
- $b : \mathbb{R} \times [0, \infty) \to \mathbb{R}$ is the *diffusion* or *volatility coefficient*.

The stochastic differential equation above is understood in the integral sense, i.e.

$$X_t = X_0 + \int_0^t a(X_s, s)\,ds + \int_0^t b(X_s, s)\,dW_s.$$

The assumption that X_t is adapted ensures that its value at time t depends only on information available up to that time and guarantees that the stochastic integral with respect to W_t is well defined.
Moreover, it is typically assumed that a and b satisfy suitable Lipschitz continuity and linear growth conditions, which ensure existence and uniqueness of a strong solution to the SDE (see, e.g. Øksendal [2003] and Karatzas and Shreve [1991]).

For a small time increment $\Delta t > 0$, the increment of an Itô process satisfies in distribution

$$X_{t+\Delta t} - X_t \approx a(X_t, t)\,\Delta t + b(X_t, t)\,\sqrt{\Delta t}\,\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0,1),$$

so that, conditionally on $\mathcal{F}_t$, this increment is approximately Gaussian with conditional mean $a(X_t, t)\Delta t$ and conditional variance $b^2(X_t, t)\Delta t$.
This representation highlights the separation between the predictable component (drift) and the random component (diffusion) in the dynamics of X_t.

Itô's Lemma. A fundamental tool for analyzing functions of Itô processes is Itô's Lemma, which plays a role analogous to the chain rule in classical calculus.
Let $X = \{X_t\}_{t\geq 0}$ be an Itô process satisfying the Stochastic Differential Equation

$$dX_t = a(X_t, t)\,dt + b(X_t, t)\,dW_t,$$

and let $G : \mathbb{R} \times [0, \infty) \to \mathbb{R}$ be a function that is twice continuously differentiable with respect to X and once with respect to t.
Then, Itô's Lemma states that $G(X_t, t)$ solves the following SDE:

$$dG(X_t, t) = \left(\frac{\partial G}{\partial t} + a(X_t, t)\frac{\partial G}{\partial X} + \frac{1}{2}b^2(X_t, t)\frac{\partial^2 G}{\partial X^2}\right) dt + b(X_t, t)\frac{\partial G}{\partial X}\,dW_t.$$

The drift rate (i.e. the conditional expected rate of change) of the process $G(X_t, t)$ is therefore:

$$\mu_G(X_t, t) = \frac{\partial G}{\partial t} + a(X_t, t)\frac{\partial G}{\partial X} + \frac{1}{2}b^2(X_t, t)\frac{\partial^2 G}{\partial X^2},$$

while the instantaneous variance rate of $G(X_t, t)$ is given by the square of the diffusion coefficient:

$$\sigma_G^2(X_t, t) = b^2(X_t, t)\left(\frac{\partial G}{\partial X}\right)^2.$$

2.5 Geometric Brownian Motion

An important application of Itô's Lemma in financial modeling is the **Geometric Brownian Motion** (GBM), which is the standard continuous-time model for the dynamics of a stock price S_t. The choice of this model is motivated by both empirical considerations and mathematical tractability, and its explicit solution implies that stock prices remain strictly positive almost surely.

A stochastic process $\{S_t\}_{t\geq 0}$ is said to follow a Geometric Brownian Motion if it satisfies the Itô stochastic differential equation

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

where $\mu \in \mathbb{R}$ is the constant drift parameter, $\sigma > 0$ is the volatility, and W_t is a standard Wiener process.

2.5.1 Derivation of the lognormal distribution

To analyze the distributional properties of S_t, apply Itô's Lemma to the logarithmic transformation $G(S_t, t) = \ln S_t$. The function $G(x, t) = \ln x$ belongs to the class $\mathcal{C}^{2,1}(\mathbb{R}_+ \times [0, T])$, that means it is twice continuously differentiable with respect to the spatial variable x and once continuously differentiable with respect to time t. In this case, G does not depend explicitly on time, therefore

$$\frac{\partial G}{\partial x} = \frac{1}{x}, \qquad \frac{\partial^2 G}{\partial x^2} = -\frac{1}{x^2}, \qquad \frac{\partial G}{\partial t} = 0.$$

Since S_t follows the Itô process

$$dS_t = a(S_t, t)\, dt + b(S_t, t)\, dW_t, \qquad a(S_t, t) = \mu S_t, \quad b(S_t, t) = \sigma S_t,$$

Applying Itô's Lemma to the logarithmic transformation $G(S_t, t) = \ln S_t$, one obtains:

$$d(\ln S_t) = \left(\frac{\partial G}{\partial t} + a(S_t, t)\frac{\partial G}{\partial S} + \frac{1}{2} b^2(S_t, t)\frac{\partial^2 G}{\partial S^2} \right) dt + b(S_t, t)\frac{\partial G}{\partial S}\, dW_t.$$

Substituting the derivatives and coefficients yields

$$d(\ln S_t) = \left(\mu - \frac{1}{2}\sigma^2 \right) dt + \sigma\, dW_t.$$

Integrating this SDE over a finite time interval from 0 to t

$$\ln S_t - \ln S_0 = \int_0^t \left(\mu - \frac{1}{2}\sigma^2 \right) du + \sigma \int_0^t dW_u,$$

$$\ln S_t = \ln S_0 + \left(\mu - \frac{1}{2}\sigma^2 \right) t + \sigma W_t. \tag{2}$$

Since $W_t \sim \mathcal{N}(0, t)$, it follows that $\sigma W_t \sim \mathcal{N}(0, \sigma^2 t)$, and thus

$$\ln S_t \sim \mathcal{N}\left(\ln S_0 + \left(\mu - \frac{1}{2}\sigma^2 \right) t,\ \sigma^2 t \right),$$

from which one derives the explicit solution

$$S_t = S_0 \exp\left[\left(\mu - \frac{1}{2}\sigma^2 \right) t + \sigma W_t \right] \tag{3}$$

and since $\ln S_t$ is normal, S_t has a **lognormal** distribution:

$$S_t \sim \mathrm{LN}\left(\ln S_0 + \left(\mu - \frac{1}{2}\sigma^2 \right) t,\ \sigma^2 t \right).$$

Logarithmic returns. For $0 \le u < v$,

$$\ln \frac{S_v}{S_u} = \ln S_v - \ln S_u = \left(\mu - \frac{1}{2}\sigma^2 \right)(v - u) + \sigma(W_v - W_u),$$

with $W_v - W_u \sim \mathcal{N}(0, v - u)$ independent of the past.
Thus, logarithmic returns are normally distributed:

$$\ln \frac{S_{t+\Delta t}}{S_t} \sim \mathcal{N}\left(\left(\mu - \frac{1}{2}\sigma^2 \right) \Delta t,\ \sigma^2 \Delta t \right).$$

Implications for option pricing. The result derived above shows that, under the Geometric Brownian Motion assumption, asset prices are **lognormally distributed**, while logarithmic returns are **normally distributed**. This structural property is fundamental in the Black–Scholes framework. In an arbitrage-free market, pricing is performed under the risk-neutral probability measure $\mathbb{Q}$, under which the drift of the asset price equals the risk-free interest rate r. Replacing μ with r in (2), the terminal price S_T admits a lognormal distribution conditional on the information available at time 0. The explicit knowledge of the conditional distribution of S_T given $\mathcal{F}_0$ (which is the information available at initial time) allows the expectation of discounted payoffs to be computed in closed form, leading to the Black–Scholes pricing formulas for European call and put options.

2.6 The Black-Scholes model

The Black-Scholes-Merton framework (see Black and Scholes [1973], Merton [1973]) constitutes the foundational continuous-time model for the valuation of derivative securities and, in particular, of European options.
The key assumption is that the price of the underlying asset evolves according to a Geometric Brownian Motion with constant volatility and that financial markets are frictionless, arbitrage-free, and permit continuous trading. Under these assumptions, it is possible to derive a closed–form pricing formula for European call and put options.

2.6.1 Option pricing and delta hedging

Let S_t denote the price of a non-dividend-paying stock at time t. Under the real-world probability measure, the stock price is assumed to follow a Geometric Brownian Motion:

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

where $\mu \in \mathbb{R}$ represents the instantaneous expected return, $\sigma > 0$ is the constant volatility, and W_t is a standard Brownian motion.

Let $C(S_t, t)$ denote the value at time t of a European call option written on the stock, with maturity T and strike price K. Since the option value depends on the current stock price and on time, it can be regarded as a sufficiently smooth function $C : \mathbb{R}_+ \times [0, T] \to \mathbb{R}$.
Applying Itô's Lemma to the process $C(S_t, t)$ yields to the SDE:

$$dC = \frac{\partial C}{\partial t}\, dt + \frac{\partial C}{\partial S}\, dS_t + \frac{1}{2}\frac{\partial^2 C}{\partial S^2}\, dS_t^2.$$

To evaluate dS_t^2, expand the square of the stock price differential:

$$dS_t^2 = \big(\mu S_t\, dt + \sigma S_t\, dW_t\big)^2 = \mu^2 S_t^2\, (dt)^2 + 2\mu\sigma S_t^2\, dt\, dW_t + \sigma^2 S_t^2\, (dW_t)^2.$$

In the calculus of Itô processes, products of differentials are ordered by their degree of infinitesimality: $(dt)^2$ and $dt\, dW_t$ are of higher order than dt and therefore vanish in the limit (Hull [2021]), while $(dW_t)^2 = dt$ by the quadratic variation property defined in Section 2.2. Applying these multiplication rules yields

$$dS_t^2 = \sigma^2 S_t^2\, dt.$$

Substituting of the stock dynamics gives:

$$dC = \left(\frac{\partial C}{\partial t} + \mu S_t \frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 C}{\partial S^2}\right) dt + \sigma S_t \frac{\partial C}{\partial S}\, dW_t.$$

Construction of a riskless portfolio with delta hedging. To eliminate uncertainty, consider a self-financing portfolio Π_t consisting of a long position in the option and a short position of Δ_t units of the stock:

$$\Pi_t = C(S_t, t) - \Delta_t S_t.$$

Since the portfolio is self-financing, its differential satisfies

$$d\Pi_t = dC - \Delta_t \, dS_t.$$

Substituting the dynamics of $C(S_t, t)$ obtained from Itô's Lemma and the stock price dynamics, we obtain

$$d\Pi_t = \left(\frac{\partial C}{\partial t} + \mu S_t \frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 C}{\partial S^2} - \Delta_t \mu S_t \right) dt + \sigma S_t \left(\frac{\partial C}{\partial S} - \Delta_t \right) dW_t.$$

Choosing the hedging strategy

$$\Delta_t = \frac{\partial C}{\partial S},$$

eliminates the stochastic term proportional to dW_t. The portfolio is therefore locally risk-free, and its dynamics reduces to

$$d\Pi_t = \left(\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 C}{\partial S^2} \right) dt.$$

By the no-arbitrage principle, a locally risk-free portfolio must earn the risk-free interest rate r, implying

$$d\Pi_t = r\Pi_t \, dt.$$

Substituting $\Pi_t = C - S_t \frac{\partial C}{\partial S}$ and rearranging terms yields the Black–Scholes partial differential equation:

$$\frac{\partial C}{\partial t} + r S_t \frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S_t^2 \frac{\partial^2 C}{\partial S^2} = rC.$$

Boundary condition and closed-form solution. For a European call option, the value at maturity is determined by its payoff function, which provides the terminal condition for the Black–Scholes partial differential equation:

$$C(S, T) = (S_T - K)^+ = \max(S_T - K, 0).$$

Solving the Black–Scholes PDE subject to this terminal condition yields the well-known Black–Scholes pricing formula, see Black and Scholes [1973] and Hull [2021]:

$$C(S, t) = S(t) \, N(d_1) - K e^{-r(T-t)} N(d_2), \tag{4}$$

where

$$d_1 = \frac{\ln(S_0/K) + \left(r + \frac{1}{2}\sigma^2\right)(T - t)}{\sigma\sqrt{T - t}}, \qquad d_2 = d_1 - \sigma\sqrt{T - t}.$$

Here, $N(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

Using the put–call parity relationship, the price of a European put option with the same strike price K and maturity T is given by

$$P(S, t) = K e^{-r(T-t)} N(-d_2) - S(t) \, N(-d_1).$$

Conclusion. As shown in the previous chapter, for non-dividend–paying stocks it is never optimal to exercise an American call option before maturity. As a consequence, the price of an American call option coincides with that of the corresponding European call, that is:

$$C^A(S,t) = C^E(S,t),$$

In contrast, American put options generally admit early exercise. Indeed, the holder of an American put may benefit from exercising prior to maturity in order to receive the strike price earlier and invest it at the risk-free rate. As a result, the American put price satisfies:

$$P^A(S,t) \geq P^E(S,t),$$

with strict inequality in regions where early exercise is optimal.

The pricing of American options leads to a *free-boundary problem*: the Black-Scholes PDE holds in the continuation region, while the value function must satisfy:

$$P^A(S,t) \geq (K-S)^+,$$

with equality along the optimal exercise boundary.
Unlike European options, no closed-form analytical solution exists for American put options. Consequently, their valuation requires the use of numerical methods, such as binomial tree models [Cox et al., 1979], finite-difference schemes for partial differential equations [Brennan and Schwartz, 1977], or simulation-based approaches [Glasserman, 2004].
In particular, Monte Carlo methods are especially attractive in high-dimensional settings. Among them, the Least Squares Monte Carlo (LSMC) method introduced by Longstaff and Schwartz [2001] provides a flexible and efficient framework for approximating the optimal exercise strategy and pricing American options.
The following chapter focuses on Monte Carlo simulation techniques and presents the implementation of the Longstaff-Schwartz's algorithm for the pricing of American put options.

3 Monte Carlo Simulation and focus on American Options

3.1 Principles of Monte Carlo

Monte Carlo methods are numerical techniques based on random sampling, widely used to approximate expectations that are difficult or impossible to compute analytically. Their theoretical foundation lies in measure theory, where probabilities are interpreted as normalized volumes of sets. From this perspective, Monte Carlo simulation inverts the usual viewpoint: instead of computing probabilities from known volumes, it estimates volumes by interpreting them as probabilities obtained through random experiments, see Glasserman [2004].

In its simplest form, the method aims to estimate the volume of a subset of a given domain. By drawing independent samples uniformly over the domain and computing the proportion of points that fall inside the target region, one obtains an unbiased estimator of the relative volume. The **law of large numbers** ensures that this empirical proportion converges to the true value as the number of samples increases. Moreover, the **central limit theorem** (CLT) characterizes the statistical error, showing that for a sufficiently large sample size the estimation error is approximately normally distributed, with a variance that decreases proportionally to the inverse of the number of samples.

This idea naturally extends to the numerical evaluation of integrals. Consider the problem of computing

$$\alpha = \int_0^1 f(x)\,dx.$$

If U is a random variable uniformly distributed on $[0,1]$, the integral can be written as the expectation $\alpha = \mathbb{E}[f(U)]$. Given n independent realizations $U_1, \ldots, U_n$ of U, the corresponding **Monte Carlo estimator** is defined as:

$$\hat{\alpha}_n = \frac{1}{n}\sum_{i=1}^{n} f(U_i).$$

By the strong law of large numbers[2], $\hat{\alpha}_n$ converges almost surely to α as $n \to \infty$. If the function f is square-integrable, the CLT further implies that the error $\hat{\alpha}_n - \alpha$ is asymptotically normal with mean zero and standard deviation $\frac{\sigma_f}{\sqrt{n}}$, where:

$$\sigma_f^2 = \int_0^1 (f(x) - \alpha)^2\,dx.$$

In practice, since σ_f is unknown, it is typically estimated using the sample variance of the observed function values.

[2]Let $(X_i)_{i\geq 1}$ be a sequence of independent and identically distributed random variables with $\mathbb{E}[|X_1|] < \infty$. The Strong Law of Large Numbers states that

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{\text{a.s.}} \mathbb{E}[X_1], \qquad \text{as } n \to \infty.$$

In the Monte Carlo setting, this result applies with $X_i = f(U_i)$, where U_i are i.i.d. uniformly distributed on $[0,1]$. A rigorous proof can be found in Durrett [2019].

A distinctive feature of Monte Carlo methods is their convergence rate. The standard error[3] decreases at rate $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$, implying that achieving a significant improvement in accuracy requires a substantial increase in the number of samples. While this convergence is slower than that of classical deterministic quadrature schemes — such as the one-dimensional trapezoidal rule, which attains an error of order $\mathcal{O}\left(\frac{1}{n^2}\right)$ for sufficiently smooth functions, Monte Carlo methods have the crucial advantage that their convergence rate is **independent of the dimensionality** of the problem (for more details, see Glasserman [2004]).

Indeed, when integrating functions over high-dimensional domains such as $[0,1]^d$, the performance of deterministic methods deteriorates rapidly as d increases. For example, multidimensional trapezoidal rule typically exhibit convergence rates of order $\mathcal{O}\left(n^{-2/d}\right)$, making them ineffective in high dimensions. In contrast, Monte Carlo estimators retain their $\mathcal{O}\left(n^{-1/2}\right)$ convergence rate regardless of the dimension, thereby avoiding the exponential complexity associated with the so-called *curse of dimensionality*.

This property is of fundamental importance in financial engineering. Many problems in derivative pricing can be formulated as the computation of expectations under a risk-neutral probability measure. When continuous-time stochastic processes are involved, these expectations often correspond to integrals over very high-dimensional, or even infinite-dimensional, spaces. Monte Carlo methods provide a flexible and robust framework for approximating such quantities.

In practice, Monte Carlo valuation of derivative securities relies on the simulation of sample paths of the stochastic processes describing the evolution of underlying asset prices, interest rates, or other model variables. Rather than sampling points in a finite-dimensional space, the method effectively samples from a space of paths. The resulting dimension is at least equal to the number of time steps used in the discretization and can therefore be very large. In such settings, the dimension-independent convergence of Monte Carlo methods makes them particularly competitive compared to alternative numerical approaches.

3.2 Monte Carlo example: European call pricing

To introduce the fundamental principles of Monte Carlo methods, we consider a simple yet representative example adapted from Glasserman [2004].
The problem consists in estimating the expected present value of the payoff of a European call option on a non-dividend paying stock. Although this quantity admits a closed-form solution under the Black-Scholes model (Section 2.6), the example demonstrates the key steps of Monte Carlo estimation: random input generation, path construction, payoff computation, and statistical analysis.

[3]The standard error $SE = \sqrt{\mathrm{Var}(\hat{\alpha}_n)}$ refers to the standard deviation of the Monte Carlo estimator $\hat{\alpha}_n$ which measures the statistical uncertainty due to random sampling and quantifies the fluctuations of the estimator around the true value α. As a consequence of the Central Limit Theorem, the standard error decreases proportionally to $1/\sqrt{n}$.

3.2.1 Problem formulation

Consider a European call option with maturity T and strike price K. Let $S(t)$ denote the price of the underlying asset at time t, with current time $t = 0$ and $S_T = S(T)$ the price at maturity. At maturity, the option payoff is given by

$$\phi_C = (S_T - K)^+ = \max\{0, S_T - K\}.$$

Discounting this payoff at a continuously compounded risk-free rate r, the expected present value can be defined

$$C = \mathbb{E}^{\mathbb{Q}}\left[e^{-rT}(S_T - K)^+\right] \tag{5}$$

where the expectation is taken under the risk-neutral probability measure $\mathbb{Q}$. Unless otherwise specified, all expectations appearing in the pricing formulas throughout this work are understood to be taken under $\mathbb{Q}$.
In order to evaluate this expectation, it is necessary to specify a probabilistic model for the evolution of the stock price. Within the Black–Scholes framework, the asset price dynamics under the risk-neutral measure are described by the Stochastic Differential Equation

$$\frac{dS(t)}{S(t)} = r\,dt + \sigma\,dW(t), \tag{6}$$

where $\sigma > 0$ denotes the volatility and $W(t)$ is a standard Brownian motion. The choice of drift equal to the interest rate reflects the risk-neutral valuation principle.
The solution of (6) gives the terminal stock price:

$$S_T = S_0 \exp\left[\left(r - \frac{1}{2}\sigma^2\right)T + \sigma W(T)\right]. \tag{7}$$

Since $W(T) \sim \mathcal{N}(0, T)$, it can be represented as $\sqrt{T}Z$, where Z is a standard normal random variable. Consequently, the terminal stock price admits the representation:

$$S_T = S_0 \exp\left[\left(r - \frac{1}{2}\sigma^2\right)T + \sigma\sqrt{T}\,Z\right], \tag{8}$$

which shows that S_T follows a lognormal distribution.

Instead of using the Black–Scholes formula, the expectation $\mathbb{E}[e^{-rT}(S_T - K)^+]$ can be also estimated via Monte Carlo simulation. Given a sequence of independent standard normal random variables $Z_1, Z_2, \ldots, Z_n$, the Monte Carlo algorithm proceeds as follows:

Algorithm 1 Monte Carlo estimation of $C = \mathbb{E}\left[e^{-rT}(S_T - K)^+\right]$

Require: Number of simulations $n \in \mathbb{N}$, model parameters S_0, r, σ, T, K.
1: **for** $i = 1, \ldots, n$ **do**
2: Draw $Z_i \sim \mathcal{N}(0, 1)$;
3: Simulate $S_T^{(i)}$ via (8);
4: Compute the discounted payoff $C_i = e^{-rT}\left(S_T^{(i)} - K\right)^+$;
5: **end for**
6: Compute the sample mean $\widehat{C}_n = \frac{1}{n}\sum_{i=1}^{n} C_i$;
7: Compute the sample standard deviation via (9)
Ensure: Return $\widehat{C}_n$ and the $1 - \delta$ confidence interval.

For any $n \geq 1$ the MC estimator $\widehat{C}_n$ is unbiased, that is, $E[\widehat{C}_n] = C$ and consequently $\widehat{C}_n \to C$ as $n \to \infty$ with probability 1.

For a finite number of simulations, the statistical uncertainty of the estimator can be quantified by means of confidence intervals. Let:

$$s_C = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(C_i - \widehat{C}_n)^2} \tag{9}$$

denotes the **sample standard deviation** of the simulated payoffs.

Let $z_{\delta/2}$ denote the $(1-\delta/2)$ quantile of the standard normal distribution, i.e. $\Phi(z_{\delta/2}) = 1-\delta/2$. Then,

$$\widehat{C}_n \pm z_{\delta/2}\frac{s_C}{\sqrt{n}}$$

is an asymptotically valid $(1-\delta)$ confidence interval for C.

3.2.2 Numerical illustration.

The theoretical example is implemented in MATLAB in order to illustrate the Monte Carlo estimation procedure for European option pricing. The European call price is computed both analytically using the closed-form Black–Scholes formula and numerically via Monte Carlo simulation of the discounted payoff.
The MATLAB implementation details are reported in Appendix A.1, while the present section focuses on the discussion of the numerical results.

The numerical evidence confirms the convergence of the Monte Carlo estimator $\widehat{C}_n$ to the analytical Black–Scholes price C_{BS} as the number of simulated paths increases, as illustrated in Figure 2a. In particular, using $N = 10^6$ simulations, the following estimates are obtained:

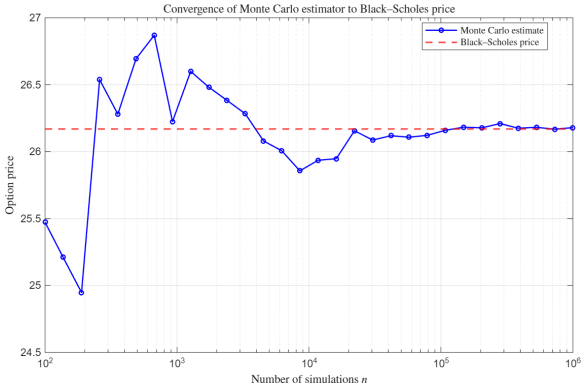
$$C_{\mathrm{BS}} = 26.1690, \quad \widehat{C}_{\mathrm{MC}} = 26.1442, \quad 95\%\ \mathrm{CI} = [26.1003, 26.1881].$$

The analytical price lies within the Monte Carlo confidence interval, providing further empirical confirmation of the theoretical convergence results.

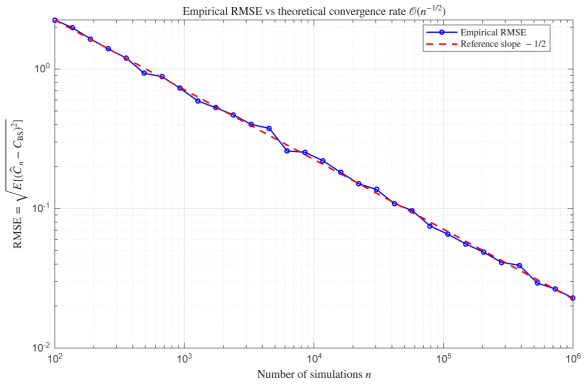
Figure 2b reports the empirical Root Mean Square Error (RMSE) of the Monte Carlo estimator as a function of the number of simulations n. For each value of n, the RMSE is estimated over $k = 200$ independent replications as:

$$\widehat{\mathrm{RMSE}}(n) = \sqrt{\frac{1}{k}\sum_{j=1}^{k}\left(\widehat{C}_n^{(j)} - C_{\mathrm{BS}}\right)^2}, \tag{10}$$

where $\widehat{C}_n^{(j)}$ denotes the Monte Carlo estimate obtained at replication j. This quantity provides an empirical estimate of $\sqrt{\mathbb{E}[|\widehat{C}_n - C_{\mathrm{BS}}|^2]}$, which by the unbiasedness of the estimator equals $\sigma_f/\sqrt{n}$, where $\sigma_f^2 = \mathrm{Var}(e^{-rT}(S_T - K)^+)$. The log–log plot confirms that the empirical RMSE decreases at the theoretical rate $\mathcal{O}(n^{-1/2})$, consistent with the convergence analysis in Glasserman [2004].



(a) Convergence of the Monte Carlo estimator $\widehat{C}_n$ to the analytical Black–Scholes price C_{BS} as the number of simulations increases.



(b) Log–log plot of the empirical RMSE estimated over $k = 200$ independent replications for each n. The dashed line represents the theoretical convergence rate $\mathcal{O}(n^{-1/2})$.

Figure 2: Convergence and accuracy of the Monte Carlo estimator for the European call price: convergence to the Black–Scholes price (left) and empirical RMSE decay (right).

3.3 Discretization of the Stochastic Process

In the Monte Carlo example of Section 3.2, the pricing of a European call option was performed by directly simulating the terminal value of the underlying asset, exploiting the closed-form solution of the Geometric Brownian Motion. This approach is feasible because the option payoff depends only on the asset price at maturity and does not require information about the intermediate evolution of the process.

However, in more general settings, in particular when dealing with path-dependent payoffs or early-exercise features, it becomes necessary to simulate the entire trajectory of the underlying asset over time. In such cases, the continuous-time process must be represented on a discrete time grid. Consider the stock price S_t modeled as a Geometric Brownian Motion under the risk-neutral measure

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

where r denotes the risk-free rate, σ the volatility, and W_t a standard Brownian motion.

Let $0 = t_0 < t_1 < \cdots < t_N = T$ be a uniform partition of the time interval $[0, T]$ with step size $\Delta t = t_{i+1} - t_i$. The discretization of time allows the simulation of the asset price at a finite number of observation dates, which is required by Monte Carlo algorithms and by the presence of early-exercise features.

The recursive simulation scheme follows directly from the analytical solution of the GBM derived in Section 2.5, where Itô's Lemma applied to $\ln S_t$ yields the explicit solution

$$\ln S_{t_{i+1}} - \ln S_{t_i} = \left(r - \tfrac{1}{2}\sigma^2\right)\Delta t + \sigma\left(W_{t_{i+1}} - W_{t_i}\right).$$

Under the risk-neutral measure, the Brownian increment satisfies $W_{t_{i+1}} - W_{t_i} \sim \mathcal{N}(0, \Delta t)$, so writing $W_{t_{i+1}} - W_{t_i} = \sqrt{\Delta t}\,Z_i$ with $Z_i \sim \mathcal{N}(0,1)$ and exponentiating gives the exact recursive relation

$$S_{t_{i+1}} = S_{t_i} \exp\Big\{\left(r - \tfrac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}\,Z_i\Big\}, \quad Z_i \sim \mathcal{N}(0,1) \text{ i.i.d.}$$

Because this formula is derived from the analytical solution of the SDE rather than from a numerical approximation, it introduces *no discretization error* in the simulation of the process at the grid nodes.

3.3.1 Consistency of the Recursive Scheme with the Exact Solution

To verify that the recursive scheme is consistent with the exact analytical solution of the GBM, consider a simple example with three time instants $0 = t_0 < t_1 < t_2$, with $t_1 = \Delta t$ and $t_2 = 2\Delta t$. The exact solution formula (8) gives directly

$$S_{t_2} = S_0 \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)2\Delta t + \sigma\sqrt{2\Delta t}\,\tilde{Z}\Big\}, \quad \tilde{Z} \sim \mathcal{N}(0,1).$$

On the other hand, applying the recursive scheme twice yields

$$S_{t_2} = S_{t_1} \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)\Delta t + \sigma\sqrt{\Delta t}\,Z_2\Big\},$$

where $S_{t_1} = S_0 \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)\Delta t + \sigma\sqrt{\Delta t}\,Z_1\Big\}$, and $Z_1, Z_2 \sim \mathcal{N}(0,1)$ are independent. Substituting the expression for S_{t_1} and using the additive property of exponents gives

$$\begin{aligned} S_{t_2} &= S_0 \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)\Delta t + \sigma\sqrt{\Delta t}\,Z_1\Big\} \cdot \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)\Delta t + \sigma\sqrt{\Delta t}\,Z_2\Big\} \\ &= S_0 \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)2\Delta t + \sigma\sqrt{\Delta t}\,(Z_1 + Z_2)\Big\}. \end{aligned}$$

Since Z_1 and Z_2 are independent standard Gaussians, each increment satisfies $\sigma\sqrt{\Delta t}\,Z_i \sim \mathcal{N}(0,\, \sigma^2\Delta t)$ for $i = 1, 2$, and by the reproductive property of the Gaussian distribution,

$$\sigma\sqrt{\Delta t}\,(Z_1 + Z_2) \sim \mathcal{N}(0,\, 2\sigma^2\Delta t),$$

which is identical in distribution to $\sigma\sqrt{2\Delta t}\,\tilde{Z}$. Hence the two expressions for S_{t_2} coincide in distribution.

This argument extends to any number N of time steps by induction. After applying the recursive formula N times one obtains

$$S_T = S_0 \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)N\Delta t + \sigma\sqrt{\Delta t}\sum_{i=1}^{N} Z_i\Big\}.$$

Since the Z_i are i.i.d. standard Gaussians, the sum satisfies

$$\sigma\sqrt{\Delta t}\sum_{i=1}^{N} Z_i \sim \mathcal{N}\big(0,\, N\sigma^2\Delta t\big) = \mathcal{N}\big(0,\, \sigma^2 T\big)\,,$$

which is identical in distribution to $\sigma\sqrt{T}\,\tilde{Z}$ with $\tilde{Z} \sim \mathcal{N}(0,1)$. Together with $N\Delta t = T$, this recovers the exact terminal distribution

$$S_T = S_0 \exp\Big\{\big(r - \tfrac{1}{2}\sigma^2\big)T + \sigma\sqrt{T}\,\tilde{Z}\Big\}, \quad \tilde{Z} \sim \mathcal{N}(0,1),$$

confirming that the recursive discretization introduces no distributional error at the grid nodes for any choice of N.

3.3.2 Practical Implications of the Time Grid Choice

Although the transition of the GBM is simulated exactly at the grid points, it is important to distinguish between three conceptually separate sources of error in the Monte Carlo pricing framework.
The first is the *process discretization error*, which arises when the exact conditional distribution of the process over a finite time step is not available and must be approximated numerically, for instance via an Euler– Maruyama scheme. For the GBM, this source of error is entirely absent, since the log-normal transition is known in closed form and reproduced exactly by the recursive scheme.
The second is the *payoff or stopping approximation error*, which arises independently of the quality of the process simulation and is intrinsic to the structure of the contract being priced. For path-dependent options, the payoff is evaluated only at the observation dates in the grid, so a coarser grid may fail to capture fine-grained features of the path. For American options, the continuous-time optimal stopping problem is replaced by a discrete-time optimization over the finite set of exercise dates $\{t_0, t_1, \ldots, t_N\}$, which introduces an approximation error that vanishes as $\Delta t \to 0$. In both cases, a finer time grid improves pricing accuracy at the expense of higher computational cost.
The third is the *sampling error*, inherent to Monte Carlo estimation itself: since expectations are approximated by empirical averages over a finite number of paths, a statistical error of order $\mathcal{O}(N^{-1/2})$ remains for any finite N, independently of the time grid. This source of error has been discussed in detail in Section 3.2.

The discretized simulation framework introduced in this section constitutes the basic building block for the Monte Carlo pricing of path-dependent and early-exercise derivatives, and will be employed in the American option pricing algorithm discussed in the following section.

3.4 From European to American options: optimal stopping and continuation value

The Monte Carlo framework developed in this chapter has been applied to the valuation of European options, whose payoff depends only on the value of the underlying asset at a fixed maturity date T. In this case, option pricing reduces to the computation of a discounted expectation under the risk–neutral probability measure.

American options, however, introduce a fundamental conceptual and numerical complication due to the possibility of early exercise. The holder is allowed to exercise the contract at any time prior to expiration, and must therefore decide *when* to exercise in an optimal way. As a consequence, the payoff is no longer determined by the terminal value of the underlying alone, but by a sequence of exercise decisions taken dynamically over time.

Formally, the valuation of an American option can be classified as an **optimal stopping problem** in continuous time. Let $\mathcal{T}$ denote the set of admissible stopping times taking values in the interval $[0, T]$. Assuming a constant risk–free rate r, the arbitrage–free price of the American option at time zero is given by:

$$V_0 = \sup_{\tau \in \mathcal{T}} \mathbb{E}^{\mathbb{Q}} \left[e^{-r\tau} \, \phi\big(S(\tau)\big) \right], \tag{11}$$

where ϕ denotes the payoff function. The supremum reflects the holder's ability to optimally choose the exercise time in order to maximize the expected discounted payoff.

This formulation highlights the key distinction with respect to European options: pricing and exercise decisions are intrinsically linked and must be determined simultaneously.
At any admissible exercise time $t \in [0, T]$, the holder chooses between two alternatives:

- the **immediate exercise value**, given by the payoff $\phi(S(t))$;
- the **continuation value**, defined as the conditional expectation of the discounted payoff obtained by continuing the contract, given the information available at time t.

The optimal stopping rule is to exercise the option whenever the immediate exercise value exceeds the continuation value.

Although the underlying price process is Markovian, the continuation value is an unknown function of the current state and cannot, in general, be computed in closed form. As a result, the valuation of American options naturally leads to a backward dynamic programming problem, in which conditional expectations must be approximated at intermediate times along the evolution of the underlying asset.

From a financial perspective, early exercise is not always optimal. For example, as seen in previous chapters, with no dividends, it is never optimal to exercise an American call option before maturity, implying that American and European call prices coincide (Theorem 2) .
In contrast, American put options typically admit early exercise, especially when the option is deep in the money or when interest rates are high as this allows the holder to receive the strike price sooner and invest it at the risk-free rate.
The additional value arising from the possibility of exercising before maturity is commonly called **early exercise premium**.

From a numerical point of view, the presence of early exercise represents a major challenge for Monte Carlo methods. While Monte Carlo simulation is well suited for estimating expectations of payoffs evaluated at a fixed maturity, it does not directly provide a mechanism for determining optimal stopping rules.
In particular, the continuation value involves conditional expectations given the current state of the system.

3.4.1 Example: American put pricing

Consider a put option with strike price K and maturity T, written on an underlying asset whose risk–neutral dynamics follows a Geometric Brownian Motion

$$dS(t) = rS(t)\,dt + \sigma S(t)\,dW(t),$$

where r is the constant risk–free interest rate, $\sigma > 0$ is the volatility parameter and $W(t)$ is a standard Brownian motion.

Specializing (11) to the put payoff $\phi(x) = (K - x)^+$, the arbitrage–free value of the American put at time zero can be expressed as

$$V_0 = \sup_{\tau \in \mathcal{T}} \mathbb{E}\left[e^{-r\tau} \left(K - S(\tau)\right)^+\right], \tag{12}$$

where $(x)^+ = \max\{x, 0\}$ ensures that exercising the option out of the money yields a zero payoff rather than a negative value.

Under regularity conditions, the optimal stopping time τ^* solving (12) can be characterized in terms of $b^*(t)$ such that

$$\tau^* = \inf\left\{\, t \geq 0 : S(t) \leq b^*(t) \,\right\}. \tag{13}$$

The function $b^*(t)$ is called **optimal exercise boundary**. It partitions the state space $(t, S) \in [0, T] \times \mathbb{R}_+$ into an *exercise region*, where immediate exercise of the option is optimal and a *continuation region*, where it is optimal to keep the option and wait [Peskir and Shiryaev, 2006]. Formally, the exercise region is defined as

$$\mathcal{E} = \{(t, S) : S \leq b^*(t)\},$$

while the continuation region is given by

$$\mathcal{C} = \{(t, S) : S > b^*(t)\}.$$

The optimal stopping rule (13) implies that the option is exercised at the first time the underlying asset price enters the exercise region, that is, when $S(t)$ falls below the optimal exercise boundary $b^*(t)$.
Figure 3 illustrates the exercise and continuation regions for an American put option.

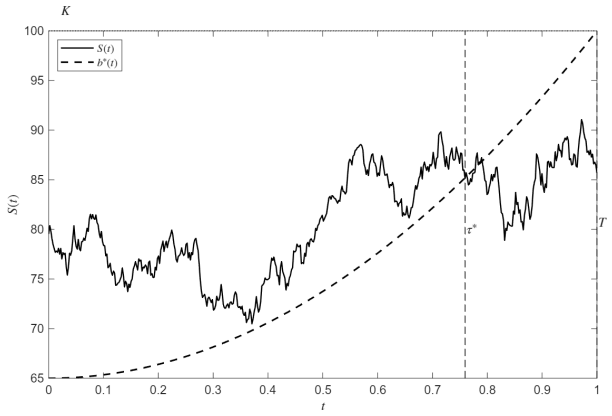


Figure 3: The line shows a sample path of the underlying asset $S(t)$, while the dashed curve represents an artificial optimal exercise boundary $b^*(t)$. The option is exercised at the stopping time τ^*, defined as the first time $S(t)$ crosses the boundary from above.

This representation highlights that the valuation of an American option reduces to the determination of an optimal exercise strategy, together with the evaluation of the corresponding expected discounted payoff under the risk–neutral probability measure.

3.4.2 Discrete exercise dates and Bermudan approximation

In line with the framework of Glasserman [2004], the focus is restricted to simulation–based methods for pricing American options, under the assumption that exercise is allowed only at a finite set of predetermined dates:

$$0 < t_1 < t_2 < \cdots < t_m = T.$$

Options with this structure are commonly referred to as *Bermudan options*, as they lie between European options, exercisable only at maturity and continuously exercisable American options.
More generally, this discrete–time formulation provides a numerical approximation of the continuous–time optimal stopping problem, allowing one to study the convergence of option prices as the number of exercise opportunities increases. Even in this finite–exercise setting, determining an optimal exercise strategy remains a non-trivial problem, particularly when Monte Carlo simulation techniques are employed.
In the following sections, the term *American option* is therefore used in a broad sense, referring both to Bermudan contracts and to continuously exercisable American options.

Let $\{t_0, t_1, \ldots, t_m\}$ denote the set of exercise dates, with $t_0 = 0$ and $t_m = T$. The state of the underlying Markov process at time t_i is denoted by

$$X_i := X(t_i), \qquad i = 0, \ldots, m,$$

so that the sequence $\{X_0, X_1, \ldots, X_m\}$ forms a discrete-time Markov chain taking values in $\mathbb{R}^d$. The pricing problem is then based on the simulation of this Markov chain under the risk–neutral measure.

In practical implementations, simulating the transition from X_i to X_{i+1} may require the simulation of intermediate states between t_i and t_{i+1}.
This typically involves the introduction of a finer time discretization than that induced by the exercise dates alone. As a consequence, the simulated values of $\{X_i\}$ may be affected by discretization error.
Since the focus of this chapter is on the optimal stopping aspect of the problem rather than on time discretization issues, such errors are ignored and the states $X_1, \ldots, X_m$ are assumed to be simulated exactly.

3.4.3 Dynamic programming and Continuation Value

Let $\hat{\phi}_i(x)$ denote the payoff obtained by exercising the option at time t_i when the state of the underlying process is $X_i = x$. Define $\hat{V}_i(x)$ as the value of the option at time t_i, conditional on $X_i = x$ and assuming that the option has not yet been exercised.
The objective is to determine the initial value $\hat{V}_0(X_0)$, which can be computed recursively by backward induction starting from the maturity of the option.
At the final exercise date, $t_m = T$, the option must be exercised and its value coincides with the terminal payoff:

$$\hat{V}_T(x) = \hat{\phi}_T(x). \tag{14}$$

At any earlier exercise date t_{i-1}, the holder faces a trade-off between immediate exercise and continuation value. This leads to the backward recursion

$$\hat{V}_{i-1}(x) = \max\left\{ \hat{\phi}_{i-1}(x),\ \mathbb{E}\Big[D_{i-1,i}(X_i)\, \hat{V}_i(X_i) \mid X_{i-1} = x \Big] \right\}, \quad i = 1, \ldots, m. \tag{15}$$

where $D_{i-1,i}(X_i)$ is the discount factor from t_{i-1} to t_i. This formula compares and takes the maximum between the value obtained by exercising immediately and the expected discounted value of continuation.

For notational convenience, it is useful to reformulate the problem in terms of values expressed in time–0 units. Let $D_{0,i}$ denote the cumulative discount factor from time 0 to t_i and define the discounted payoff and value functions

$$\phi_i(x) := D_{0,i}\,\hat{\phi}_i(x), \qquad V_i(x) := D_{0,i}\,\hat{V}_i(x).$$

Under this change of variables, the dynamic programming equations simplify considerably. At maturity,

$$V_T(x) = \phi_T(x), \tag{16}$$

while for $i = 1, \ldots, m$ the backward recursion becomes

$$V_{i-1}(x) = \max\Big\{\phi_{i-1}(x),\ \mathbb{E}\big[V_i(X_i) \,\big|\, X_{i-1} = x\big]\Big\}. \tag{17}$$

This formulation is algebraically equivalent to (15), but avoids the explicit appearance of discount factors inside the conditional expectations. As a result, it will be particularly convenient for the development of simulation–based algorithms for the approximation of continuation values.

3.4.4 Stopping Rules and Continuation Values

The dynamic programming relations in (14)–(15) and (16)–(17) characterize the evolution of the American option value through backward induction. An equivalent and often insightful viewpoint describes the pricing problem directly in terms of *stopping rules*, their associated *exercise regions* and the *continuation values*.

Stopping rules and exercise regions. Let $\{X_0, X_1, \ldots, X_m\}$ denote the discrete–time Markov chain describing the state of the underlying process at the admissible exercise dates $\{t_0, \ldots, t_m\}$, with $t_0 = 0$ and $t_m = T$. A stopping time τ taking values in $\{0, 1, \ldots, m\}$ represents an exercise strategy, not necessarily optimal, that prescribes at which exercise date the option is exercised, based on the information available up to that time.

Given the discounted payoffs $\{\phi_i(x)\}_{i=0}^m$, the arbitrage–free price of the American option is obtained by maximizing over all admissible stopping times:

$$V_0(X_0) = \sup_{\tau} \mathbb{E}[\phi_\tau(X_\tau)]\,,$$

which is the discrete–time analogue of the optimal stopping formulation in (11).

Conversely, consider a family of discounted value functions $\{V_i(x)\}_{i=0}^m$ satisfying the dynamic programming relations

$$V_m(x) = \phi_m(x), \qquad V_{i-1}(x) = \max\Big\{\phi_{i-1}(x),\, \mathbb{E}[V_i(X_i) \mid X_{i-1} = x]\Big\}, \quad i = 1, \ldots, m.$$

Such a family induces a stopping rule

$$\tau = \min\{i \in \{1, \dots, m\} : \phi_i(X_i) \geq V_i(X_i)\}, \tag{18}$$

which prescribes to exercise at the first date t_i where immediate exercise is greater than continuation.

For each exercise date t_i, the associated *exercise region* and *continuation region* in the state space are defined as

$$\mathcal{E}_i := \{x \in \mathbb{R}^d : \phi_i(x) \geq V_i(x)\}, \qquad \mathcal{C}_i := \{x \in \mathbb{R}^d : \phi_i(x) < V_i(x)\}. \tag{19}$$

Continuation values. The continuation value measures the benefit of keeping the option alive rather than exercising immediately. In the discounted formulation, the continuation value at time t_i and state x is defined by

$$C_i(x) := \mathbb{E}[V_{i+1}(X_{i+1}) \mid X_i = x], \qquad i = 0, \dots, m-1, \tag{20}$$

with $C_m(x) \equiv 0$, since continuation beyond maturity is not possible.
By construction, value and continuation functions are linked through

$$V_i(x) = \max\{\, \phi_i(x),\, C_i(x) \,\}, \qquad i = 1, \dots, m, \tag{21}$$

and at time 0 (where exercise is not allowed),

$$V_0(X_0) = C_0(X_0).$$

Substituting (21) into (20) yields the backward recursion for continuation values:

$$C_i(x) = \mathbb{E}\big[\max\{\, \phi_{i+1}(X_{i+1}),\, C_{i+1}(X_{i+1}) \,\} \,\big|\, X_i = x\big], \qquad i = 0, \dots, m-1. \tag{22}$$

Thus, either the sequence of value functions $\{V_i\}$ or the sequence of continuation values $\{C_i\}$ uniquely determines the other.

Approximate stopping rules from approximate continuation values. In numerical implementations, the exact continuation values $\{C_i\}$ are rarely available in closed form. Simulation–based methods, such as the least–squares Monte Carlo algorithm of Longstaff and Schwartz [2001], therefore construct approximations $\{\hat{C}_i\}$ to the true continuation values at each exercise date.

Given a family of approximate continuation values $\{\hat{C}_i(x)\}_{i=0}^m$, an approximate value function can be defined as

$$\hat{V}_i(x) := \max\{\phi_i(x), \hat{C}_i(x)\}, \qquad i = 1, \dots, m,$$

and the corresponding stopping rule is

$$\hat{\tau} = \min\Big\{i \in \{1, \dots, m\} : \phi_i(X_i) \geq \hat{C}_i(X_i)\Big\}. \tag{23}$$

This stopping rule coincides with (18) when $\hat{V}_i(x) = V_i(x)$, that is, when $\hat{C}_i(x) = C_i(x)$ for all i and x. In general, $\hat{\tau}$ is suboptimal, and the value $\mathbb{E}[\phi_{\hat{\tau}}(X_{\hat{\tau}})]$ provides a lower bound on the true American option price.

From a Monte Carlo perspective, the central numerical challenge is therefore the accurate estimation of the conditional expectations in (20)–(22). This is precisely the role played by regression–based approaches and other simulation techniques for American–style derivatives.

4 Least Squares Monte Carlo Method

4.1 The Intuition Behind the LSM Approach

The Monte Carlo framework developed in Chapter 3 provides an effective tool for approximating expectations under the risk–neutral measure, particularly in high–dimensional settings and in the presence of path dependence [Glasserman, 2004]. In its standard form, however, Monte Carlo simulation is naturally suited to derivatives whose payoff is evaluated at a fixed maturity date, as in the case of European options, where the option value can be expressed as a single conditional expectation.

American and Bermudan options present a fundamentally different challenge. The holder of such a contract possesses the right to exercise at any admissible date prior to maturity, and rational exercise requires solving an optimal stopping problem: at each exercise date, the holder must compare the immediate exercise payoff against the expected value of keeping the option alive. This introduces a recursive structure in which the value of the contract at any given time depends on future optimal decisions, which in turn depend on future states of the underlying process.

Formally, let $\{t_0, t_1, \ldots, t_N\}$ denote the discrete set of admissible exercise dates and let X_i denote the state of the underlying process at time t_i under the risk-neutral measure $\mathbb{Q}$. The value function satisfies the dynamic programming recursion

$$V_i(x) = \max\left\{\phi(x),\, C_i(x)\right\},$$

where $\phi(x)$ denotes the immediate exercise payoff and the continuation value is defined as

$$C_i(x) = \mathbb{E}^{\mathbb{Q}}\left[e^{-r\Delta t} V_{i+1}(X_{i+1}) \mid X_i = x\right].$$

The optimal stopping time is then

$$\tau^* = \min\left\{t_i \,:\, \phi(X_i) > C_i(X_i)\right\},$$

and the American option price at t_0 is

$$V_0 = \mathbb{E}^{\mathbb{Q}}\left[e^{-r\tau^*}\phi(X_{\tau^*})\right].$$

While this recursion is theoretically well-defined, its numerical implementation via Monte Carlo simulation is non-trivial. Forward simulation readily generates sample paths of the underlying process, but does not directly provide the conditional expectations required to evaluate $C_i(x)$. In complex or high-dimensional settings, alternative methods such as finite differences or binomial trees become computationally infeasible due to the curse of dimensionality.

To address this issue, Longstaff and Schwartz [2001] proposed the Least Squares Monte Carlo (LSM) method. The key idea of the LSM approach is to approximate the continuation value function by projecting future realized discounted option values onto a finite-dimensional space spanned by chosen basis functions of the current state variables. This transforms the problem of estimating conditional expectations into a cross-sectional regression problem.

At each exercise date t_i, the discounted option values along simulated paths that continue beyond that date serve as observations of the continuation value, becoming the dependent variable in a regression against basis functions of the current state variables, such as polynomials of the underlying asset price. The fitted values from this regression approximate the conditional expectation defining the continuation value.

By repeating this estimation procedure at each exercise date—starting from maturity and proceeding backward in time—the LSM method yields an approximation of the optimal exercise strategy along every simulated path. The rational exercise policy dictates exercising whenever the immediate payoff exceeds the estimated continuation value. Once this exercise policy has been determined, the American option price is computed as the discounted average of the realized cash flows across all simulated paths.

In essence, the Least Squares Monte Carlo algorithm combines two fundamental components:

1. Monte Carlo simulation is employed to generate a large number of possible future paths of the underlying asset under the risk–neutral measure.

2. At each potential exercise date, least–squares regression is applied to the simulated data in order to estimate the expected future value of continuation as a function of the current state.

The estimated continuation value serves as a benchmark for the exercise decision: if the immediate exercise payoff exceeds this estimate, the option is exercised; otherwise, it is kept alive. Iterating this logic backward through time produces both an estimated optimal stopping policy and a consistent valuation of the American option. The method remains computationally efficient even in high–dimensional settings, where finite–difference or binomial tree methods typically become infeasible.

In the following sections, the LSM algorithm is discussed in detail and its convergence properties are analyzed. Before introducing it formally, a simple numerical example adapted from Hull [2021] and Longstaff and Schwartz [2001] is presented to illustrate the backward induction mechanism and clarify the role of regression and In-The-Money (ITM) filtering in approximating the optimal exercise strategy.

Problem setup. Consider a three–year American put option written on a non–dividend–paying stock. The option can be exercised at the end of years $t = 1, 2$, and 3. The model parameters are

$$S_0 = 1.00, \qquad K = 1.10, \qquad r = 6\% \text{ (continuously compounded)}.$$

To illustrate the method, eight simulated stock price paths, taken directly from Longstaff and Schwartz [2001], are used throughout this example.[4] Each path represents a possible realization of the stock price dynamics under the risk-neutral measure. The simulated prices are reported in Table 2.

[4]In practical applications, the number of simulated paths is typically several orders of magnitude larger.

Table 2: Sample stock price paths for the American put option example.

Path	$t=0$	$t=1$	$t=2$	$t=3$
1	1.00	1.09	1.08	1.34
2	1.00	1.16	1.26	1.54
3	1.00	1.22	1.07	1.03
4	1.00	0.93	0.97	0.92
5	1.00	1.11	1.56	1.52
6	1.00	0.76	0.77	0.90
7	1.00	0.92	0.84	1.01
8	1.00	0.88	1.22	1.34

As a preliminary step, it is interesting to value the option as if it were European, i.e. exercisable only at maturity. This value is a natural lower bound for the American option price.
The European put payoff at $t = 3$ is given by

$$\phi_{EU} = \max(K - S_3, 0),$$

and the corresponding discounted Monte Carlo estimator is

$$V_{\text{EU}} = e^{-rT} \frac{1}{N} \sum_{j=1}^{N} \max(K - S_{3,j}, 0).$$

Using the terminal prices reported in Table 2, the non-zero payoffs at maturity are:

$$\begin{aligned} &\max(1.10 - 1.03, 0) = 0.07, \qquad &&\max(1.10 - 0.92, 0) = 0.18, \\ &\max(1.10 - 0.90, 0) = 0.20, \qquad &&\max(1.10 - 1.01, 0) = 0.09. \end{aligned}$$

corresponding to paths 3, 4, 6, and 7 respectively. Hence, the European value is

$$V_{\text{EU}} = \frac{(0.07 + 0.18 + 0.20 + 0.09)e^{-0.06 \cdot 3}}{8} \approx 0.0564.$$

Any American option value exceeding this benchmark reflects the economic value of the early exercise feature.

Step 1: Regression at $t = 2$. The LSM algorithm proceeds backward in time. At each exercise date, the continuation value is estimated via regression using only those paths for which the option is In-The-Money (ITM). From a financial perspective, when the option is Out-Of-The-Money, immediate exercise is never optimal and such paths do not provide information about the optimal exercise boundary.

At time $t = 2$, the option is ITM for paths 1, 3, 4, 6, and 7, i.e. those satisfying $S_{2,j} < K = 1.10$. Note that path 1, although ITM at $t = 2$, expires out of the money at $t = 3$ and therefore contributes a discounted option value of zero to the regression, since $V(t_3, S_{t_3}^{(1)}) = 0$. This illustrates that ITM filtering is applied at the *current* exercise date,

not at maturity. For these paths, the Least Squares Monte Carlo method approximates the **continuation value**

$$C_2(S_2) = \mathbb{E}\big[e^{-r\Delta t} V_3 \mid S_2\big]$$

by projecting it onto a finite-dimensional space spanned by basis functions of the stock price. In this example, a quadratic polynomial basis is employed:

$$\hat{C}_2(S) = a + bS + cS^2.$$

The coefficients (a, b, c) are estimated by ordinary least squares (OLS) using simulated data. For each ITM path i, the explanatory variable is the observed stock price at $t = 2$, denoted by $S_{2,i}$. The dependent variable is the discounted option value at the next exercise date:

$$Y_i = e^{-r\Delta t} V_3^{(i)}.$$

Since $t = 3$ is the maturity date, the option value coincides with the intrinsic value:

$$V_3^{(i)} = \max(K - S_3^{(i)}, 0).$$

Using the simulated paths, the regression vectors are therefore

$$S = \begin{bmatrix} S_{2,1} \\ S_{2,3} \\ S_{2,4} \\ S_{2,6} \\ S_{2,7} \end{bmatrix} = \begin{bmatrix} 1.08 \\ 1.07 \\ 0.97 \\ 0.77 \\ 0.84 \end{bmatrix}, \qquad Y = \begin{bmatrix} e^{-0.06}\max(1.10 - 1.34, 0) \\ e^{-0.06}\max(1.10 - 1.03, 0) \\ e^{-0.06}\max(1.10 - 0.92, 0) \\ e^{-0.06}\max(1.10 - 0.90, 0) \\ e^{-0.06}\max(1.10 - 1.01, 0) \end{bmatrix} = \begin{bmatrix} 0.00 \\ 0.07e^{-0.06} \\ 0.18e^{-0.06} \\ 0.20e^{-0.06} \\ 0.09e^{-0.06} \end{bmatrix}.$$

Matrix formulation of the regression. Define the design matrix

$$X = \begin{bmatrix} 1 & S_{2,1} & S_{2,1}^2 \\ 1 & S_{2,3} & S_{2,3}^2 \\ 1 & S_{2,4} & S_{2,4}^2 \\ 1 & S_{2,6} & S_{2,6}^2 \\ 1 & S_{2,7} & S_{2,7}^2 \end{bmatrix} = \begin{bmatrix} 1 & 1.08 & 1.1664 \\ 1 & 1.07 & 1.1449 \\ 1 & 0.97 & 0.9409 \\ 1 & 0.77 & 0.5929 \\ 1 & 0.84 & 0.7056 \end{bmatrix}.$$

Let $\beta = (a, b, c)^\top$. The regression model can be written compactly as

$$Y = X\beta + \varepsilon,$$

where ε is the vector of regression residuals.

OLS estimation. The ordinary least squares estimator minimizes the sum of squared residuals

$$\min_{a,b,c} \sum_{i=1}^{5} \left(Y_i - a - bS_{2,i} - cS_{2,i}^2\right)^2,$$

and admits the closed-form solution

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y.$$

Substituting the numerical values of X and Y into this expression yields

$$\hat{a} = -1.070, \qquad \hat{b} = 2.983, \qquad \hat{c} = -1.814.$$

Therefore, the estimated continuation value function at time $t = 2$ is

$$\hat{C}_2(S) = -1.070 + 2.983S - 1.814S^2.$$

Exercise decision at $t = 2$. For each ITM path, the optimal decision rule compares the immediate exercise value $K - S_{2,i}$ with the estimated continuation value $\hat{C}_2(S_{2,i})$:

$$V_2(S_{2,i}) = \max\Big(K - S_{2,i},\ \hat{C}_2(S_{2,i})\Big)\,.$$

Early exercise is optimal whenever $K - S_{2,i} > \hat{C}_2(S_{2,i})$ and continuation is optimal otherwise. The resulting decisions are summarized in Table 3.

Table 3: Stock prices and continuation values at $t = 2$ for ITM paths.

Path	$S_{2,i}$	$\hat{C}_2(S_{2,i})$	$K - S_{2,i}$	**Exercise?**
1	1.08	0.0369	0.02	No
3	1.07	0.0461	0.03	No
4	0.97	0.1176	0.13	Yes
6	0.77	0.1520	0.33	Yes
7	0.84	0.1565	0.26	Yes

Figure 4 provides a visual representation of this regression step, as produced by the MATLAB implementation of the LSM algorithm. Each dot represents the realized discounted option values Y_i of one ITM path, plotted against the corresponding stock price $S_{2,i}$; the green curve is the fitted parabola $\hat{C}_2(S)$, which is the OLS approximation of the continuation value. Red dots identify paths for which $K - S_{2,i} > \hat{C}_2(S_{2,i})$ (paths 4, 6, and 7), which are therefore exercised early; blue dots identify paths for which continuation is optimal (paths 1 and 3). Note that path 1 contributes $Y_1 = 0$ to the regression despite being ITM at $t = 2$, since it expires out of the money at $t = 3$.

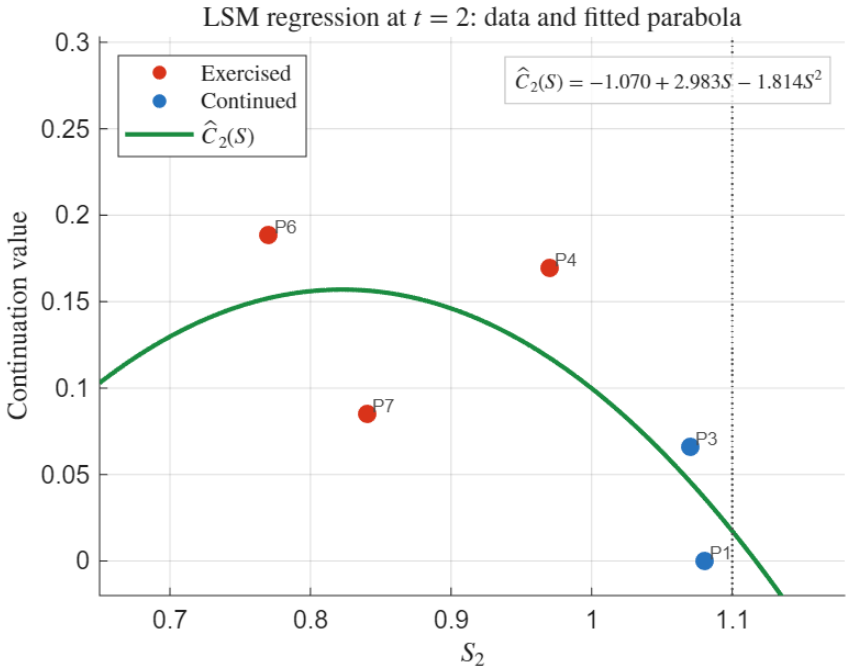


Figure 4: LSM regression at $t = 2$: discounted option values $Y_i = e^{-r\Delta t}V(t_3, S_{t_3}^{(i)})$ for ITM paths (dots) and fitted quadratic continuation value $\widehat{C}_2(S)$ (green curve). Red dots: paths exercised early ($K - S_{2,i} > \hat{C}_2(S_{2,i})$, i.e. paths 4, 6, 7); blue dots: paths continued ($K - S_{2,i} \leq \hat{C}_2(S_{2,i})$, i.e. paths 1, 3).

Step 2: Regression at $t = 1$ and optimal stopping. The algorithm then moves backward to time $t = 1$. At this exercise date, the option is ITM for paths 1, 4, 6, 7 and 8. As in the previous step, the continuation value

$$C_1(S_1) = \mathbb{E}\left[e^{-r\Delta t}V_2 \mid S_1\right]$$

where V_2 denotes the option value resulting from the optimal exercise decision taken at time $t = 2$, is again approximated using a quadratic polynomial basis:

$$\hat{C}_1(S) = a + bS + cS^2.$$

The regression data are collected in Table 4. For each ITM path, the dependent variable is the realized discounted cash flow generated by the optimal stopping rule at $t = 2$: paths exercised at $t = 2$ contribute $K - S_2^{(i)}$, while paths continued to maturity contribute $\max(K - S_3^{(i)}, 0)$.

Table 4: Regression data at $t = 1$: ITM paths.

Path	$S_{1,i}$	$V_2^{(i)}$	$Y_i = e^{-r\Delta t}V_2^{(i)}$
1	1.09	0.00	0.00
4	0.93	0.13	$0.13\, e^{-0.06}$
6	0.76	0.33	$0.33\, e^{-0.06}$
7	0.92	0.26	$0.26\, e^{-0.06}$
8	0.88	0.00	0.00

Applying the OLS formula $\hat{\beta} = (X^\top X)^{-1} X^\top Y$ yields

$$\hat{a} = 2.038, \qquad \hat{b} = -3.335, \qquad \hat{c} = 1.356,$$

so the estimated continuation value function at $t = 1$ is

$$\hat{C}_1(S) = 2.038 - 3.335\, S + 1.356\, S^2.$$

Exercise decision at $t = 1$. For each ITM path, the optimal rule is

$$V_1(S_{1,i}) = \max\Big(K - S_{1,i},\ \hat{C}_1(S_{1,i})\Big).$$

The results are reported in Table 5. Early exercise at $t = 1$ is optimal for paths 4, 6, 7, and 8, while path 1 is optimally continued to the next exercise date.

Table 5: Stock prices and continuation values at $t = 1$ for ITM paths.

Path	$S_{1,i}$	$\hat{C}_1(S_{1,i})$	$K - S_{1,i}$	**Exercise?**
1	1.09	0.0140	0.01	No
4	0.93	0.1110	0.17	Yes
6	0.76	0.2870	0.34	Yes
7	0.92	0.1180	0.18	Yes
8	0.88	0.1540	0.22	Yes

Figure 5 shows the regression at $t = 1$, again as produced by the MATLAB implementation. As before, the dots represent the realized discounted option values Y_i, while the exercise decision is determined by comparing the intrinsic value $K - S_{1,i}$ with the fitted continuation value $\hat{C}_1(S_{1,i})$. Two paths deserve special attention. Path 8, despite contributing $Y_8 = 0$ to the regression (its option value at $t = 2$ was zero, since it was out of the money at both $t = 2$ and $t = 3$), is nonetheless exercised at $t = 1$ because its intrinsic value $K - S_{1,8} = 0.22$ exceeds $\hat{C}_1(0.88) \approx 0.154$. Path 1, by contrast, has intrinsic value $K - S_{1,1} = 0.01$ marginally below $\hat{C}_1(1.09) \approx 0.014$ and is therefore optimally continued to maturity.

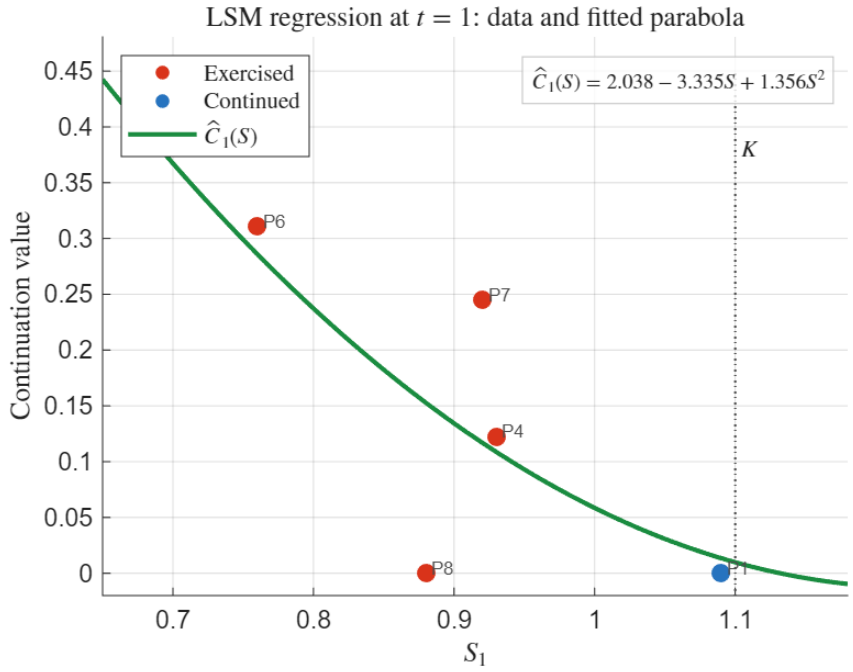


Figure 5: LSM regression at $t = 1$: discounted option values $Y_i = e^{-r\Delta t}V(t_2, S_{t_2}^{(i)})$ for ITM paths (dots) and fitted quadratic continuation value $\widehat{C}_1(S)$ (green curve). Red dots: paths exercised early ($K - S_{1,i} > \hat{C}_1(S_{1,i})$, i.e. paths 4, 6, 7, 8); blue dots: paths continued ($K - S_{1,i} \leq \hat{C}_1(S_{1,i})$, i.e. path 1).

American value and early exercise premium. Once the optimal stopping time τ has been determined for each simulated path, the American option value is computed as the Monte Carlo estimator

$$V_{\text{Amer}} = \frac{1}{N}\sum_{j=1}^{N} e^{-r\tau_j}\,\text{Payoff}_j,$$

where τ_j denotes the optimal exercise time along path j and

$$\text{Payoff}_j = \max\Big(K - S_{\tau_j}^{(j)},\, 0\Big)$$

is the put payoff realized at exercise. For paths along which the LSM backward induction never identifies early exercise as optimal, $\tau_j = T$ and the payoff reduces to the European put payoff at maturity.

In the present example, the discounted cash flows generated by the optimal policy are summarized in Table 6.

Table 6: Optimal exercise policy and realized discounted cash flows.

Path	**Exercise time τ_j**	**Payoff**	$e^{-r\tau_j} \cdot$ Payoff
1	—	0	0
2	—	0	0
3	$t = 3$	0.07	$0.07\, e^{-0.06 \cdot 3}$
4	$t = 1$	0.17	$0.17\, e^{-0.06}$
5	—	0	0
6	$t = 1$	0.34	$0.34\, e^{-0.06}$
7	$t = 1$	0.18	$0.18\, e^{-0.06}$
8	$t = 1$	0.22	$0.22\, e^{-0.06}$

Hence,

$$V_{\text{Amer}} = \frac{1}{8}\left(0.07e^{-0.06\cdot 3} + 0.17e^{-0.06} + 0.34e^{-0.06} + 0.18e^{-0.06} + 0.22e^{-0.06}\right) = 0.1144.$$

The difference between the American and European prices defines the **early exercise premium**:

$$\text{EEP} = V_{\text{Amer}} - V_{\text{EU}} = 0.1144 - 0.0564 = 0.0580.$$

This quantity measures the economic value of the flexibility to optimally exercise the option prior to maturity.

4.2 Least Squares Monte Carlo algorithm

The Least Squares Monte Carlo (LSM) method, introduced by Longstaff and Schwartz [2001], provides a numerically efficient procedure for pricing American-style derivatives within the standard risk-neutral valuation framework. In an arbitrage-free market, there exists an equivalent martingale measure $\mathbb{Q}$ under which discounted asset prices are martingales. Under this measure, derivative prices can be expressed in terms of expected discounted future cash flows [Glasserman, 2004].

Consider an American-style derivative written on an underlying asset whose dynamics are defined on a finite time horizon $[0, T]$. Exercise is allowed at a finite set of dates

$$0 = t_0 < t_1 < \cdots < t_m = T.$$

All expectations are taken under the risk-neutral measure $\mathbb{Q}$ and future cash flows are discounted at the continuously compounded risk-free rate r.

The value of an American option can be characterized as the supremum of the expected discounted payoff over all admissible stopping times taking values in $\{t_1, \ldots, t_m\}$, as in 11.
At maturity T, the optimal strategy is trivial: the option is exercised if and only if the payoff is positive. At earlier exercise dates, however, the holder faces a trade-off between immediate exercise and the value of continuation.

Let $\phi(t_i, S_{t_i})$ denote the payoff obtained from immediate exercise at time t_i. Conditional on the information available at t_i, the continuation value is defined as the expected discounted value of future option value assuming that the option is not exercised at t_i. Formally,

$$C(t_i, S_{t_i}) = \mathbb{E}^{\mathbb{Q}}\left[e^{-r(t_{i+1}-t_i)}V(t_{i+1}, S_{t_{i+1}}) \,\middle|\, S_{t_i}\right],$$

where $V(t_{i+1}, S_{t_{i+1}})$ already incorporates the optimal exercise decision at t_{i+1}.

The optimal exercise rule is therefore characterized by the comparison

$$\text{exercise at } t_i \quad \Longleftrightarrow \quad \phi(t_i, S_{t_i}) \geq C(t_i, S_{t_i}).$$

Since the continuation value involves a conditional expectation that is generally not available in closed form—particularly for path-dependent derivatives or multi-dimensional problems—the LSM method approximates it using regression techniques applied to Monte Carlo simulated paths.

4.2.1 Regression-based approximation of the continuation value

The LSM algorithm proceeds backward in time, exploiting the principle of dynamic programming. Suppose that a large number N of independent sample paths $\{S_{t_i}^{(n)}\}_{i=0}^{m}$, $n = 1, \ldots, N$, have been simulated under the risk-neutral measure. At the final exercise date $t_m = T$, the option value is simply the exercise payoff: $V(T, S_T) = \phi(T, S_T)$. The algorithm then works recursively backward through the exercise dates $t_{m-1}, t_{m-2}, \ldots, t_1$.

At each time step t_i, the continuation value is a conditional expectation that depends on the current state S_{t_i}. The key insight of the LSM method is to approximate this conditional expectation by projecting it onto a finite-dimensional subspace spanned by a set of basis functions. Specifically, we assume that

$$\hat{C}(t_i, S_{t_i}) \approx \sum_{j=1}^{J} \beta_j \, \psi_j(S_{t_i}),$$

where $\{\psi_j\}_{j=1}^{J}$ are predetermined basis functions (e.g., polynomials, Laguerre, Hermite, Chebyshev functions) and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^\top$ is a vector of coefficients to be estimated by least squares regression.

Choice of basis functions. At each exercise date, the Least Squares Monte Carlo (LSM) algorithm approximates the continuation value by projecting an unknown conditional expectation onto a finite-dimensional linear space spanned by a chosen set of basis functions. Theoretical convergence results (see Clément et al. [2002] Stentoft [2004]) show that, as the number of simulated paths tends to infinity and the approximation space becomes sufficiently rich, the method converges to the true value provided that the sequence of regression spaces is dense in $L^2(\Omega, \mathcal{F}, \mathbb{Q})$, the space of square-integrable functions on the probability space $(\Omega, \mathcal{F}, \mathbb{Q})$ introduced in Section 2.2, where $\mathbb{Q}$ denotes the equivalent martingale measure[5] [Longstaff and Schwartz, 2001]. From this perspective,

[5]Formally, a function $f(S_{t_i})$ belongs to this space if and only if

$$\mathbb{E}^{\mathbb{Q}}\left[f(S_{t_i})^2\right] < \infty$$

what matters asymptotically is the approximation space itself, rather than the particular basis used to generate it.

In particular, if two different families of basis functions span the same linear space (for instance, monomials and orthogonal polynomials of the same maximal degree), the associated least-squares estimator represents the same orthogonal projection in L^2. In exact arithmetic, the continuation value approximation is therefore identical: changing from a monomial basis to Laguerre or Hermite polynomials does not modify the space onto which the conditional expectation is projected, but only its coordinate representation.

However, in finite-sample implementations the numerical properties of the chosen basis become crucial. As emphasized in Longstaff and Schwartz [2001] and in Glasserman [2004], monomial bases lead to Vandermonde-type design matrices that may be severely ill-conditioned, especially for high polynomial degrees. This ill-conditioning amplifies statistical noise and round-off errors, potentially degrading the stability of the regression estimates. By contrast, orthogonal polynomial families such as Laguerre, Hermite, or Chebyshev polynomials generate better-conditioned regression matrices, thereby improving numerical robustness and reducing variance inflation in finite samples. Consequently, the practical relevance of the basis choice is primarily numerical and statistical rather than theoretical.

Polynomial basis. The simplest and most widely adopted choice is the polynomial basis. Given a state variable x, the basis functions are defined as

$$\psi_j(x) = x^j, \qquad j = 0, 1, \ldots, M-1.$$

The continuation value estimator therefore belongs to the space of polynomials of degree at most $M-1$:

$$\widehat{C}(t_i, x) = \sum_{j=0}^{M-1} \beta_j x^j = \beta_0 + \beta_1 x^1 + \beta_2 x^2 + \cdots + \beta_{M-1} x^{M-1}.$$

Polynomials provide a flexible and easily interpretable approximation class. By the Weierstrass approximation theorem, any continuous function on a compact interval can be approximated arbitrarily well by polynomials. From a numerical perspective, however, monomials generate Vandermonde-type regression matrices, which may become ill-conditioned as the degree increases. Consequently, moderate values of M are typically preferred in practical implementations.

Weighted Laguerre polynomials. An alternative specification is based on weighted Laguerre polynomials, as proposed in the original LSM formulation [Longstaff and Schwartz, 2001]. The basis functions are defined as

$$\psi_j(x) = e^{-x/2} L_j(x), \qquad j = 0, 1, \ldots, M-1,$$

where $L_j(x)$ denotes the j-th Laguerre polynomial. The first polynomials are given by

$$L_0(x) = 1, \qquad L_1(x) = 1 - x, \qquad L_2(x) = \tfrac{1}{2}(x^2 - 4x + 2),$$

while higher-order terms are generated through the recurrence relation

$$L_{j+1}(x) = \frac{(2j+1-x)L_j(x) - jL_{j-1}(x)}{j+1}.$$

Laguerre polynomials form an orthogonal family with respect to an exponential weight function. The additional factor $e^{-x/2}$ ensures decay for large values of the state variable and contributes to improved numerical conditioning relative to the monomial basis (for more details, see Abramowitz and Stegun [1964]).

Hermite polynomials. Another orthogonal family frequently employed in regression-based methods is given by the probabilist's Hermite polynomials $He_j(x)$, defined by the recurrence relation

$$He_0(x) = 1, \qquad He_1(x) = x, \qquad He_{j+1}(x) = x\,He_j(x) - j\,He_{j-1}(x). \tag{24}$$

The corresponding basis functions are

$$\psi_j(x) = He_j(x), \qquad j = 0, 1, \ldots, M-1.$$

These polynomials are orthogonal with respect to the standard normal density $\varphi(x) = \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2}$ [6]. This property makes them particularly well-suited for financial applications in which the state variable is approximately log-normally distributed: since $\log S_t$ follows a normal distribution under the risk-neutral measure in the Black–Scholes framework, the probabilist's convention provides a natural alignment between the orthogonality structure of the basis and the distribution of the simulated paths.

It should be noted that an alternative convention, known as the physicist's Hermite polynomials $H_j(x)$, is also common in the literature. These satisfy the recurrence

$$H_{j+1}(x) = 2xH_j(x) - 2jH_{j-1}(x)$$

with $H_0(x) = 1$ and $H_1(x) = 2x$ and are orthogonal with respect to e^{-x^2}. Although both families span the same polynomial spaces and yield identical least-squares projections in exact arithmetic, the probabilist's convention is preferred here for its closer alignment with the Gaussian structure of the underlying asset dynamics.

Chebyshev polynomials. Chebyshev polynomials of the first kind constitute another orthogonal basis widely used in approximation problems. They are defined on the interval $[-1, 1]$ by

$$T_j(x) = \cos(j \arccos x)$$

[6] Formally, the probabilist's Hermite polynomials satisfy the orthogonality relation with respect to the L^2 inner product

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} f(x)\, g(x)\, \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2}\, dx,$$

that is, for $j \neq k$,

$$\int_{-\infty}^{+\infty} \mathrm{He}_j(x)\, \mathrm{He}_k(x)\, \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2}\, dx = 0.$$

which yields the recurrence relations

$$T_0(x) = 1, \qquad T_1(x) = x, \qquad T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad j \geq 2.$$

Since Chebyshev polynomials are defined and orthogonal only on $[-1, 1]$, a preliminary rescaling step is required whenever the state variable does not naturally lie in this range. In the LSM context the state variable is the simulated asset price $S_{t_i}^{(n)}$, which is strictly positive and unbounded. A linear map is therefore introduced to bring every observed price into $[-1, 1]$:

$$\tilde{x} = \frac{2(x - x_{\min})}{x_{\max} - x_{\min}} - 1, \tag{25}$$

where $x_{\min}$ and $x_{\max}$ are two scalars that define the effective support of the simulated price distribution.

Choice of $x_{\min}$ and $x_{\max}$. A natural but naive choice is to set $x_{\min}$ and $x_{\max}$ equal to the cross-sectional minimum and maximum of the simulated prices at each individual exercise date t_i:

$$x_{\min}^{(i)} = \min_n S_{t_i}^{(n)}, \qquad x_{\max}^{(i)} = \max_n S_{t_i}^{(n)}.$$

While this guarantees that every price at time t_i is mapped into $[-1, 1]$, it produces a different affine transformation at each step of the backward induction. As a consequence, the Chebyshev basis functions evaluated at t_i and at t_{i-1} span different regions of the input space and the regression coefficients $\widehat{\boldsymbol{\beta}}_i$ estimated at consecutive dates are not directly comparable.

A more robust approach, which ensures that the rescaling is **consistent** across all exercise dates and across all initial price levels, is to fix $x_{\min}$ and $x_{\max}$ once before the algorithm begins, as the global minimum and maximum of the normalised asset price S/K taken over all simulated paths and all exercise dates:

$$x_{\min} = \frac{\min_{n,\,i}\, S_{t_i}^{(n)}}{K}, \qquad x_{\max} = \frac{\max_{n,\,i}\, S_{t_i}^{(n)}}{K},$$

where prices have been normalised by the strike K for additional numerical stability.

With this choice, the same linear map $x \mapsto \tilde{x}$ is applied at every exercise date, ensuring that the basis functions span a consistent polynomial space throughout the entire backward induction and that the regression coefficients are estimated in a common coordinate system at every step.

The basis functions are then defined as

$$\psi_j(x) = T_j(\tilde{x}), \qquad j = 0, 1, \ldots, M - 1,$$

Chebyshev polynomials are useful for regression-based approximation since they are bounded in absolute value on $[-1, 1]$, i.e.

$$|T_j(x)| \leq 1, \quad x \in [-1, 1],$$

and orthogonal with respect to the weight function $\frac{1}{\sqrt{1-x^2}}$. These properties reduce numerical instabilities, limit collinearity among the columns of the regression matrix and improve its conditioning compared to monomial bases.

Restriction to ITM paths. The **regression** is performed using only the paths for which the option is **In-The-Money (ITM)** at time t_i, i.e. those satisfying $\phi(t_i, S_{t_i}^{(n)}) > 0$. This restriction is justified on both economic and numerical grounds:

- **Economic relevance:** For Out-Of-The-Money (OTM) paths, the immediate exercise value is zero (or negative) and the exercise decision is trivial, i.e. never exercise. Including these paths in the regression would introduce noise without adding information relevant to the exercise decision.
- **Numerical stability:** Restricting to ITM paths concentrates the regression effort on the region of the state space where the continuation value meaningfully affects the optimal policy, thereby improving the quality of the approximation.

4.2.2 Ordinary Least Squares estimation

Let $\mathcal{I}_i \subseteq \{1, \dots, N\}$ denote the index set of ITM paths at time t_i and let $N_i = |\mathcal{I}_i|$ be its cardinality. For each path $n \in \mathcal{I}_i$, define the response variable

$$Y_i^{(n)} = e^{-r(t_{i+1}-t_i)} V(t_{i+1}, S_{t_{i+1}}^{(n)}),$$

which represents the discounted option value observed along path n at the following exercise date. Indeed, $V(t_{i+1}, S_{t_{i+1}}^{(n)})$ is the option value at the next time step, already computed in the previous backward step and incorporating the optimal exercise decisions at t_{i+1} and all subsequent times.

Let $\mathbf{\Psi}_i \in \mathbb{R}^{N_i \times J}$ denote the *design matrix* (or *regressor matrix*) whose (n, j)-th entry is given by

$$(\mathbf{\Psi}_i)_{n,j} = \psi_j(S_{t_i}^{(n)}), \qquad n \in \mathcal{I}_i,\ j = 1, \dots, J.$$

Each row of $\mathbf{\Psi}_i$ corresponds to one ITM path and contains the evaluations of all J basis functions at the spot price $S_{t_i}^{(n)}$ for that path. Let $\mathbf{Y}_i \in \mathbb{R}^{N_i}$ be the vector collecting the corresponding responses:

$$\mathbf{Y}_i = \begin{pmatrix} Y_i^{(n_1)} \\ Y_i^{(n_2)} \\ \vdots \\ Y_i^{(n_{N_i})} \end{pmatrix}, \qquad n_1, n_2, \dots, n_{N_i} \in \mathcal{I}_i.$$

The coefficient vector $\boldsymbol{\beta}_i$ is estimated by solving the Ordinary Least Squares (OLS) problem

$$\widehat{\boldsymbol{\beta}}_i = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^J} \left\| \mathbf{Y}_i - \mathbf{\Psi}_i \boldsymbol{\beta} \right\|^2 = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^J} \sum_{n \in \mathcal{I}_i} \left(Y_i^{(n)} - \sum_{j=1}^{J} \beta_j \psi_j(S_{t_i}^{(n)}) \right)^2.$$

This is a standard linear regression problem. Taking the derivative of the objective function with respect to $\boldsymbol{\beta}$ and setting it equal to zero yields the *normal equations*:

$$\mathbf{\Psi}_i^\top \mathbf{\Psi}_i \, \boldsymbol{\beta} = \mathbf{\Psi}_i^\top \mathbf{Y}_i.$$

Provided that the matrix $\mathbf{\Psi}_i^\top \mathbf{\Psi}_i \in \mathbb{R}^{J \times J}$ is invertible (which requires that the columns of $\mathbf{\Psi}_i$ are linearly independent and typically that $N_i \geq J$), the solution is given explicitly by

$$\widehat{\boldsymbol{\beta}}_i = (\mathbf{\Psi}_i^\top \mathbf{\Psi}_i)^{-1} \mathbf{\Psi}_i^\top \mathbf{Y}_i.$$

Once the optimal coefficients $\widehat{\boldsymbol{\beta}}_i$ are determined, the regression function provides an approximation for the continuation value at any state S_{t_i}. Specifically, for a generic spot price S_{t_i} at time t_i, the estimated continuation value is obtained as:

$$\widehat{C}(t_i, S_{t_i}) = \sum_{j=1}^{J} \widehat{\beta}_{i,j}\, \psi_j(S_{t_i}) = \boldsymbol{\psi}(S_{t_i})^\top \widehat{\boldsymbol{\beta}}_i,$$

where $\boldsymbol{\psi}(S_{t_i}) = (\psi_1(S_{t_i}), \ldots, \psi_J(S_{t_i}))^\top$.

Exercise decision and cash-flow update. The estimated continuation value $\widehat{C}(t_i, S_{t_i}^{(n)})$ is used as a decision criterion: for each path $n \in \mathcal{I}_i$, it is compared with the immediate exercise payoff $\phi(t_i, S_{t_i}^{(n)})$ to determine whether early exercise is optimal at time t_i:

- If $\phi(t_i, S_{t_i}^{(n)}) > \widehat{C}(t_i, S_{t_i}^{(n)})$, early exercise is optimal. The cash flow along path n is updated to the immediate exercise payoff $\phi(t_i, S_{t_i}^{(n)})$, the exercise time is set to $\tau_n = t_i$ and all future cash flows along that path are discarded.
- If $\phi(t_i, S_{t_i}^{(n)}) \leq \widehat{C}(t_i, S_{t_i}^{(n)})$, continuation is optimal. The cash flow and exercise time along path n are left unchanged.

For paths $n \notin \mathcal{I}_i$, the immediate exercise payoff is non-positive and early exercise is never optimal; no comparison is therefore performed.

This procedure is repeated recursively for $t_{m-1}, t_{m-2}, \ldots, t_1$, updating at each step the realized cash flows and exercise times according to the estimated exercise policy. Once the backward induction is complete, each path n is associated with a single realized payoff $\phi(\tau_n, S_{\tau_n}^{(n)})$ at the estimated optimal stopping time τ_n. The American option price is then estimated as the discounted average of these payoffs across all simulated paths:

$$\widehat{V}(0, S_0) = \frac{1}{N} \sum_{n=1}^{N} e^{-r\tau_n} \phi\big(\tau_n, S_{\tau_n}^{(n)}\big)\,.$$

4.2.3 Convergence properties

A fundamental property of the LSM algorithm is that it produces a *lower bound* for the true American option value. The LSM procedure does not search over all possible exercise strategies; rather, it generates a specific exercise rule: exercise at time t_i whenever the immediate payoff exceeds the estimated continuation value. The true American option value is defined as the supremum over all admissible stopping times $\mathcal{T}$,

$$V(0, S_0) = \sup_{\tau \in \mathcal{T}} \mathbb{E}\big[e^{-r\tau} \phi(\tau, S_\tau)\big]\,.$$

Since the LSM rule induces only one particular stopping time $\widehat{\tau} \in \mathcal{T}$, the associated value cannot exceed the supremum:

$$\widehat{V}_{\mathrm{LSM}}(0, S_0) = \mathbb{E}\big[e^{-r\widehat{\tau}} \phi(\widehat{\tau}, S_{\widehat{\tau}})\big] \leq V(0, S_0).$$

The quality of this lower bound improves as the regression approximation of the continuation value becomes more accurate. Longstaff and Schwartz [2001] show that, as the basis

functions form a dense system in $L^2(\Omega, \mathcal{F}, \mathbb{Q})$, the estimated exercise rule approaches the true optimal stopping rule and $\widehat{V}_{\mathrm{LSM}}(0, S_0)$ converges to $V(0, S_0)$.
A practical implication is that an excessively large number of basis functions is not required: even a small number, typically J between 3 and 6 for single-asset options, is sufficient to capture the essential features of the optimal exercise boundary. Increasing J beyond this range yields diminishing improvements and may introduce numerical ill-conditioning in the regression matrix.

A more rigorous asymptotic treatment is provided by Stentoft [2004], who analyzes the LSM estimator under a double asymptotic regime in which both N and J are allowed to grow simultaneously. Under regularity conditions on the payoff function and the underlying process, Stentoft [2004] establishes consistency of the LSM estimator and derives the rate at which J must grow relative to N in order for the approximation error to vanish asymptotically. If J grows too slowly relative to N, the bias introduced by the finite-dimensional projection dominates; conversely, if J grows too fast, the estimator suffers from overfitting and variance of the regression estimator inflates. The analysis therefore formalizes the **bias-variance trade-off** inherent in the LSM approach, providing theoretical grounding for the practical recommendation of Longstaff and Schwartz [2001] on the choice of J.

5 Numerical Implementation of the Least Squares Monte Carlo Method

This section describes the numerical implementation of the Longstaff–Schwartz Method (LSM) used to price American put options. The focus here is on explaining the algorithmic workflow and the modeling choices adopted. The complete MATLAB source code is reported in Appendix A.3.

5.1 Model Parameters and Simulation Setup

Consider an American put option written on a non-dividend-paying underlying asset with strike price K and maturity T. Under the risk-neutral measure $\mathbb{Q}$, the asset price dynamics are assumed to follow a Geometric Brownian Motion (GBM):

$$\mathrm{d}S_t \;=\; r\,S_t\,\mathrm{d}t \;+\; \sigma\,S_t\,\mathrm{d}W_t, \tag{26}$$

where r is the continuously compounded risk-free rate, σ is the constant volatility and W_t is a standard Brownian motion.

The baseline parameter configuration adopted in the implementation is summarised in Table 7.

Parameter	**Description**	**Value**
K	Strike price	100
r	Risk-free rate	0.05
σ	Volatility	0.20
T	Time to maturity (years)	$\{1/6,\ 1/2,\ 1\}$
N_{steps}	Number of time steps	100
N_{paths}	Number of Monte Carlo paths	10^3–10^6
M	Number of basis functions	2–6
Basis type	Orthogonal polynomial family	varies by experiment
S_0	Initial stock price (test vector)	$\{80, 90, 100, 110, 120\}$

Table 7: Parameters configuration for the simulation

Although American options allow for continuous exercise in theory, the numerical implementation relies on a discrete-time approximation. The interval $[0, T]$ is partitioned into $N_{\text{steps}} = 100$ equally spaced sub-intervals of length $\Delta t = T/N_{\text{steps}}$, defining a finite set of admissible exercise dates. This discretization effectively transforms the American option into a Bermudan-style contract, whose value converges to the continuous-time American option price as $\Delta t \to 0$. Increasing the number of exercise dates improves the approximation of the early exercise feature, but also increases the computational cost of the regression procedure, since one OLS regression is performed at each backward time step. The choice $N_{\text{steps}} = 100$ therefore represents a compromise between numerical accuracy and computational efficiency.

A preliminary sensitivity analysis of the LSM price with respect to N_{steps} is conducted at $S_0 = 100$ for $N_{\text{steps}} \in \{10,\ 25,\ 50,\ 100,\ 200,\ 500\}$, across three maturities $T \in \{1/6,\ 1/2,\ 1\}$ years, with $N_{\text{paths}} = 2 \times 10^5$ and $M = 4$ basis functions. The results, reported in Section 7.1.6, show that for $N_{\text{steps}} \geq 25$ no configuration produces a price statistically distinguishable from the baseline $N_{\text{steps}} = 100$ at the 95% level and that the LSM price lies strictly above the European BS lower bound for every configuration tested. On the basis of these results, $N_{\text{steps}} = 100$ is adopted throughout as a conservative choice valid across all maturities considered; for $T = 1$ year this corresponds to a time increment of approximately 3.65 days per step.

5.2 Monte Carlo Simulation of the Underlying Process

The LSM algorithm begins with the simulation of N_{paths} independent realizations of the asset price process. The exact log-normal discretization of the GBM is employed:

$$S_{t+\Delta t} \;=\; S_t \exp\!\left[\left(r - \tfrac{1}{2}\sigma^2\right)\Delta t \;+\; \sigma\sqrt{\Delta t}\, Z\right], \qquad Z \sim \mathcal{N}(0,1).$$

This formulation is exact in the sense that it exploits the closed-form solution of the GBM stochastic differential equation, preserving the exact marginal distribution of S_{t_i} at each discretization node t_i avoiding the approximation error inherent in general Euler–Maruyama discretizations of SDEs. In MATLAB, the matrix of random increments (`increments` $\in \mathbb{R}^{N_{\text{paths}} \times N_{\text{steps}}}$) is pre-computed for all paths and time steps simultaneously. The path matrix `S` is then built through a sequential loop: at each step t, the next price S_{t+1} is calculated based on the current price S_t.

Algorithm 2 Simulation of GBM paths

Require: $S_0, r, \sigma, T, N_{\text{steps}}, N_{\text{paths}}$
1: $\Delta t \leftarrow T/N_{\text{steps}}$
2: Generate $Z \sim \mathcal{N}(0,1)$ of size $N_{\text{paths}} \times N_{\text{steps}}$
3: increments $\leftarrow (r - \frac{1}{2}\sigma^2)\Delta t + \sigma\sqrt{\Delta t}\, Z$
4: Initialize $S \in \mathbb{R}^{N_{\text{paths}} \times (N_{\text{steps}}+1)}$
5: $S(:,1) \leftarrow S_0$ ▷ column $j = 1$ corresponds to $t_0 = 0$
6: **for** $j = 1$ to N_{steps} **do**
7: $\quad S(:, j+1) \leftarrow S(:, j) \cdot \exp(\texttt{increments}(:, j))$
8: **end for**

Each row of S corresponds to a single Monte Carlo path; column j corresponds to exercise date $t_{j-1} = (j-1)\,\Delta t$, for $j = 1, \ldots, N_{\text{steps}} + 1$.

5.3 Antithetic Variates

In Monte Carlo simulations the accuracy of an estimator improves as the number of simulated paths increases. However, the convergence rate of the standard Monte Carlo estimator is relatively slow since the standard error decreases at rate $\mathcal{O}(1/\sqrt{N_{\text{paths}}})$ where the hidden constant $\sigma = \sqrt{\text{Var}(Y)}$ depends on the intrinsic variability of the simulated discounted payoff Y. To enhance numerical efficiency, *variance reduction techniques* are employed, aiming to reduce σ directly without increasing N_{paths}. In particular, the method of **antithetic variates** is used, following the treatment in Glasserman [2004].

The theoretical foundation of the antithetic variates method relies on the symmetry of the standard normal distribution. If $Z \sim \mathcal{N}(0,1)$, then $-Z$ follows the same distribution. In the simulation of the GBM for every sample path generated using the sequence of random draws $\{Z_i\}$ it is simultaneously computed also an *antithetic path* constructed using $\{-Z_i\}$.

In the implementation, only $N_{\text{paths}}/2$ independent standard normal draws are generated. Denoting this matrix by `Z_half`, the full increment matrix is then assembled as:

```
Z = [Z_half; -Z_half];
```

This construction ensures that the sample mean of each column of Z is exactly zero. As a consequence, the simulated increments satisfy the risk–neutral drift condition exactly in sample, rather than only in expectation.

Variance reduction mechanism. The variance reduction effect can be understood by considering the payoff values generated by a pair of antithetic paths. Let $Y_1 = f(Z)$ denote the discounted payoff obtained from the original path and $Y_2 = f(-Z)$ the payoff obtained from its antithetic counterpart. The antithetic estimator for the pair is defined as

$$\hat{Y}_{\text{ant}} = \frac{1}{2}(Y_1 + Y_2).$$

Its variance is

$$\text{Var}(\hat{Y}_{\text{ant}}) = \frac{1}{4}\left[\text{Var}(Y_1) + \text{Var}(Y_2) + 2\ \text{Cov}(Y_1, Y_2)\right].$$

Since Y_1 and Y_2 are identically distributed, $\text{Var}(Y_1) = \text{Var}(Y_2)$, which leads to

$$\text{Var}(\hat{Y}_{\text{ant}}) = \frac{1}{2}\left[\text{Var}(Y_1) + \text{Cov}(Y_1, Y_2)\right]. \tag{27}$$

Variance reduction occurs whenever $\text{Cov}(Y_1, Y_2) < 0$. In the case of an American put option, the payoff is a decreasing function of the underlying asset price. Consequently, a path generated by large positive increments tends to produce a relatively high asset price and therefore a low intrinsic value, whereas the corresponding antithetic path produces a lower asset price and a higher intrinsic value. This opposite behavior tends to induce a negative covariance between Y_1 and Y_2, which reduces the variance of the estimator relative to standard Monte Carlo simulation [Hull, 2021].

Within the Least Squares Monte Carlo algorithm, the use of antithetic variates provides two practical advantages:

- the symmetry of the constructed increments ensures that the simulated paths reproduce the risk–neutral drift more accurately in finite samples,
- the reduction in payoff variance leads to more stable regression inputs during the backward induction procedure.

Since the continuation value is estimated through least–squares regression on simulated path data, lower dispersion in the simulated payoffs contributes to more robust and numerically stable estimates of the continuation function.

5.4 Backward Induction and Regression-Based Continuation Value

The core of the LSM algorithm is the dynamic programming recursion that proceeds **backward** from maturity. At each exercise opportunity, the holder of the option faces a binary decision: exercise immediately and receive the immediate payoff $(K - S_t)^+$, or continue holding the option and receive the continuation value C_t (defined in Section 4). Consequently, the optimal stopping rule is:

$$\text{Exercise at } t \quad \Longleftrightarrow \quad (K - S_t)^+ > C_t.$$

The recursion is initialized at maturity $t = T$ (column $N_{\text{steps}} + 1$ of the path matrix). Since no continuation is possible at maturity, $\mathcal{CF}^{(n)}$ is initialized to the terminal payoff:

$$\mathcal{CF}^{(n)} = (K - S_T^{(n)})^+, \qquad n = 1, \ldots, N_{\text{paths}}. \tag{28}$$

At every stage of the backward loop, $\mathcal{CF}^{(n)}$ holds the realized cash flow obtained by path n under the exercise policy estimated for all dates strictly later than the current step: it equals the immediate payoff realized at the date where path n was marked for early exercise, or, if no such date has yet occurred, the terminal payoff at maturity; in either case discounted back to time $t + 1$. This quantity is path-specific: two paths sharing the same price at a given date may carry different values of $\mathcal{CF}^{(n)}$, since it depends on the entire future trajectory realized along that path.

Recursive backward step. The backward loop runs from $t = N_{\text{steps}}$ down to $t = 2$. At each step t, the algorithm proceeds as follows:

1. Discounts all cashflows back by one time step:

   $$\mathcal{CF} \;\leftarrow\; e^{-r\Delta t}\,\mathcal{CF},$$

   where $\mathcal{CF}$ denotes the vector of cashflows across all paths, so that each entry $\mathcal{CF}^{(n)}$ becomes the discounted realized value carried from $t + 1$, i.e. $e^{-r\Delta t}V(t + 1, S_{t+1}^{(n)})$, serving as the response variable in the OLS regression at time t.

2. Identifies the set of In-The-Money (ITM) paths: $\mathcal{I}_t = \left\{n : S_t^{(n)} < K\right\}$. (see Section 4 for theoretical details).

3. For paths in $\mathcal{I}_t$, the continuation value $\hat{C}_t^{(n)}$ is estimated via OLS regression of $\mathcal{CF}^{(n)}$ onto a set of basis functions evaluated at the current asset prices. The construction of the regression matrix and the choice of basis functions are detailed in the following sections.

4. For each path $n \in \mathcal{I}_t$, the immediate payoff is compared with the estimated continuation value. If early exercise is optimal, $\mathcal{CF}^{(n)}$ is overwritten with the immediate payoff and the exercise time is recorded:

   $$\mathcal{CF}^{(n)} \leftarrow (K - S_t^{(n)})^+, \qquad \tau^{(n)} \leftarrow t.$$

   Otherwise, $\mathcal{CF}^{(n)}$ retains its current value $e^{-r\Delta t}V(t + 1, S_{t+1}^{(n)})$, which will be discounted further at the next backward step.

The full backward procedure, including the computation of the option price, is presented in full implementation detail in Algorithm 3.

5.5 Construction of the Regression Matrix

For the paths in $\mathcal{I}_t$, the normalized state variable $X^{(n)} = S_t^{(n)}/K$ is computed and used to evaluate the M basis functions, forming the regression matrix $\mathbf{\Psi}_t \in \mathbb{R}^{|\mathcal{I}_t| \times M}$:

$$\mathbf{\Psi}_t = \begin{bmatrix} \psi_1(X_t^{(n_1)}) & \psi_2(X_t^{(n_1)}) & \cdots & \psi_M(X_t^{(n_1)}) \\ \psi_1(X_t^{(n_2)}) & \psi_2(X_t^{(n_2)}) & \cdots & \psi_M(X_t^{(n_2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(X_t^{(n_{|\mathcal{I}_t|})}) & \psi_2(X_t^{(n_{|\mathcal{I}_t|})}) & \cdots & \psi_M(X_t^{(n_{|\mathcal{I}_t|})}) \end{bmatrix}.$$

where each row corresponds to a single ITM path and each column represents the evaluation of one basis function.
The normalization by K follows the recommendation of Longstaff and Schwartz [2001]: working with the renormalized variables avoids numerical errors arising from scaling problems, particularly when exponential terms appear in the basis functions such as the weighted Laguerre family. This is purely a numerical convenience, the regression response is normalized consistently as $\mathbf{Y}_t = \mathcal{CF}_t/K$, so that the fitted continuation values are of order unity and the monetary scale is restored by multiplication by K when evaluating (30). The option value is therefore unaffected by this rescaling.

5.5.1 Basis Functions

The MATLAB implementation supports four families of basis functions: polynomial, weighted Laguerre, probabilist's Hermite and Chebyshev, whose theoretical properties and motivations are discussed in detail in Section 4.2.1. The choice of basis functions affects both the approximation quality of the continuation value and the numerical stability of the regression problem. Nevertheless, several empirical studies, including Moreno and Navas [2003], show that for standard American put options, the LSM approach exhibits a certain degree of robustness with respect to the choice and the number of basis functions, within reasonable ranges. A comparison of the price estimates obtained with all four basis families is reported in Section 7.1.4.

Two implementation-specific choices deserve mention. First, MATLAB's built-in `hermiteH` function implements the *physicist's* convention, since the probabilist's convention is adopted here, the polynomials are computed manually via the recurrence (24). Second, the Chebyshev basis requires a fixed linear rescaling of the normalized state variable $X = S_t/K$ into $[-1, 1]$ via Equation (25). Since the Chebyshev polynomials T_j are defined and orthogonal on $[-1, 1]$, these bounds must be held *fixed* throughout the entire algorithm for two distinct but related reasons.

1. **Internal consistency of the backward induction.** If the bounds were recomputed locally at each time step t using only the ITM paths available at that date, the affine map $x \mapsto \tilde{x}$ would differ at every step. As a consequence, the Chebyshev basis functions $T_j(\tilde{x})$ evaluated at consecutive exercise dates would then correspond to different parameterizations of the state space, reducing the consistency and interpretability of the continuation value approximation across exercise dates.

2. **Out-of-sample applicability.** The coefficients $\{\widehat{\boldsymbol{\beta}}_t\}$ estimated during training encode the exercise rule in terms of the rescaled variable $\tilde{x}$. During out-of-sample

validation (Section 5.8), the state variable of the new paths must be rescaled using exactly the same bounds; otherwise the stored coefficients would be evaluated at incorrectly transformed inputs, yielding wrong continuation value estimates and invalidating the comparison.

To satisfy both requirements, the bounds are computed *once* before the backward induction begins, from a preliminary simulation initiated at $S_0 = \min(S_{0,\,\text{vec}}) = 80$. The upper bound $x_{\max} = \max_{n,t}(S_t^{(n)})/K$ is determined by the highest simulated price but plays no active role during the regression, since the OLS step uses only ITM paths for which $S_t < K$ by definition, so $X = S_t/K < 1$ at every exercise date. The lower bound $x_{\min}$, by contrast, directly affects the regression: starting from the lowest $S_0 = 80$ ensures that the preliminary simulation visits the low-price region of the state space, where most ITM paths concentrate and where the regression is actually performed. As an additional safeguard, the implementation clamps the rescaled variable to $[-1, 1]$ after applying (25), replacing any value outside this interval with the nearest endpoint:

$$\tilde{x} \leftarrow \max\big(-1,\, \min(1,\, \tilde{x})\big),$$

This prevents numerical instability arising from the unbounded growth of $|T_j(x)|$ outside $[-1, 1]$ [Abramowitz and Stegun, 1964]. The scalars $x_{\min}$ and $x_{\max}$ are stored in `results_struct` and passed unchanged to `lsm_american_put_outofsample`, ensuring a consistent coordinate system throughout all phases of the analysis.

5.6 Ordinary Least Squares Regression

Given the regression matrix $\mathbf{\Psi}_t \in \mathbb{R}^{|\mathcal{I}_t| \times M}$, the regression is performed using all ITM paths at time t. The response vector is defined as $\mathbf{Y}_t = \mathcal{CF}_t/K \in \mathbb{R}^{|\mathcal{I}_t|}$, consistent with the normalized regressors $X = S_t/K$ introduced in Section 5.5. The OLS coefficient vector $\widehat{\boldsymbol{\beta}}_t$ is obtained by solving:

$$\widehat{\boldsymbol{\beta}}_t \;=\; \arg\min_{\boldsymbol{\beta}} \; \|\mathbf{Y}_t - \mathbf{\Psi}_t\,\boldsymbol{\beta}\|_2^2\,, \tag{29}$$

whose closed-form solution is

$$\widehat{\boldsymbol{\beta}}_t = (\mathbf{\Psi}_t^\top \mathbf{\Psi}_t)^{-1} \mathbf{\Psi}_t^\top \mathbf{Y}_t$$

provided that $\mathbf{\Psi}_t^\top \mathbf{\Psi}_t$ is invertible, as discussed in Section 4.2.2. In MATLAB, this system is solved efficiently using the backslash operator `beta = Psi \ Y`, which automatically selects an appropriate numerical method (specifically, QR factorization for overdetermined systems) to ensure numerical stability and computational efficiency. This is numerically preferable to explicitly solving the normal equations, since the condition number of $\mathbf{\Psi}_t^\top \mathbf{\Psi}_t$ equals the square of that of $\mathbf{\Psi}_t$ (i.e. $\kappa(\mathbf{\Psi}_t^\top \mathbf{\Psi}_t) = \kappa(\mathbf{\Psi}_t)^2$), amplifying any ill-conditioning present in the regression matrix.
Once the coefficient vector $\widehat{\boldsymbol{\beta}}_t$ is estimated, the fitted continuation value for each ITM path is computed as:

$$\widehat{C}_t^{(n)} = K \cdot \sum_{m=1}^{M} \widehat{\beta}_{t,m}\, \psi_m\Big(S_t^{(n)}/K\Big)\,, \qquad n \in \mathcal{I}_t, \tag{30}$$

The factor K outside the sum restores the monetary scale: since both the regressor and the response variable are normalized by K, the fitted values are of order unity and must be multiplied by K to recover continuation values expressed in the same units as the option price. The coefficient vectors $\{\widehat{\boldsymbol{\beta}}_t\}_{t=2}^{N_{\text{steps}}}$ are stored for subsequent out-of-sample use.

5.7 Exercise Decision and Cashflow Update

Given the estimated continuation values from the regression step, the exercise decision is implemented by comparing the immediate payoff with the estimated continuation value. For each path $n \in \mathcal{I}_t$, the exercise condition is evaluated:

$$(K - S_t^{(n)})^+ > \widehat{C}_t^{(n)} \quad \Longrightarrow \quad \mathcal{CF}^{(n)} \leftarrow (K - S_t^{(n)})^+, \quad \tau^{(n)} \leftarrow t, \tag{31}$$

Otherwise, $\mathcal{CF}^{(n)}$ retains its current value, which is discounted further at the next backward step. This procedure ensures that each path is associated with a single realized cashflow corresponding to the payoff obtained under the estimated optimal stopping strategy.
The exercise time $\tau^{(n)}$ is initialized to $N_{\text{steps}} + 1$ (indicating no early exercise) and is updated only if the option is exercised before maturity.

5.7.1 Computation of the Option Value at Time Zero

After completing the backward induction, each simulated path $n = 1, \ldots, N_{\text{paths}}$ is associated with a realized cashflow $\mathcal{CF}^{(n)}$, representing the payoff obtained by following the estimated optimal exercise strategy. These cashflows, which are already expressed in present value terms through the backward induction procedure, are discounted to $t = 0$ by applying the discount factor $e^{-r\Delta t}$ corresponding to the first time step not covered by the backward induction loop. The **American put option value at time zero** is then estimated as the sample mean:

$$\widehat{V}(0, S_0) = \frac{1}{N_{\text{paths}}} \sum_{n=1}^{N_{\text{paths}}} \mathcal{CF}^{(n)}, \tag{32}$$

which corresponds to the Monte Carlo approximation of $\widehat{V}_{\text{LSM}}(0, S_0)$ discussed in Section 4.2.3. The associated standard error is:

$$\text{SE} = \sqrt{\frac{1}{N_{\text{paths}} \cdot (N_{\text{paths}} - 1)} \sum_{n=1}^{N_{\text{paths}}} \left(\mathcal{CF}^{(n)} - \widehat{V}(0, S_0) \right)^2}, \tag{33}$$

which quantifies the Monte Carlo sampling uncertainty. By the central limit theorem, this yields an approximate 95% confidence interval given by $\widehat{V}(0, S_0) \pm 1.96\,\text{SE}$ and serves as a key diagnostic for assessing the convergence of the simulation.

Algorithm 3 LSM Backward Induction – American Put

Require: Path matrix $S \in \mathbb{R}^{N_{\text{paths}} \times (N_{\text{steps}}+1)}$, strike price K, basis type and order M, range $[x_{\min}, x_{\max}]$

Initialization: $\mathcal{CF}^{(n)} \leftarrow \max(K - S^{(n)}_{N_{\text{steps}}+1}, 0)$ for all n ▷ Terminal payoff at T
$\tau^{(n)} \leftarrow N_{\text{steps}} + 1$ for all n ▷ Default: hold to maturity
for $t = N_{\text{steps}}$ **down to** 2 **do**
  $\mathcal{CF} \leftarrow e^{-r\Delta t} \cdot \mathcal{CF}$ ▷ $\mathcal{CF} \in \mathbb{R}^{N_{\text{paths}}}$
  $\mathcal{I}_t \leftarrow \{n : S^{(n)}_t < K\}$
  if $\mathcal{I}_t \neq \emptyset$ **then**
    $X^{(n)} \leftarrow S^{(n)}_t / K, \quad Y^{(n)} \leftarrow \mathcal{CF}^{(n)} / K \quad$ for $n \in \mathcal{I}_t$ ▷ Normalize by K
    $\Phi \leftarrow$ `basis_function`$(X, M, \text{type}, x_{\min}, x_{\max})$ ▷ Design matrix, $|\mathcal{I}_t| \times M$
    $\hat{\beta}_t \leftarrow \Phi \setminus Y$ ▷ OLS: $\hat{\beta}_t = (\Phi^\top \Phi)^{-1} \Phi^\top Y$
    $\hat{C}^{(n)}_t \leftarrow K \cdot [\Phi\, \hat{\beta}_t]^{(n)} \quad$ for $n \in \mathcal{I}_t$
    for each $n \in \mathcal{I}_t$ **do**
      if $(K - S^{(n)}_t) > \hat{C}^{(n)}_t$ **then** ▷ Early exercise is optimal
        $\mathcal{CF}^{(n)} \leftarrow (K - S^{(n)}_t)^+$
        $\tau^{(n)} \leftarrow t$
      end if
    end for
  end if
end for
$\mathcal{CF} \leftarrow e^{-r\Delta t} \cdot \mathcal{CF}$ ▷ Final discounting to $t = 0$
$V_0 \leftarrow \dfrac{1}{N_{\text{paths}}} \displaystyle\sum_{n=1}^{N_{\text{paths}}} \mathcal{CF}^{(n)}$
$\widehat{\text{SE}} \leftarrow \text{std}(\mathcal{CF}) \,/\, \sqrt{N_{\text{paths}}}$
return V_0, $\widehat{\text{SE}}$, $\{\widehat{\boldsymbol{\beta}}_t\}_{t=2}^{N_{\text{steps}}}$, $\{\widehat{\tau}^{(n)}\}_{n=1}^{N_{\text{paths}}}$

For diagnostic and validation purposes, the implementation stores a set of quantities that characterize the estimated optimal stopping strategy and its numerical properties.

- **Regression coefficients** $\{\widehat{\boldsymbol{\beta}}_t\}_{t=2}^{N_{\text{steps}}}$: these coefficients define the estimated continuation value at each time step and can be reused to evaluate the exercise policy on newly simulated paths, enabling out-of-sample validation.
- **Exercise times** $\widehat{\tau}^{(n)}$: a vector recording the time step at which each path exercises, or $N_{\text{steps}} + 1$ if the option is held to maturity. These are used to reconstruct the estimated exercise boundary and analyze its evolution over time.
- **Estimated continuation values** $\widehat{C}^{(n)}_t$: provide insight into the quality of the regression approximation and are useful for convergence diagnostics and comparison with theoretical benchmarks.

5.8 Out-of-Sample Validation

A critical aspect to consider is overfitting: the learned exercise strategy may be tailored to the specific sample of training paths and fail to generalize. Out-of-sample validation is particularly important in the LSM framework, as the use of the same simulated paths

for both regression and valuation may introduce an upward bias[7] due to overfitting.
To assess robustness, an out-of-sample validation procedure is implemented in the function `lsm_american_put_outofsample` whose complete source code is reported in Appendix A.3. The procedure consists of two stages:

1. **In-sample training**. The LSM algorithm is run on a training set of N_{paths} simulated paths, and the regression coefficients $\{\widehat{\boldsymbol{\beta}}_t\}_{t=2}^{N_{\text{steps}}}$ (together with the Chebyshev rescaling bounds, when Chebyshev basis functions are used) are stored in the regression model structure `regModel`.

2. **Out-of-sample testing**. A new set of N_{paths} paths is simulated using a different random seed to ensure independence between the training and test samples. The backward induction is then performed using the stored coefficients $\widehat{\boldsymbol{\beta}}_t$, without re-estimating the regression. This allows one to test the generalization ability of the estimated exercise rule and, more generally, the accuracy of the continuation value approximation underlying the stopping decision. The option price, standard error and early exercise fraction are recomputed on this independent sample.

If in-sample and out-of-sample prices agree within the expected Monte Carlo error, this provides evidence that the regression captures the underlying continuation value with sufficient accuracy, rather than overfitting noise in the training sample.
This aspect is particularly important when using Chebyshev basis functions. As explained in Section 5.5.1, the Chebyshev basis requires fixed rescaling bounds $x_{\min}$ and $x_{\max}$ that must remain identical across all phases of the analysis. During out-of-sample validation, the bounds stored in `regModel` are therefore retrieved and passed unchanged to `basis_function`, ensuring that the new paths use exactly the same affine transformation as the training paths. Without this consistency, the stored coefficients $\{\widehat{\boldsymbol{\beta}}_t\}$ would be evaluated at incorrectly rescaled inputs, invalidating the comparison between in-sample and out-of-sample prices.

5.9 Convergence Diagnostics

The accuracy of the LSM estimator depends on two key parameters: N_{paths}, which controls the Monte Carlo sampling error and M which determines the flexibility of the approximation of the continuation value. A systematic grid search is performed over:

$$N_{\text{paths}} \in \{10^4,\ 5\times10^4,\ 10^5,\ 2\times10^5\}, \qquad M \in \{2,\ 3,\ 4,\ 5,\ 6\}.$$

For each of the $4\times5 = 20$ combinations, the LSM algorithm is executed and the estimated option price, standard error, and early exercise fraction are recorded. Two convergence criteria are assessed:

[7] An upward bias arises when the same simulated paths are used both to estimate the continuation value function and to evaluate the resulting stopping strategy. Since the continuation value is approximated by fitting a regression model to noisy simulated cashflows, the estimated exercise strategy may partially adapt to sample-specific noise. When the same paths are reused for valuation, the exercise decisions can exploit this noise rather than the true underlying conditional expectation, leading to a systematic overestimation of the option value. The high bias of this pure regression estimator relative to estimators that use an independent set of paths for valuation is documented empirically in [Glasserman, 2004, Sec. 8].

- **Convergence in N_{paths}.** The estimated price should stabilize as N_{paths} increases, while the standard error SE is expected to decay at the rate $\mathcal{O}(1/\sqrt{N_{\text{paths}}})$, consistent with the central limit theorem. Variance reduction techniques, such as antithetic variates, do not alter this convergence rate but reduce the proportionality constant $\sqrt{\text{Var}(Y)}$, lowering the absolute level of SE for any fixed N_{paths}.
- **Convergence in M.** The estimated price should stabilize as M increases, reflecting an improved approximation of the continuation value. However, if M becomes too large relative to the number of ITM paths at a given time step, the regression may overfit, resulting in increased variance and reduced numerical stability.

The results of this convergence analysis are summarised in Table 14 and discussed in detail in Section 7.1.5.In practice, $M = 4$ basis functions and $N_{\text{paths}} = 10^4$ are found to provide a good balance between accuracy and computational cost for the baseline experiment, while $N_{\text{paths}} = 2 \times 10^5$ is used for the sensitivity and convergence diagnostics.

5.10 Comparison with European Put Prices

To quantify the value of the early exercise feature, the Black–Scholes European put price is computed in closed form:

$$V_{\text{Eur}}(S_0) = Ke^{-rT}\,\Phi(-d_2) - S_0\,\Phi(-d_1), \tag{34}$$

$$d_1 = \frac{\ln(S_0/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}, \qquad d_2 = d_1 - \sigma\sqrt{T}. \tag{35}$$

The early exercise value is then defined as:

$$\text{EEV} = \widehat{V}(0, S_0) - V_{\text{Eur}}(S_0), \tag{36}$$

and is non-negative by construction. This follows directly from Property 1 in Section 1.2.2, which states that the American put price is always at least as large as its European counterpart, i.e. $\widehat{V}(0, S_0) \geq V_{\text{Eur}}(S_0)$, since the American holder retains all the rights of the European holder together with the additional right of early exercise.
The EEV tends to increase when the option is deep ITM or when interest rates are high, as early exercise allows the holder to receive the strike price sooner and reinvest it at the higer rate. The EEV is computed for each $S_0 \in \{80, 90, 100, 110, 120\}$ in order to provide a complete comparison across different moneyness levels.

5.11 Computational Framework

The overall implementation integrates the components introduced in the previous sections into a coherent computational framework for pricing and validating American options under the LSM framework. In particular, the methodology combines the backward induction procedure for pricing (Section 5.7), the construction of the regression basis (Section 5.5.1), the out-of-sample validation approach (Section 5.8) and the Black–Scholes benchmark used to quantify the early exercise value (Section 5.10).
These elements are combined into a unified workflow, which proceeds as follows.

1. **Parameter setup and Chebyshev pre-processing.** Model and algorithmic parameters are defined. When Chebyshev basis functions are employed, the rescaling bounds are computed once from a preliminary simulation, initiated at $S_0 =$

$\min(S_{0,\text{vec}})$ corresponding to the most ITM initial price. These bounds are then kept fixed throughout the analysis to ensure consistency across time steps and between in-sample and out-of-sample evaluations.

2. **Pricing.** Given the specified parameters, the LSM algorithm is applied for each initial value S_0 to estimate the American put price. At the same time, the corresponding European benchmark and the early exercise value are computed, providing a direct comparison across different moneyness levels.

3. **Out-of-sample validation.** The estimated exercise strategy is evaluated on an independent set of simulated paths.This step provides a diagnostic tool to detect potential overfitting and to assess whether the continuation value approximation generalizes beyond the training sample.

4. **Convergence analysis.** A grid search over the number of simulated paths and basis functions is performed to analyze the stability of the estimator and guide the choice of appropriate numerical parameters.

Taken together, these steps define a consistent workflow through which pricing, validation, convergence can be analyzed in a unified setting.

Overall, the chapter has translated the theoretical framework of the LSM method into a concrete numerical implementation, highlighting the key modeling choices and their practical implications. This implementation provides the basis for the empirical analysis presented in Section 7, where the accuracy, stability and robustness of the method are examined in detail.

6 Extension to American Basket Options

The Longstaff–Schwartz algorithm developed and implemented in the previous chapters provides an efficient approach for pricing American-style derivatives. A natural extension of this method is the valuation of **basket options**, i.e. derivatives whose payoff depends on a weighted average of multiple underlying assets, widely used in equity index products and portfolio hedging strategies. As originally emphasized by Longstaff and Schwartz [2001], the LSM approach is particularly well-suited to high-dimensional problems where traditional grid-based methods, such as finite-difference or binomial trees, become computationally infeasible due to the *curse of dimensionality*, since the number of grid points grows exponentially with the number of state variables. The LSM algorithm avoids this exponential growth, as it relies on regression rather than on a discretized grid; however, its accuracy may still deteriorate as the dimension increases, since a larger number of assets typically requires a richer regression basis to approximate the continuation value well and a larger number of simulated paths to estimate the resulting coefficients reliably.

This chapter considers an **American basket put option** written on d correlated assets following a Multidimensional Geometric Brownian Motion. The extension of LSM framework requires three main modifications with respect to the single-asset case: the simulation of a correlated multi-dimensional price process, the definition of the basket value as a scalar aggregate derived from the asset prices and the choice of state variables for the continuation value regression. Regarding the last point, two natural alternatives are discussed: a **univariate regression** that uses only the scalar basket value as the regressor and a **multivariate regression** that uses the full vector of individual asset prices.

6.1 Model Setup and Parameters

Consider a financial market with d risky assets whose prices at time t are denoted by $\{S_t^{(1)}, S_t^{(2)}, \dots, S_t^{(d)}\}$. Under the risk-neutral measure $\mathbb{Q}$, the dynamics of the asset price vector $\mathbf{S}_t = (S_t^{(1)}, \dots, S_t^{(d)})^\top$ follow a Multidimensional Geometric Brownian Motion. Adapting the formulation presented in Glasserman [2004], the Stochastic Differential Equation for asset j is given by:

$$\mathrm{d}S_t^{(j)} = rS_t^{(j)}\,\mathrm{d}t + S_t^{(j)} \sum_{k=1}^{d} a_{jk}\,\mathrm{d}W_t^{(k)}, \quad j = 1, \dots, d, \tag{37}$$

where:

- r is the constant risk-free interest rate;
- $\mathbf{W}_t = (W_t^{(1)}, \dots, W_t^{(d)})^\top$ is a d-dimensional standard Brownian motion with independent components under $\mathbb{Q}$;
- $A = (a_{jk})$ is a $d \times d$ lower-triangular matrix that introduces the correlation among the assets.

The covariance matrix Σ is constructed from the $d \times d$ correlation matrix $\rho = (\rho_{jk})$ and the vector of volatilities $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)^\top$ as:

$$\Sigma = \operatorname{diag}(\boldsymbol{\sigma})\, \rho\, \operatorname{diag}(\boldsymbol{\sigma}) = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_d \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1d} \\ \rho_{21} & 1 & \cdots & \rho_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{d1} & \rho_{d2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_d \end{bmatrix}$$

The formulation (37) implies that the instantaneous covariance between log-returns is given by

$$\operatorname{Cov}\left[\mathrm{d}\log S_t^{(i)}, \mathrm{d}\log S_t^{(j)}\right] = (AA^\top)_{ij}\, \mathrm{d}t = \Sigma_{ij}\, \mathrm{d}t.$$

A detailed derivation of this covariance structure is provided in Appendix A.2.

The matrix A is chosen to be the Cholesky factor of Σ, i.e. A is the unique lower-triangular matrix with positive diagonal entries such that $\Sigma = AA^\top$. Since A is computed directly from Σ, it incorporates both volatility and correlation effects. This factorization plays a crucial role in Monte Carlo simulation: if $\mathbf{Z} = (Z_1, \ldots, Z_d)^\top$ is a vector of independent standard normal random variables, then the transformation $A\mathbf{Z}$ produces a vector of correlated normal variables with covariance matrix Σ. Consequently, the Cholesky decomposition provides an efficient and numerically stable way to generate correlated Brownian increments in the simulation of the Multidimensional GBM as it reduces the number of multiplications and additions required at each simulation step Glasserman [2004].

The Cholesky decomposition exists if and only if Σ is positive definite. Although Σ is symmetric by construction, positive definiteness is not guaranteed by the structure $\Sigma = \operatorname{diag}(\boldsymbol{\sigma})\, \rho\, \operatorname{diag}(\boldsymbol{\sigma})$ alone, since a parametrically specified correlation matrix ρ may fail to be positive definite for certain choices of off-diagonal entries. Since $\operatorname{diag}(\boldsymbol{\sigma})$ is invertible, positive definiteness of Σ is equivalent to that of ρ, which must therefore be verified explicitly. For the correlation matrix

$$\rho = \begin{pmatrix} 1 & 0.6 & 0.5 & 0.4 \\ 0.6 & 1 & 0.4 & 0.3 \\ 0.5 & 0.4 & 1 & 0.2 \\ 0.4 & 0.3 & 0.2 & 1 \end{pmatrix}$$

adopted here, all eigenvalues of Σ are strictly positive,

$$\lambda(\Sigma) \approx (0.2091,\ 0.0785,\ 0.0589,\ 0.0264),$$

confirming that the Cholesky factorization is well defined.

The baseline parameter configuration adopted in the implementation is summarised in Table 8.

Parameter	Description	Value
r	Risk-free rate	0.05
T	Maturity (years)	1
K	Strike price	100
N_{steps}	Number of time steps	100
N_{paths}	Number of Monte Carlo paths	50 000
d	Number of assets	4
$S_0^{(j)}$	Initial asset prices (ATM)	100 (all j)
w_j	Basket weights	1/4 (equal)
$\boldsymbol{\sigma}$	Asset volatilities	(0.25, 0.32, 0.36, 0.28)

Table 8: Parameter configuration for the American basket put option.

6.2 Monte Carlo Simulation of the Multidimensional Process

The Stochastic Differential Equation (37) admits a closed-form solution for each asset j, analogous to the single-asset case. On a uniform time grid with step size $\Delta t = T/N_{\text{steps}}$, the exact simulation formula is

$$S_{t+\Delta t}^{(j)} = S_t^{(j)} \exp\left[\left(r - \tfrac{1}{2}\sigma_j^2\right)\Delta t + \sqrt{\Delta t}\sum_{k=1}^{d} a_{jk}\, Z^{(k)}\right], \qquad j = 1, \ldots, d, \tag{38}$$

where $Z = (Z^{(1)}, \ldots, Z^{(d)})^\top \sim \mathcal{N}(0, I_d)$ is a vector of independent standard normal random variables drawn at each time step. In matrix form, defining the vector of correlated Brownian increments $Z_{\text{corr}} = AZ$, one verifies that

$$\mathbb{E}\left[Z_{\text{corr}}\left(Z_{\text{corr}}\right)^\top\right] = A\,\mathbb{E}[ZZ^\top]A^\top = AI_dA^\top = \Sigma,$$

so that the simulated increments reproduce exactly the prescribed covariance structure of the Brownian motion. This construction provides an efficient implementation of correlated Gaussian increments via the Cholesky factorization of Σ, as discussed in the previous paragraph.

The exponential discretization (38) coincides with the exact solution of the Multidimensional GBM evaluated at the grid points. This follows from the fact that the SDE coefficients r, σ_j, a_{jk} are constant over time, so that, as in the one-dimensional case of Section 5.2, both the marginal lognormal distribution of each asset and the covariance structure across assets (derived in Appendix A.2) are reproduced exactly at each grid date. Consequently, the scheme introduces no discretization error in the joint distribution of the asset price vector, unlike Euler–Maruyama schemes.

The simulated asset prices are stored in a three-dimensional array $S \in \mathbb{R}^{N_{\text{paths}} \times d \times (N_{\text{steps}}+1)}$, where the (i, j, t)-th entry $S_t^{(i,j)}$ denotes the price of asset j at time t along Monte Carlo path i. All paths share the common initial condition $S_0^{(i,j)} = S_0^{(j)}$ for all i.

Antithetic variates. To reduce variance without increasing the number of paths, the antithetic variates technique of Section 5.3 is applied. Only $N_{\text{paths}}/2$ independent standard normal draws are generated, stored in a half-matrix $Z_{\text{half}} \in \mathbb{R}^{N_{\text{paths}}/2 \times d \times N_{\text{steps}}}$; the full draw matrix is then assembled as $Z = [Z_{\text{half}}; -Z_{\text{half}}]$. As in the single-asset case, this symmetry reduces the variance of the estimator by inducing negative correlation between paired payoffs.
The full simulation procedure, including the construction of correlated increments and the time evolution of the asset prices, is summarized in Algorithm 4.

Algorithm 4 Simulation of Multidimensional GBM Paths

Require: $S_0 \in \mathbb{R}^d$, r, $\sigma \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$, T, N_{steps}, N_{paths}, d
$\Delta t \leftarrow T/N_{\text{steps}}$
Generate $Z_{\text{half}} \sim \mathcal{N}(0,1)$ of size $\frac{N_{\text{paths}}}{2} \times d \times N_{\text{steps}}$
$Z \leftarrow [Z_{\text{half}}; -Z_{\text{half}}]$ ▷ Antithetic pairs: $N_{\text{paths}} \times d \times N_{\text{steps}}$
Initialize $S \in \mathbb{R}^{N_{\text{paths}} \times d \times (N_{\text{steps}}+1)}$
Set $S_{i,j,1} \leftarrow S_0^{(j)}$ for all $i = 1, \ldots, N_{\text{paths}}$, $j = 1, \ldots, d$ ▷ At $t = 0$ initial price S_0
for $t = 1$ **to** N_{steps} **do**
    Extract $Z_t \in \mathbb{R}^{N_{\text{paths}} \times d}$ from Z at time slice t ▷ Independent standard normals
    Compute $Z_t^{\text{corr}} \leftarrow Z_t A^\top \in \mathbb{R}^{N_{\text{paths}} \times d}$ ▷ Correlated increments with Cov $= \Sigma$
    for $j = 1$ **to** d **do**
        for $i = 1$ **to** N_{paths} **do**
            $S_{i,j,t+1} \leftarrow S_{i,j,t} \cdot \exp\left[\left(r - \frac{1}{2}\sigma_j^2\right)\Delta t + \sqrt{\Delta t}\,(Z_t^{\text{corr}})_{i,j}\right]$
        end for
    end for
end for
Ensure: Asset price array $S \in \mathbb{R}^{N_{\text{paths}} \times d \times (N_{\text{steps}}+1)}$

6.3 The Basket Value

A basket option depends on a weighted average of the d underlying asset prices. The **basket value** at time t is defined as

$$B_t = \sum_{j=1}^{d} w_j\, S_t^{(j)} = \mathbf{w}^\top \mathbf{S}_t, \tag{39}$$

where $\mathbf{w} = (w_1, \ldots, w_d)^\top$ is a vector of non-negative weights satisfying $\sum_{j=1}^{d} w_j = 1$.
In the numerical experiments, equal weights $w_j = 1/d$ are adopted for all j, corresponding to an equally-weighted basket.

The payoff of an American basket put option with strike K and maturity T is

$$\phi(B_t) = (K - B_t)^+ = \max(K - B_t,\, 0).$$

Although the underlying price process $\mathbf{S}_t \in \mathbb{R}^d$ is d-dimensional, the payoff depends only on the scalar B_t. However, B_t is not itself a Markovian state variable: since $B_t = \mathbf{w}^\top \mathbf{S}_t$ is a many-to-one projection of $\mathbf{S}_t$, distinct asset price vectors $\mathbf{S}_t \neq \mathbf{S}'_t$ satisfying $\mathbf{w}^\top \mathbf{S}_t = \mathbf{w}^\top \mathbf{S}'_t$ generally induce different conditional distributions for the future basket value B_{t+s},

$s > 0$, so that a regression based only on B_t may fail to capture state information that is relevant for the optimal stopping decision. This motivates two alternative approaches to estimating the conditional expectation $\mathbb{E}^{\mathbb{Q}}\left[\mathrm{e}^{-r\Delta t} V_{t+\Delta t} \mid \mathcal{F}_t\right]$ depending on the choice of state variable:

- In the **univariate regression**, the continuation value is approximated as a function of the basket B_t alone, exploiting the fact that B_t is the natural scalar state variable for the payoff.
- In the **multivariate regression**, the continuation value is approximated as a function of the full price vector $\mathbf{S}_t = (S_t^{(1)}, \ldots, S_t^{(d)})^\top$, which retains the full information contained in the individual asset dynamics.

This distinction introduces a trade-off: the univariate approach reduces the dimensionality of the regression problem and the associated computational cost, while the multivariate approach can potentially provide a more accurate approximation of the continuation value, at the cost of a higher-dimensional regression.

From an implementation perspective, for each Monte Carlo path i and time step t, the basket value is computed as $B_t^{(i)} = \mathbf{w}^\top \mathbf{S}_t^{(i)}$, where $\mathbf{S}_t^{(i)} = (S_t^{(i,1)}, \ldots, S_t^{(i,d)})^\top$ denotes the vector of asset prices at time t along path i. The resulting matrix $\mathbf{B} \in \mathbb{R}^{N_{\text{paths}} \times (N_{\text{steps}}+1)}$, with entries $B_{i,t}$, condenses the d-dimensional price information into a scalar time series per path and provides the state variable for the payoff computation in both approaches.

At maturity $T = t_{N_{\text{steps}}}$, the holder of the American basket put exercises the option if and only if $B_T^{(i)} < K$. The value of the option at maturity therefore coincides with its intrinsic value, providing the terminal condition for the backward dynamic programming recursion:

$$V_T^{(i)} = \phi\Big(B_T^{(i)}\Big) = \big(K - B_T^{(i)}\big)^+, \qquad i = 1, \ldots, N_{\text{paths}}.$$

The cashflow vector is initialized at maturity T as

$$\mathcal{CF}^{(i)} = \big(K - B_T^{(i)}\big)^+, \qquad i = 1, \ldots, N_{\text{paths}},$$

and is subsequently updated during the backward induction described in the following sections.

6.4 Univariate Regression on the Basket Value

The first approach to the continuation value regression exploits the scalar structure of the payoff: since ϕ depends on $\mathbf{S}_t$ only through the aggregate $B_t = \mathbf{w}^\top \mathbf{S}_t$, one can approximate the conditional expectation

$$C_t\Big(B_t^{(i)}\Big) \;\approx\; \mathbb{E}^{\mathbb{Q}}\Big[e^{-r\Delta t} V_{t+\Delta t} \mid B_t = B_t^{(i)}\Big]$$

as a function of the scalar B_t alone. This **univariate regression** strategy reduces the dimensionality of the regression problem to one, regardless of d, thereby improving numerical stability and reducing computational cost.

Choice of basis functions. Following the practical recommendation of Glasserman [2004] [8], a standard polynomial basis of degree two is adopted

$$\frac{\widehat{C}_t(B_t)}{K} \;\approx\; \hat{\beta}_0 + \hat{\beta}_1\left(\frac{B_t}{K}\right) + \hat{\beta}_2\left(\frac{B_t}{K}\right)^2, \tag{40}$$

where each term is evaluated at the normalized basket value B_t/K. The normalization by the strike K ensures that the regressors are dimensionless and of order one for At-The-Money paths, which improves the conditioning of the design matrix and the numerical stability of the OLS step. As discussed in Section 5.5.1, the quadratic degree provides sufficient flexibility to capture the convex shape of the continuation value near the exercise boundary, while keeping the number of parameters small relative to the number of simulated paths, thus avoiding overfitting.

In-the-money filtering. As in the single-asset case, the regression is restricted to ITM paths, identified by the index set $\mathcal{I}_t = \{i : B_t^{(i)} < K\}$; the justification for this restriction is discussed in Section 4.2.1.

OLS estimation. Let $n_t = |\mathcal{I}_t|$ and define the shorthand $\tilde{B}_t^{(i)} \equiv B_t^{(i)}/K$. The design matrix $\mathbf{\Psi}_t \in \mathbb{R}^{n_t \times 3}$ and the response vector $\mathbf{Y}_t \in \mathbb{R}^{n_t}$ are constructed as

$$\mathbf{\Psi}_t = \begin{pmatrix} 1 & \tilde{B}_t^{(i_1)} & \left(\tilde{B}_t^{(i_1)}\right)^2 \\ \vdots & \vdots & \vdots \\ 1 & \tilde{B}_t^{(i_{n_t})} & \left(\tilde{B}_t^{(i_{n_t})}\right)^2 \end{pmatrix}, \qquad \mathbf{Y}_t = \frac{1}{K}\begin{pmatrix} \mathcal{CF}^{(i_1)} \\ \vdots \\ \mathcal{CF}^{(i_{n_t})} \end{pmatrix}, \tag{41}$$

where the entries of $\mathbf{Y}_t$ contain the cashflows already discounted from $t + \Delta t$ to t, i.e. $\mathcal{CF}_t^{(i)} = e^{-r\Delta t}\,\mathcal{CF}_{t+\Delta t}^{(i)}$, and further normalized by K.
The OLS coefficient vector $\widehat{\boldsymbol{\beta}}_t = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)^\top$ is obtained by solving the least-squares problem

$$\widehat{\boldsymbol{\beta}}_t \;=\; \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^3} \left\|\mathbf{Y}_t - \mathbf{\Psi}_t\,\boldsymbol{\beta}\right\|_2^2 \;=\; \left(\mathbf{\Psi}_t^\top \mathbf{\Psi}_t\right)^{-1}\mathbf{\Psi}_t^\top \mathbf{Y}_t, \tag{42}$$

solved in practice via QR decomposition using MATLAB's backslash operator (`Psi \ Y`), as discussed in Section 5.6. This avoids explicit formation of the normal equations $\mathbf{\Psi}_t^\top \mathbf{\Psi}_t$, whose condition number equals the square of that of $\mathbf{\Psi}_t$ and thus amplifies any ill conditioning present in the design matrix.
Since the regression is performed on the normalized response $\mathbf{Y}_t = \mathcal{CF}/K$ and normalized regressors $\tilde{B}_t = B_t/K$, the estimated coefficient vector $\widehat{\boldsymbol{\beta}}_t$ is dimensionless. The estimated continuation value in the original monetary scale is therefore recovered by multiplying the fitted value $\mathbf{\Psi}_t\widehat{\boldsymbol{\beta}}_t$ by K:

$$\widehat{C}_t^{(i)} \;=\; K\left[\hat{\beta}_0 + \hat{\beta}_1\left(\frac{B_t^{(i)}}{K}\right) + \hat{\beta}_2\left(\frac{B_t^{(i)}}{K}\right)^2\right] \;=\; K\hat{\beta}_0 \;+\; \hat{\beta}_1\,B_t^{(i)} \;+\; \hat{\beta}_2\,\frac{\left(B_t^{(i)}\right)^2}{K}, \qquad i \in \mathcal{I}_t \tag{43}$$

[8] Glasserman [2004] emphasizes that, in practical implementations of the Longstaff–Schwartz method, low-order polynomial basis functions are typically sufficient to obtain accurate estimates of the continuation value. Increasing the degree of the polynomial may improve the approximation, but often leads to overfitting and numerical instability, especially when the number of simulated paths is limited. Consequently, a small set of basis functions provides a good balance between flexibility and robustness.

where each term carries the correct unit of price, thereby restoring the original monetary scale.

Exercise decision and cashflow update. For each path $i \in \mathcal{I}_t$, the option holder compares the immediate payoff with the estimated continuation value. Early exercise is optimal if and only if

$$\phi\Big(B_t^{(i)}\Big) > \widehat{C}_t^{(i)} \implies \mathcal{CF}^{(i)} \leftarrow \phi\Big(B_t^{(i)}\Big).$$

If the condition is not met, the cashflow is left unchanged and will be discounted further in the next backward step, so that the algorithm selects the first time at which early exercise is optimal along each simulated path. The complete procedure is summarised in Algorithm 5.

Algorithm 5 LSM Backward Induction – Univariate Basket Regression

Require: Basket matrix $B \in \mathbb{R}^{N_{\text{paths}} \times (N_{\text{steps}}+1)}$, asset price array S, K, r, Δt, N_{steps}

$\mathcal{CF}^{(i)} \leftarrow \phi\Big(B_T^{(i)}\Big)$ for all i
for $t = N_{\text{steps}}$ **down to** 2 **do**
  $\mathcal{CF}^{(i)} \leftarrow e^{-r\Delta t}\,\mathcal{CF}^{(i)}$ for all i ▷ Discount one step to time t
  $\mathcal{I}_t \leftarrow \{i : B_t^{(i)} < K\}$ ▷ ITM paths
  if $\mathcal{I}_t \neq \emptyset$ **then**
    $\tilde{X} \leftarrow B_t^{(\mathcal{I}_t)}/K;\ Y \leftarrow \mathcal{CF}^{(\mathcal{I}_t)}/K$
    $\boldsymbol{\Psi} \leftarrow \big[\mathbf{1},\ \tilde{X},\ \tilde{X}^2\big]$ ▷ $n_t \times 3$ design matrix
    $\widehat{\boldsymbol{\beta}}_t \leftarrow \boldsymbol{\Psi} \setminus Y$ ▷ QR-based OLS
    $\widehat{C}^{(i)} \leftarrow K\left(\boldsymbol{\Psi}\,\widehat{\boldsymbol{\beta}}_t\right)^{(i)}$ for all $i \in \mathcal{I}_t$
    for each $i \in \mathcal{I}_t$ **do**
      if $\phi(B_t^{(i)}) > \widehat{C}^{(i)}$ **then**
        $\mathcal{CF}^{(i)} \leftarrow \phi(B_t^{(i)})$ ▷ Early exercise
      end if
    end for
  end if
end for
$\mathcal{CF}^{(i)} \leftarrow e^{-r\Delta t}\,\mathcal{CF}^{(i)}$ for all i ▷ Final discount to $t = 0$
$\hat{V}_0 \leftarrow \frac{1}{N_{\text{paths}}} \sum_{i=1}^{N_{\text{paths}}} \mathcal{CF}^{(i)}$
$\widehat{SE} \leftarrow \frac{\text{std}(\mathcal{CF})}{\sqrt{N_{\text{paths}}}}$

The potential limitation of this dimensionality reduction is that B_t may not fully capture the information in $\mathbf{S}_t$ relevant for the continuation value, since two portfolios with the same basket value but different asset compositions can have different future dynamics. This motivates the multivariate extension discussed in the following section.

6.5 Multivariate Regression on the Full Asset Price Vector

The second approach replaces the scalar basket value with the full vector of individual asset prices as the state variable in the continuation value regression. This **multivariate regression** strategy retains the complete information available at each exercise date,

allowing for a more flexible approximation of the continuation value, at the cost of a higher-dimensional regression problem.

Choice of basis functions. At each exercise date t, the continuation value for ITM paths is approximated using a linear-quadratic basis without cross terms:

$$\frac{\widehat{C}_t(\mathbf{S}_t)}{K} \;\approx\; \hat{\beta}_0 + \sum_{j=1}^{d} \hat{\beta}_j \left(\frac{S_t^{(j)}}{K}\right) + \sum_{j=1}^{d} \hat{\beta}_{d+j} \left(\frac{S_t^{(j)}}{K}\right)^2 . \tag{44}$$

The omission of cross terms $S_t^{(j)} S_t^{(k)}$, $j \neq k$, is a deliberate modelling choice, motivated by the need to control the dimensionality of the regression problem and to avoid overfitting, in line with the general guidelines of Glasserman [2004]. A full second-order polynomial basis in d variables would include, in addition to the $2d+1$ terms already present, all $\binom{d}{2}$ pairwise products, bringing the total number of regressors to

$$1 + 2d + \binom{d}{2} \;=\; 1 + 2d + \frac{d(d-1)}{2} \;=\; \mathcal{O}(d^2).$$

For $d = 4$, this would yield 15 regressors instead of the 9 used here.
As discussed in Glasserman [2004], increasing the number of basis functions beyond what is necessary may deteriorate the quality of the approximation due to overfitting, particularly when the number of available ITM paths is limited. This trade-off is further formalized by Glasserman and Yu [2004], who show that the number of Monte Carlo paths required for reliable convergence of the LSM estimator grows rapidly with the number of the basis functions. Since $n_t = |\mathcal{I}_t|$ varies across exercise dates and may become small near maturity or for deep Out-Of-The-Money initial conditions, keeping the number of regressors low is essential to ensure numerical stability of the regression step.
The linear-quadratic basis without cross terms therefore provides a parsimonious specification, with $2d + 1$ regressors growing only linearly in the dimension d, while retaining sufficient flexibility to capture the main features of the continuation value.

In-the-money filtering and OLS estimation. The ITM index set is defined as in the univariate case, $\mathcal{I}_t = \{i : B_t^{(i)} < K\}$, where the basket value $B_t^{(i)}$ serves as the scalar criterion for ITM filtering in both approaches, since the payoff depends on B_t alone. Let $n_t = |\mathcal{I}_t|$ and denote by $X_t \in \mathbb{R}^{n_t \times d}$ the matrix of normalized individual asset prices for ITM paths, with (i, j)-th entry $\tilde{S}_t^{(i,j)} \equiv S_t^{(i,j)}/K$. The design matrix is then

$$\mathbf{\Psi}_t \;=\; \left[\, \mathbf{1}_{n_t},\;\; X_t,\;\; X_t^2 \,\right] \;\in\; \mathbb{R}^{n_t \times (2d+1)}, \tag{45}$$

where X_t^2 is the element-wise square of X_t. The response vector $\mathbf{Y}_t \in \mathbb{R}^{n_t}$ is defined identically to the univariate case (41). The OLS coefficient vector $\widehat{\boldsymbol{\beta}}_t \in \mathbb{R}^{2d+1}$ is obtained by solving

$$\widehat{\boldsymbol{\beta}}_t \;=\; \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{2d+1}} \left\| \mathbf{Y}_t - \mathbf{\Psi}_t\, \boldsymbol{\beta} \right\|_2^2, \tag{46}$$

again solved via QR decomposition using MATLAB's backslash operator. The fitted value $\mathbf{\Psi}_t \widehat{\boldsymbol{\beta}}_t$ approximates $\mathcal{CF}/K$, since both the response and the regressors are normalized by K. Multiplying back by K yields the continuation value estimate in the original monetary

scale, which in compact form reads

$$\widehat{C}_t^{(i)} \;=\; K\left[\hat{\beta}_0 + \sum_{j=1}^{d} \hat{\beta}_j\, \tilde{S}_t^{(i,j)} + \sum_{j=1}^{d} \hat{\beta}_{d+j}\left(\tilde{S}_t^{(i,j)}\right)^2\right], \qquad i \in \mathcal{I}_t, \tag{47}$$

or equivalently, substituting $\tilde{S}_t^{(i,j)} = S_t^{(i,j)}/K$ and distributing the factor K,

$$\widehat{C}_t^{(i)} \;=\; K\hat{\beta}_0 \;+\; \sum_{j=1}^{d} \hat{\beta}_j\, S_t^{(i,j)} \;+\; \frac{1}{K}\sum_{j=1}^{d} \hat{\beta}_{d+j}\big(S_t^{(i,j)}\big)^2, \qquad i \in \mathcal{I}_t, \tag{48}$$

The exercise decision and cashflow update follow the same rule as in the univariate case, with $B_t^{(i)}$ as the payoff argument.

The complete backward induction procedure follows exactly Algorithm 5, with the only difference that at steps 6–7 the design matrix is constructed using the full asset price vector $X_t \in \mathbb{R}^{n_t \times d}$ instead of the scalar basket value, yielding a $(2d+1)$-column design matrix $\mathbf{\Psi} = [\mathbf{1}, X_t, X_t^2]$ instead of a 3-column matrix. The implementation differences of the two regression approaches are summarized in Table 9.

Aspect	**Univariate**	**Multivariate**
State variable	B_t (scalar)	$\mathbf{S}_t$ (vector)
Normalized input	$\tilde{X} = B_t^{(\mathcal{I}_t)}/K$	$X_t = S^{(\mathcal{I}_t,:,t)}/K$
Design matrix	$[1, \tilde{X}, \tilde{X}^2]$	$[1, X_t, X_t^2]$
Number of regressors	3	$2d+1$

Table 9: Comparison of univariate and multivariate regression approaches.

The theoretical advantage is that the full price vector $\mathbf{S}_t$ provides strictly more information than the scalar B_t for approximating the continuation value, since B_t is a deterministic function of $\mathbf{S}_t$.
The numerical comparison between the two approaches, reported in Section 7.2, quantifies the practical impact of this additional information on the estimated option price.

6.6 Computation of the Option Value at Time Zero

After completing the backward induction, each simulated path $i = 1, \ldots, N_{\text{paths}}$ carries a realized cashflow $\mathcal{CF}^{(i)}$, representing the discounted payoff obtained by following the estimated optimal exercise strategy on the basket process $\{B_t^{(i)}\}$. Unlike the single-asset case, the cashflows now reflect stopping decisions made on a scalar aggregate $B_t = \mathbf{w}^\top \mathbf{S}_t$ of a d-dimensional price process and their distribution depends on both the individual asset volatilities and the inter-asset correlation structure encoded in Σ.

After the final discounting step to $t = 0$, the **American basket put price** is estimated as the sample mean

$$\hat{V}_0 \;=\; \frac{1}{N_{\text{paths}}} \sum_{i=1}^{N_{\text{paths}}} \mathcal{CF}^{(i)}, \tag{49}$$

which constitutes the Monte Carlo approximation of the true option value

$$V_0 = \sup_{\tau \in \mathcal{T}} \mathbb{E}^{\mathbb{Q}}\big[e^{-r\tau}\,\phi(B_\tau)\big]$$

where $\mathcal{T}$ denotes the set of stopping times taking values in $[0, T]$. As established by Longstaff and Schwartz [2001] and discussed in Section 4.2.3, this estimator provides a *lower bound* for the true price, since the regression-based exercise rule is suboptimal with respect to the true optimal stopping boundary.

The associated Monte Carlo standard error is

$$\widehat{SE} \;=\; \sqrt{\frac{1}{N_{\text{paths}}\,(N_{\text{paths}}-1)} \sum_{i=1}^{N_{\text{paths}}} \big(\mathcal{CF}^{(i)} - \hat{V}_0\big)^2},$$

which quantifies the statistical uncertainty arising from the finite number of simulated paths. By the central limit theorem, $\widehat{SE}$ yields an approximate 95% confidence interval $\hat{V}_0 \pm 1.96\,\widehat{SE}$ and serves as a key diagnostic for assessing simulation convergence [Glasserman, 2004]. It is important to note that $\widehat{SE}$ captures only the sampling error due to the finite number of Monte Carlo paths; it does not account for the *approximation error* introduced by the regression-based exercise boundary, which constitutes a separate and generally unquantifiable source of bias [Longstaff and Schwartz, 2001, Clément et al., 2002].

The estimator (49) takes the same form for both the univariate and multivariate regression approaches. The two methods differ only in how the continuation value is approximated during backward induction and therefore in the realized cashflows $\mathcal{CF}^{(i)}$. Any difference in the final price estimates $\hat{V}_0$ across the two approaches reflects the impact of the choice of state variable on the estimated exercise strategy and constitutes the object of the numerical comparison in Section 7.2.

6.7 Comparison of the Two Regression Approaches

The univariate and multivariate regression strategies represent different trade-offs between the richness of the state representation used in the regression and the associated computational cost. From a theoretical standpoint, the continuation value of the American basket put is defined as

$$C_t(\mathbf{S}_t) = \mathbb{E}^{\mathbb{Q}}\big[e^{-r\Delta t}\,V_{t+1}(\mathbf{S}_{t+1}) \,\big|\, \mathbf{S}_t\big]$$

and in principle depends on the full state vector $\mathbf{S}_t = (S_t^{(1)}, \ldots, S_t^{(d)})^\top$, making the multivariate regression the more general of the two approaches. However, the payoff depends on $\mathbf{S}_t$ only through the basket value $B_t = \mathbf{w}^\top \mathbf{S}_t$ and the terminal condition $V_T = \phi(B_T)$ is therefore a function of B_T alone. By backward induction, the continuation value and the option value at each intermediate date are often well approximated as function of B_t, which provides the theoretical justification for the univariate regression as a well-motivated and often sufficient approximation [Glasserman, 2004].
From a computational standpoint, the univariate approach requires solving a 3×3 least-squares system at each time step regardless of d, whereas the multivariate approach solves

a $(2d+1) \times (2d+1)$ system. For $d = 4$, this corresponds to nine regressors, which remains manageable, but the risk of overfitting increases whenever the number of ITM paths is limited relative to the number of parameters, as discussed in Section 6.5.

The numerical comparison is performed by running both algorithms on the same set of simulated paths, controlled by fixing the random seed, and computing the difference $\Delta\hat{V} = \hat{V}_0^{\text{multi}} - \hat{V}_0^{\text{uni}}$. If $|\Delta\hat{V}|$ is small relative to the price level and this pattern is robust across different parameter configurations, this is consistent with the basket value constituting a sufficient state variable for the continuation value regression, suggesting that the univariate approach is adequate for practical purposes.

Overall, this chapter has extended the LSM framework to the multi-asset setting, introducing the simulation of correlated asset dynamics via Cholesky factorization and examining two regression strategies that differ in the choice of state variable for the continuation value approximation. The numerical results comparing the univariate and multivariate approaches are presented in Section 7.2.

7 Results

This chapter presents the numerical results obtained from the two implementations of the LSM algorithm described in Chapter 4: a single-asset American put option on a simulated GBM process (Section 7.1) and a multi-asset basket option under synthetic parameter configurations (Section 7.2). The empirical application to real market data is developed separately in Chapter 8.
Throughout this chapter, the Monte Carlo standard error is reported as

$$SE = \frac{\hat{\sigma}}{\sqrt{N_{\text{paths}}}}, \tag{50}$$

where $\hat{\sigma}$ is the sample standard deviation of the discounted cash flows across paths (see Section 3.1 for the general definition); confidence intervals are at the 95% level unless otherwise stated.

7.1 Single-Asset American Put

The baseline experiment prices an American put option on a single underlying asset simulated under the exact GBM discretization of Section 5.2, with the parameters configuration reported in Table 10.

Table 10: Baseline parameters for the single-asset LSM experiment.

Parameter	**Description**	**Value**
K	Strike price	100
r	Risk-free rate (annual)	0.05
σ	Volatility (annual)	0.20
T	Maturity (years)	1
N_{steps}	Time steps (exercise dates)	100
N_{paths}	Monte Carlo paths	10,000
M	Number of basis functions	4
Basis	Polynomial family	Polynomial
Antithetic variates	Variance-reduction technique	Enabled

Five initial stock prices are considered: $S_0 \in \{80, 90, 100, 110, 120\}$, covering the full range of moneyness. For a put option with strike $K = 100$: $S_0 \in \{80, 90\}$ are In-The-Money (ITM), $S_0 = 100$ is At-The-Money (ATM) and $S_0 \in \{110, 120\}$ are Out-Of-The-Money (OTM). This range is deliberately wide to assess the algorithm's behaviour in qualitatively different regimes, since the structure of the optimal stopping problem differs substantially across moneyness levels.

7.1.1 Baseline Prices, Early Exercise Premium and Lower Bound Verification

Table 11 reports the LSM-estimated American put price $\hat{P}_A$, the closed-form Black–Scholes European put price P_E^{BS} (Section 5.10), the Early Exercise Value EEV $= \hat{P}_A - P_E^{\text{BS}}$,

the relative early exercise premium EEV/$\hat{P}_A$ and the early exercise fraction, defined as the proportion of all simulated paths that are exercised strictly before maturity.

Table 11: Baseline LSM results across moneyness levels from deep ITM to OTM

S_0	$\hat{P}_A$	P_E^{BS}	EEV	EEV/$\hat{P}_A$	Early Ex. Frac.
80	19.9490	16.9820	2.9670	14.9%	1.0000
90	11.4800	10.2140	1.2662	11.0%	0.6741
100	6.1268	5.5735	0.5533	9.0%	0.4347
110	3.0085	2.7859	0.2226	7.4%	0.2241
120	1.4034	1.2920	0.1114	7.9%	0.0974

Lower bound verification. A necessary condition for any valid American option pricer is that the American price dominates the corresponding European price,

$$P_A(S_0) \geq P_E(S_0) \quad \forall\, S_0,$$

since the American contract grants all the rights of the European holder plus the additional right to exercise early (Property 1, Section 1.2.2). Table 11 confirms that this inequality holds strictly for all five initial prices considered.

Moneyness and the Early Exercise Value. The EEV decreases monotonically from 2.967 at $S_0 = 80$ to 0.111 at $S_0 = 120$, indicating that the value of the early exercise feature becomes progressively smaller as the option moves from ITM to OTM. The relative premium EEV/$\hat{P}_A$ measures what fraction of the total American option price is attributable to the early exercise right. This proportion decreases from 14.9% at $S_0 = 80$ to 7.4% at $S_0 = 110$, with a marginal increase to 7.9% at $S_0 = 120$ that does not alter the overall decreasing trend. The fraction of paths exercising before maturity decreases correspondingly, from 100% at $S_0 = 80$ to 9.7% at $S_0 = 120$: early exercise becomes both less valuable and less frequent as moneyness decreases.

This pattern reflects the structure of the optimal stopping problem. When the option is deep ITM, the holder already has a substantial positive payoff available and immediate exercise allows the strike price to be received and reinvested at the risk-free rate, while waiting risks an increase in the stock price that would reduce the payoff. Consequently, early exercise is frequently triggered and represents a large fraction of the total option value. As the option moves towards OTM, its value increasingly reflects the possibility of future favourable downward price movements rather than current intrinsic value and the time value of waiting dominates the exercise decision. At $S_0 = 100$, where approximately 43.5% of paths exercise early, the two quantities are of comparable magnitude and the exercise decision is most sensitive to the accuracy of the estimated continuation value $\hat{C}(t, S_t)$: small errors in the regression approximation can shift paths across the exercise boundary, making the ATM case the most sensitive regime for the LSM algorithm.

7.1.2 Single-Path Illustration of the LSM Decision

To provide concrete intuition for the LSM decision mechanism, Figure 6 displays the full evolution of a selected sample path ($S_0 = 90$, path index 9). The upper panel shows the stock price trajectory together with the strike level $K = 100$; the lower panel overlays the immediate payoff $\max(K - S_t, 0)$ (blue) and the estimated continuation value $\hat{C}(t, S_t)$ (red) at each time step.

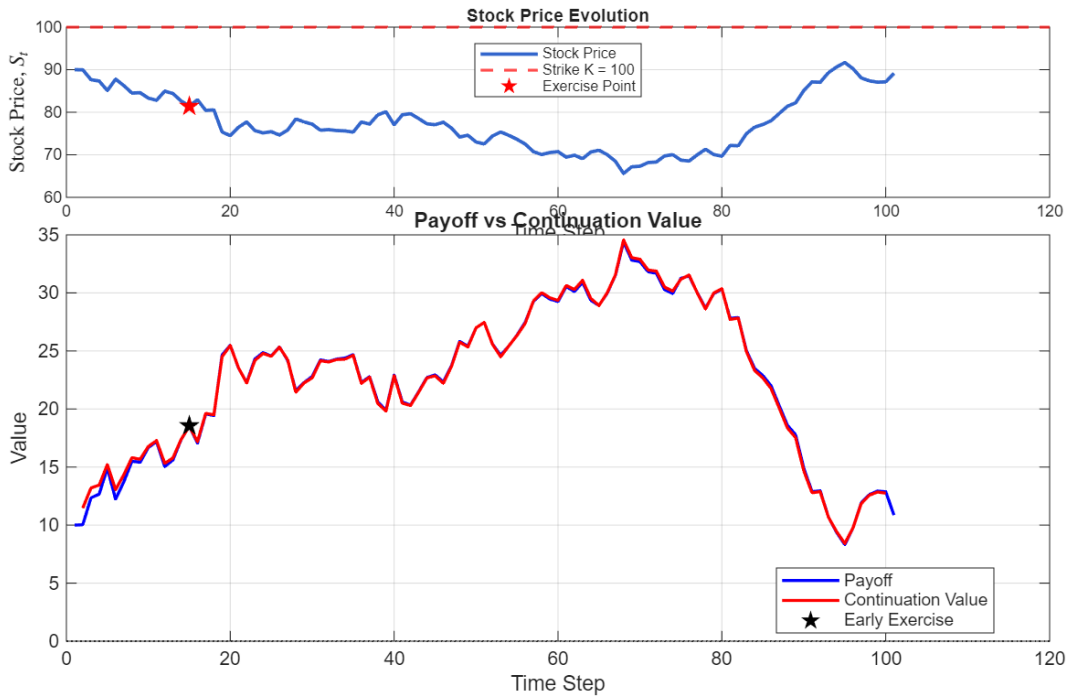


Figure 6: Single-path illustration of the LSM exercise decision ($S_0 = 90$, path index 9). *Upper panel*: stock price trajectory and strike $K = 100$ (dashed red); the red star marks the early exercise point at step 15. *Lower panel*: immediate payoff (blue) and estimated continuation value $\hat{C}(t, S_t)$ (red); curves are shown for all time steps, including those after the exercise point, since both quantities are recorded along the full simulated path. The black star marks the crossing at step 15 where the payoff first exceeds the continuation value.

Step-by-step decision analysis. During the first 14 steps, the continuation value consistently exceeds the immediate payoff, making early exercise suboptimal. For instance, at step $t = 5$ the stock has fallen to $S_5 = 85.10$, giving a payoff of 14.90, but the estimated continuation value is $\hat{C}(5, S_5) = 15.21$: since the expected discounted future payoff exceeds what can be obtained by stopping now, the algorithm waits. This pattern repeats for steps 6–14, with the continuation value remaining between 0.02 and 0.85 above the payoff. At step $t = 15$, the stock has fallen further to $S_{15} = 81.47$, giving a payoff of 18.53. The estimated continuation value is $\hat{C}(15, S_{15}) = 18.51$, marginally below the payoff. The exercise condition $\max(K - S_t, 0) > \hat{C}(t, S_t)$ is satisfied by a margin of only 0.02 and the option is exercised.

Ex-post suboptimality and the role of expectation. After exercise at step 15, the stock continues to fall, reaching $S_{68} = 65.62$ (payoff = 34.38) at step 68, before recovering to approximately 87 at maturity. In this particular realization, exercising at step 15 for a

payoff of 18.53 was suboptimal ex-post, since waiting until step 68 would have yielded a larger payoff. This illustrates a fundamental feature of the LSM algorithm: the exercise decision at each step is based on the conditional expectation of future cash flows, not on the actual realization of the path. The regression estimate $\hat{C}(t_i, S_{t_i}) = \psi(S_{t_i})^\top \hat{\beta}_i$ approximates $\mathbb{E}^{\mathbb{Q}}[e^{-r\Delta t} V_{t_{i+1}} \mid S_{t_i}]$, which is an average over all possible future paths starting from the current state S_{t_i}. At step 15, this estimate was 18.51, marginally below the immediate payoff of 18.53, indicating that the path lies very close to the estimated exercise boundary. Consequently, a single realized path provides only one outcome of the stopping policy and does not determine its overall quality. The objective of the algorithm is to approximate the optimal stopping rule, defined as the one maximizing the expected discounted payoff over all admissible stopping times under the risk-neutral measure, not to select the ex-post optimal exercise time for each individual realization.

7.1.3 Out-of-Sample Validation

A potential concern with any regression-based pricer is overfitting: the exercise rule estimated on a given set of simulated paths might be tailored too closely to sample-specific noise, yielding an in-sample price that does not generalize to independent paths. As discussed in Section 5.8, this effect arises because the same paths are used both to estimate the continuation value function and to evaluate the resulting stopping rule, so that exercise decisions may reflect noise in the training sample rather than the true conditional expectation [Glasserman, 2004].

To assess whether this effect is present in practice, the regression coefficients $\{\hat{\boldsymbol{\beta}}_t\}_{t=2}^{N_{\text{steps}}}$ are estimated once on an in-sample set of paths (seed = 123) and then applied, without re-estimation, to an entirely independent out-of-sample set (seed = 999). If the regression captures the true structure of the continuation value rather than sample-specific noise, the two price estimates should agree to within the order of magnitude of the Monte Carlo standard errors, which quantify the irreducible sampling variability of each estimate independently of the exercise rule.

Table 12: Out-of-sample validation. The regression coefficients $\hat{\boldsymbol{\beta}}_t$ are estimated on in-sample paths (seed = 123) and applied to independent out-of-sample paths (seed = 999). $\Delta P = |\hat{P}_A^{\text{in}} - \hat{P}_A^{\text{out}}|$. $N_{\text{paths}} = 10{,}000$, $M = 4$, polynomial basis.

	In-sample (seed 123)			**Out-of-sample (seed 999)**			
S_0	Price	SE	Early %	Price	SE	Early %	ΔP
80	19.9493	0.0161	100.00	19.9496	0.0159	100.00	0.0003
90	11.4804	0.0809	67.41	11.4128	0.0810	67.12	0.0676
100	6.1268	0.0708	43.47	6.0368	0.0706	42.77	0.0900
110	3.0085	0.0556	22.41	3.0358	0.0560	22.56	0.0273
120	1.4034	0.0383	9.74	1.3690	0.0377	9.48	0.0344

In all five cases, the in-sample and out-of-sample prices are in close agreement, with discrepancies small in absolute terms and comparable to the Monte Carlo standard errors of the individual estimates. The largest discrepancy occurs at $S_0 = 100$, where $\Delta P = 0.090$.

Since the two samples are generated with independent random seeds, the two estimators are independent and their standard errors combine as $\sqrt{0.0708^2 + 0.0706^2} \approx 0.100$: the observed discrepancy falls within one combined standard deviation, indicating consistency with sampling variability rather than systematic overfitting. The early exercise fractions differ by less than one percentage point across all moneyness levels, confirming that the stored coefficients reproduce virtually the same stopping boundary on the independent paths.

Two features of the results deserve specific comment. At $S_0 = 80$, the two prices are virtually identical ($\Delta P = 0.0003$) and the early exercise fraction is 100% in both samples. When the option is sufficiently deep ITM, the continuation value is consistently dominated by the immediate payoff across essentially all paths, so the exercise decision becomes insensitive to the specific regression coefficients estimated in-sample. The largest discrepancy occurs at $S_0 = 100$, consistently with the observation in Section 7.1.1 that the ATM case is the most sensitive regime for the LSM algorithm: it is precisely where payoff and continuation value are of comparable magnitude that small differences in the estimated coefficients across independent samples are most likely to shift paths across the exercise boundary.

Overall, the results indicate that with $M = 4$ basis functions the estimated exercise rule generalizes well to independent paths, providing no evidence of material overfitting.

7.1.4 Basis Function Comparison

Four basis function families are compared: monomial (Polynomial), probabilist's Hermite, weighted Laguerre and Chebyshev of the first kind. Their theoretical properties and motivations are discussed in Section 4.2.1; here the focus is on the numerical behaviour of the resulting price estimates. All experiments use $M = 4$, $N_{\text{paths}} = 50{,}000$, antithetic variates and the same random seed as the baseline analysis to ensure that differences across families reflect the choice of basis rather than sampling variability. Results are reported in Table 13.

Table 13: *Upper panel*: LSM American put prices, standard errors and *Lower panel*: early exercise fraction (%).

	$S_0 = 80$		$S_0 = 90$		$S_0 = 100$		$S_0 = 110$		$S_0 = 120$	
Basis	Price	SE	Price	SE	Price	SE	Price	SE	Price	SE
Polynomial	19.9549	0.0092	11.4334	0.0350	6.1051	0.0320	3.0121	0.0245	1.3922	0.0169
Hermite	19.9549	0.0092	11.4334	0.0350	6.1051	0.0320	3.0121	0.0245	1.3922	0.0169
Laguerre	19.9569	0.0092	11.4367	0.0349	6.1081	0.0321	3.0110	0.0245	1.3925	0.0168
Chebyshev	19.9558	0.0092	11.4334	0.0350	6.1051	0.0320	3.0121	0.0245	1.3922	0.0169

Basis	$S_0 = 80$	$S_0 = 90$	$S_0 = 100$	$S_0 = 110$	$S_0 = 120$
Polynomial	99.49	67.60	40.27	21.64	11.38
Hermite	99.49	67.60	40.27	21.64	11.38
Laguerre	99.46	67.48	40.00	21.51	11.34
Chebyshev	99.47	67.60	40.27	21.64	11.38

Agreement among Polynomial, Hermite and Chebyshev. Polynomial and Hermite produce prices that agree to machine precision across all five values of S_0 and the corresponding early exercise fractions agree to two decimal places. Chebyshev agrees with the other two families to machine precision at $S_0 \in \{90, 100, 110, 120\}$, with a marginal difference of 0.0009 at $S_0 = 80$.[9] As discussed in Section 4.2.1, with $M = 4$ all three families span the same approximation space of degree at most 3 in the normalized state variable $X = S_t/K$. For Hermite this follows directly from the recurrence (24), which produces polynomials of degrees 0–3; for Chebyshev it follows from the same argument applied to $\tilde{x}$, combined with the invertibility of the affine map $x \mapsto \tilde{x}$ in equation (25). Since OLS computes the orthogonal projection onto the column space of the design matrix, identical column spaces imply identical fitted continuation values up to numerical precision.
These results confirm the theoretical observation of Section 4.2.1 that the choice among basis families spanning the same approximation space has no effect on the estimated continuation value.

Mild discrepancy for Laguerre. The weighted Laguerre basis produces slightly different prices and early exercise fractions across all five values of S_0. The largest price discrepancy relative to the polynomial basis occurs at $S_0 = 90$, where the difference is 0.0033, corresponding to approximately 0.09 standard errors. The early exercise fractions differ by at most 0.27 percentage points, occurring at $S_0 = 100$. As noted in Section 4.2.1, the weighted Laguerre functions $e^{-x/2}L_j(x)$ are not polynomials and therefore span a different function space from the other three families, which all generate the

[9]This isolated discrepancy is a numerical artifact rather than evidence against the theoretical equivalence of the two bases: the Chebyshev rescaling bounds $[x_{\min}, x_{\max}]$ (Section 4.2.1) are calibrated once on a preliminary sample, and a small number of paths at $S_0 = 80$ subsequently fall marginally outside this range and are clamped, which perturbs the fitted regression coefficients through the higher leverage of boundary points in the polynomial regression.

same polynomial approximation space of degree at most $M - 1$. This structural difference leads to slightly different fitted continuation values in finite samples. Nevertheless, all discrepancies remain well within one standard error, consistent with the robustness findings of Moreno and Navas [2003].

7.1.5 Convergence Diagnostics

The accuracy of the LSM estimator depends on two parameters: N_{paths}, which controls Monte Carlo sampling error and M, which determines the flexibility of the continuation value approximation. The theoretical role of both parameters is discussed in Section 4.2.3; here their joint effect is assessed empirically at $S_0 = 100$ (ATM), the moneyness level where the regression approximation is most consequential, since approximately 43% of paths exercise early and the quality of the continuation value estimate directly affects the exercise boundary.

Monte Carlo standard error scaling. Figure 7 compares the empirical standard error of the LSM estimator ($M = 4$, $S_0 = 100$) with the theoretical Monte Carlo benchmark $c/\sqrt{N}$, normalized to pass through the first observation at $N = 10$k. On a log–log scale, exact $\frac{1}{\sqrt{N}}$ convergence appears as a straight line with slope $-1/2$.

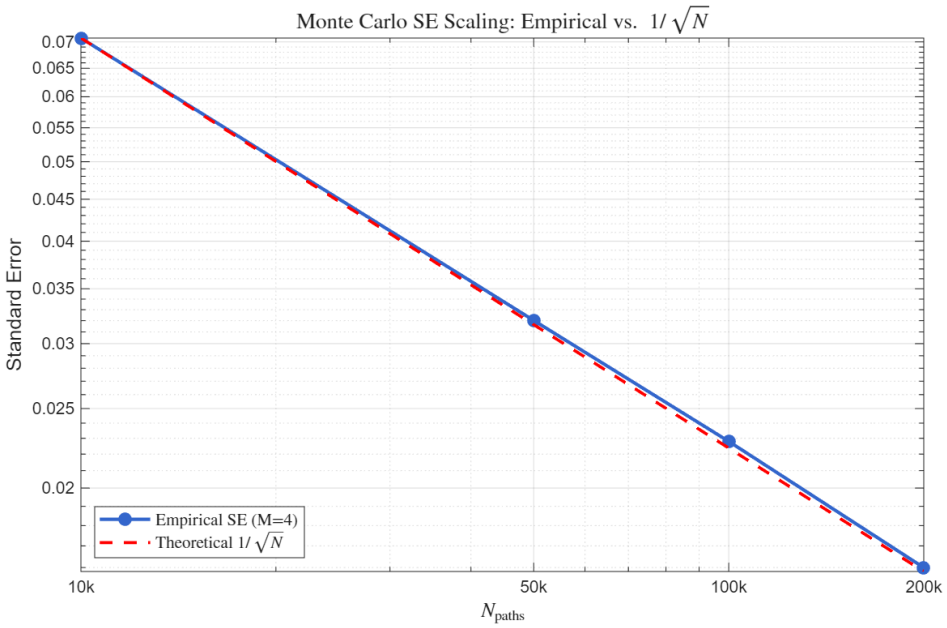


Figure 7: Empirical standard error of the LSM estimator (blue, $M = 4$, $S_0 = 100$) and the theoretical $c/\sqrt{N}$ benchmark (red dashed), normalized to the first observation at $N = 10$k, on a log–log scale.

The empirical and theoretical curves are nearly indistinguishable over the entire range of simulated paths. The estimated slope[10] on the log–log scale is -0.496, very close to

[10]The slope is computed from the first and last observations for $M = 4$ in Table 14 as

$$\frac{\ln(SE_2) - \ln(SE_1)}{\ln(N_2) - \ln(N_1)},$$

yielding a value of -0.496.

the theoretical value of -0.5. Moreover, from Table 14, the SE (Eq. (50)) decreases from 0.0708 at $N = 10$k to 0.0160 at $N = 200$k, consistent with the theoretical prediction $0.0708/\sqrt{20} \approx 0.0158$. This confirms that the LSM estimator preserves the standard Monte Carlo convergence rate despite the regression step used to approximate continuation values. Increasing N therefore reduces the standard error at the expected $N^{-1/2}$ rate, while the choice of basis functions primarily affects approximation bias rather than statistical convergence.

Convergence in N_{paths} and M. Table 14 reports the LSM price, standard error and early exercise fraction for all 20 combinations of $M \in \{2, 3, 4, 5, 6\}$ and $N_{\text{paths}} \in \{10\text{k}, 50\text{k}, 100\text{k}, 200\text{k}\}$, at $S_0 = 100$.

Table 14: Convergence diagnostics: LSM price, standard error, and early exercise fraction for all combinations of M and N_{paths}. $S_0 = 100$, polynomial basis, antithetic variates.

M	N_{paths}	LSM Price	Std. Error	Early Ex. Frac.
2	10,000	6.0037	0.0734	0.4097
2	50,000	6.0131	0.0336	0.2908
2	100,000	6.0004	0.0237	0.3100
2	200,000	5.9920	0.0167	0.3062
3	10,000	6.0919	0.0701	0.4114
3	50,000	6.0963	0.0322	0.3710
3	100,000	6.0771	0.0228	0.3816
3	200,000	6.0693	0.0159	0.3784
4	10,000	6.1268	0.0708	0.4347
4	50,000	6.1051	0.0320	0.4027
4	100,000	6.1000	0.0228	0.4117
4	200,000	6.0917	0.0160	0.4083
5	10,000	6.1211	0.0695	0.4512
5	50,000	6.1079	0.0321	0.4205
5	100,000	6.1031	0.0228	0.4261
5	200,000	6.0935	0.0160	0.4252
6	10,000	6.1539	0.0706	0.4479
6	50,000	6.1170	0.0322	0.4295
6	100,000	6.1029	0.0227	0.4347
6	200,000	6.0990	0.0159	0.4359

Three patterns emerge from Table 14, collectively illustrating the two independent sources of error in the LSM estimator: Monte Carlo sampling noise, which diminishes as $N^{-1/2}$ and approximation bias, which depends on M and does not vanish with additional paths.

First, the standard error decreases monotonically with N_{paths} for every value of M. Since

$SE \propto N^{-1/2}$, moving from $N = 10\text{k}$ to $N = 200\text{k}$ corresponds to a reduction factor $k = 20$, i.e. $\sqrt{20} \approx 4.47$. For $M = 4$, the observed ratio is $0.0708/0.0160 = 4.43$, in close agreement with this prediction. A similar ratio is observed across all values of M.

Second, the price estimate for $M = 2$ is systematically lower than for $M \geq 3$ and this gap persists at large N. At $N = 200\text{k}$, the prices are 5.9920 for $M = 2$ and 6.0693–6.0990 for $M \in \{3, 4, 5, 6\}$, with differences ranging from approximately 0.08 to 0.11, several times larger than the corresponding Monte Carlo standard error. This is consistent with approximation bias: the gap does not vanish as N increases, since it originates from the limited flexibility of the approximation space, as discussed in Section 4.2.3.

Third, increasing M beyond 3 provides negligible additional improvement. At $N = 200\text{k}$, the prices for $M \in \{3, 4, 5, 6\}$ range from 6.0693 to 6.0990, a spread of 0.030, approximately two standard errors. These results confirm the practical recommendation of Longstaff and Schwartz [2001] that a small number of basis functions, typically between 3 and 6, is sufficient for single-asset American options.

Taken together, these patterns illustrate the bias–variance trade-off formalized by Stentoft [2004]: increasing N reduces Monte Carlo sampling error at rate $N^{-1/2}$, while increasing M reduces approximation bias. However, Stentoft [2004] also warns that M must not grow too rapidly relative to N, as too many parameters estimated from too few paths inflates regression variance and risks overfitting. In the present experiments there is no evidence of such behaviour, since the estimated prices remain stable for $M \in \{3, 4, 5, 6\}$ and the standard errors remain broadly unchanged across values of M. The theoretical argument therefore supports the use of a moderate number of basis functions, motivating the choice of $M = 4$ as the baseline specification.

Figure 8 presents the same information as a 5×4 heatmap, making the two error sources simultaneously visible. The right panel reports the **relative deviation**, defined as

$$\text{RelDev}(M, N) = \frac{|\hat{V}(M, N) - V_{\text{ref}}|}{V_{\text{ref}}} \times 100\%,$$

where $V_{\text{ref}} = \hat{V}(M = 6, N = 200\text{k}) = 6.0990$ serves as a proxy for the true American option price, corresponding to the richest basis specification and largest number of simulated paths considered in the experiment.

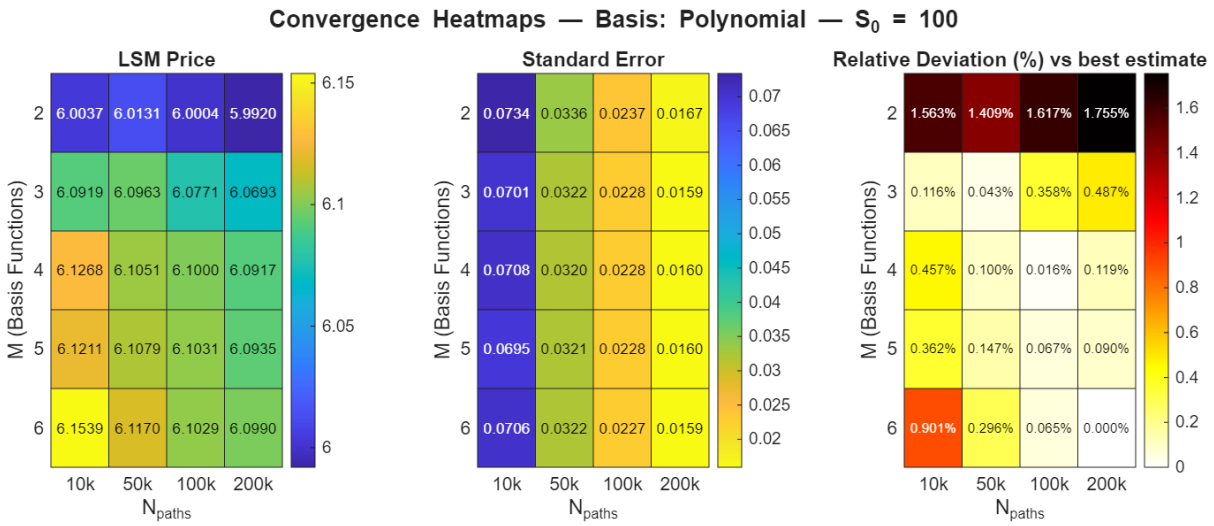
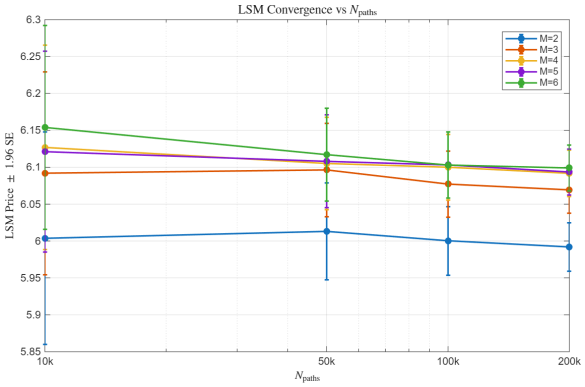


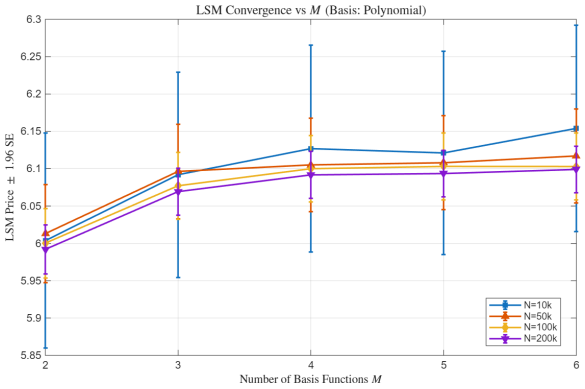
Figure 8: LSM American put price (left), standard error (centre) and relative deviation from $\hat{V}(M = 6, N = 200\text{k})$ (right), over the grid $M \in \{2, 3, 4, 5, 6\}$ and $N_{\text{paths}} \in \{10\text{k}, 50\text{k}, 100\text{k}, 200\text{k}\}$. $S_0 = 100$, polynomial basis, antithetic variates.

The central panel shows that the standard error decreases uniformly as N increases, with little variation across values of M, consistent with the $N^{-1/2}$ scaling established in the previous paragraph indicating that sampling uncertainty is driven primarily by the number of simulated paths. The left panel reveals that for $M \geq 3$ the estimated price is essentially stable across the grid; the colour gradient is driven almost entirely by the $M = 2$ row, which stabilizes at a systematically lower level.
The right panel reveals a pattern not visible in the convergence plots alone. For $M \geq 3$, the relative deviation falls below 0.5% for $N_{\text{paths}} \geq 50\text{k}$ and below 0.1% for $M \geq 4$ at $N_{\text{paths}} = 100\text{k}$. The $M = 2$ row, by contrast, displays deviations ranging from 1.4–1.8% that do not decrease with N_{paths}: the cell ($M = 2$, $N = 200\text{k}$) registers a relative deviation of 1.755%, larger than the corresponding cell at $N = 10\text{k}$ (1.563%). This visual evidence reinforces the interpretation from Table 14 that the remaining error for $M = 2$ is dominated by approximation bias rather than Monte Carlo sampling variability.

Convergence plots. Figures 9a and 9b provide a complementary visual representation of the patterns identified in Table 14.



(a) LSM price ± 1.96 SE vs. N_{paths}, for $M \in \{2, 3, 4, 5, 6\}$.



(b) LSM price ± 1.96 SE vs. M, for $N_{\text{paths}} \in \{10\text{k}, 50\text{k}, 100\text{k}, 200\text{k}\}$.

Figure 9: LSM American put price $\pm 1.96\,\text{SE}$ ($S_0 = 100$, polynomial basis, antithetic variates). *Left*: price vs. N_{paths} for $M \in \{2, 3, 4, 5, 6\}$. *Right*: price vs. M for $N_{\text{paths}} \in \{10\text{k}, 50\text{k}, 100\text{k}, 200\text{k}\}$.

Figure 9a shows that the $M = 2$ curve stabilizes at a level clearly separated from the $M \geq 3$ cluster, with non-overlapping confidence intervals at large N, providing a visual confirmation of the approximation bias identified in the previous paragraph.
Figure 9b highlights the sharp transition from $M = 2$ to $M = 3$, after which the price stabilizes for large N_{paths}; the residual upward trend visible at $N = 10\text{k}$ largely disappears as N_{paths} increases, suggesting that a substantial part of this variation is attributable to Monte Carlo sampling variability rather than to a persistent dependence on M. On the basis of these results, $M = 4$ is adopted as the baseline specification: it lies above the underfitting threshold at $M = 2$ and within the stable region where further increases in M provide no measurable improvement at the path counts considered.

7.1.6 Sensitivity to Time Discretization

Table 15 and Figure 10 report the LSM price and standard error for $N_{\text{steps}} \in \{10, 25, 50, 100, 200, 500\}$ across three maturities $T \in \{1/6, 1/2, 1\}$ years, at $S_0 = K = 100$, $M = 4$, $N_{\text{paths}} = 200{,}000$, with independent random seeds across configurations. As discussed in Section 5.1, increasing N_{steps} reduces the approximation error introduced by replacing the continuous optimal stopping problem with a discrete one; the purpose of this experiment is to verify that the baseline choice $N_{\text{steps}} = 100$ does not introduce a systematic pricing error relative to finer grids.

Table 15: LSM price sensitivity to time discretization. $S_0 = K = 100$, $r = 0.05$, $\sigma = 0.20$, $M = 4$, $N_{\text{paths}} = 200{,}000$, polynomial basis, antithetic variates. The European BS closed-form price is reported as a theoretical lower bound for each maturity.

	$T = 1/6$ yr (BS: 2.8449)		$T = 1/2$ yr (BS: 4.4197)		$T = 1$ yr (BS: 5.5735)	
N_{steps}	Price	SE	Price	SE	Price	SE
10	2.9052	0.0083	4.6130	0.0127	6.0298	0.0164
25	2.9044	0.0081	4.6447	0.0128	6.0725	0.0162
50	2.9064	0.0080	4.6436	0.0125	6.0810	0.0161
100	2.9121	0.0079	4.6495	0.0124	6.0743	0.0160
200	2.9040	0.0079	4.6446	0.0123	6.0887	0.0159
500	2.9070	0.0077	4.6624	0.0122	6.1020	0.0157

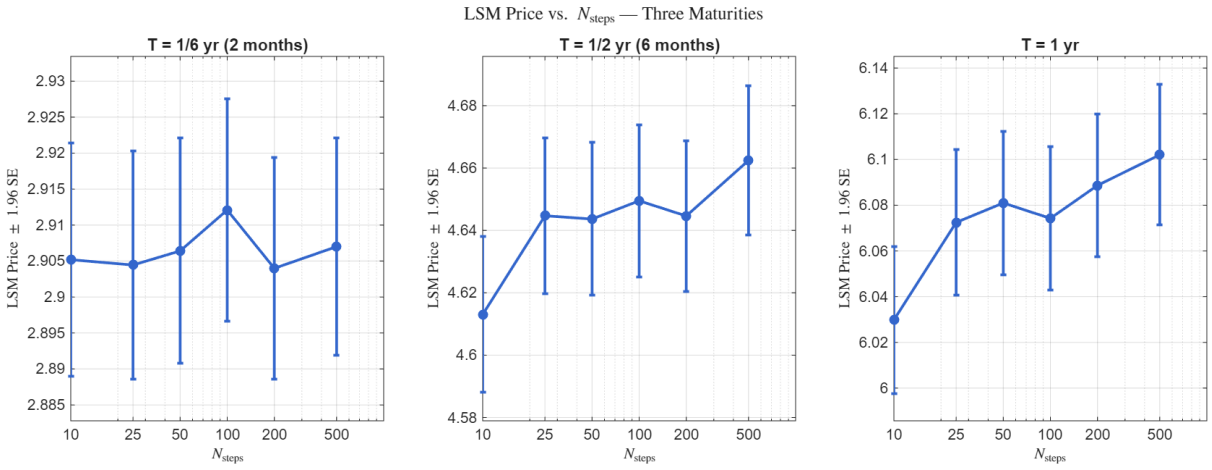


Figure 10: LSM American put price ± 1.96 SE as a function of N_{steps} for $T \in \{1/6, 1/2, 1\}$ years ($S_0 = 100$, $K = 100$, polynomial basis, antithetic variates). European BS lower bounds: 2.8449, 4.4197, 5.5735 respectively.

In all 18 configurations the LSM price lies strictly above the European BS lower bound, confirming that the early exercise feature is captured regardless of grid density or maturity. The price at $N_{\text{steps}} = 10$ lies visibly below the remaining configurations for $T \geq 1/2$ year, a pattern consistent with the reduced flexibility of the exercise policy on a coarse time grid; for $T = 1/6$ year this effect appears negligible, since even 10 exercise dates correspond to opportunities approximately every six days.

For $N_{\text{steps}} \geq 25$, the absolute difference between each price estimate and the baseline $N_{\text{steps}} = 100$ is smaller than the corresponding 95% confidence threshold $1.96\sqrt{SE_1^2 + SE_2^2}$ across all three maturities, so no configuration produces a price statistically distinguishable from the baseline at the 95% level.[11] The total variation across all six grid configurations is small relative to the price level: 0.008 (0.3%) for $T = 1/6$ year, 0.049 (1.1%) for

[11] The largest discrepancy occurs at $T = 1$ year, $N_{\text{steps}} = 500$: $|6.1020 - 6.0743| = 0.0277$, against a 95% threshold of $1.96\sqrt{0.0157^2 + 0.0160^2} \approx 0.0439$. This is the only case exceeding one combined standard error (0.0224), although it remains below the 95% confidence threshold.

$T = 1/2$ year and 0.072 (1.2%) for $T = 1$ year. Within the range considered, increasing N_{steps} beyond 100 produces changes that are small relative to Monte Carlo sampling error, suggesting that any remaining discretization error is negligible for the present application. On this basis, $N_{\text{steps}} = 100$ is adopted as the baseline throughout the analysis.

7.2 American Basket Put Option

This section presents the numerical results for an American basket put option written on $d = 4$ correlated assets, priced via the LSM algorithm described in Chapter 6. Two regression specifications are compared throughout: the **univariate** approach, which approximates the continuation value as a quadratic function of the scalar basket value $B_t = \mathbf{w}^\top \mathbf{S}_t$ and the **multivariate** approach, which uses the full vector of individual asset prices in a linear-quadratic basis with $2d + 1 = 9$ regressors (Section 6.5). The baseline parameter configuration follows Table 8. The number of Monte Carlo paths is increased to $N_{\text{paths}} = 50{,}000$ wrt single-asset case in order to reduce Monte Carlo noise, yielding standard errors on the order of 10^{-2}.

7.2.1 Baseline Prices and Comparison of Regression Approaches

Table 16 reports the LSM-estimated American basket put price and Monte Carlo standard error for both regression approaches at the ATM initial condition $S_0^{(j)} = 100$ for all $j = 1, \ldots, 4$.

Table 16: Baseline American basket put prices for both specifications. Parameters: $K = 100$, $r = 0.05$, $T = 1$, $d = 4$, $\boldsymbol{\sigma} = (0.25, 0.32, 0.36, 0.28)$, $N_{\text{paths}} = 50{,}000$, antithetic variates, seed = 123.

Method	**Price**	**SE**	**No. regressors**
Univariate	6.9778	0.0359	3
Multivariate	6.9369	0.0373	9

The difference between the two estimated prices is small in economic terms: $\Delta\hat{V} = \hat{V}_{\text{multi}} - \hat{V}_{\text{uni}} = -0.0409$, which corresponds to approximately 0.6% of the univariate estimate. The similarity of the two estimates is consistent with the structure of the option payoff. Since the basket put payoff depends on the underlying assets only through the basket value B_t, the univariate specification employs directly the variable that determines moneyness at each exercise date. The close agreement between the two prices suggests that, for the present parameter configuration, B_t captures a substantial share of the information required to approximate the continuation value.

The multivariate specification employs a larger regression basis (9 regressors instead of 3), which increases estimation complexity at a fixed number of simulated paths and may introduce additional coefficient-estimation uncertainty [Glasserman, 2004]. The marginally larger standard error observed for the multivariate specification (0.0373 vs. 0.0359) is consistent with this increase in model complexity, although the difference remains small, and the slightly lower price should not be interpreted as evidence of inferior accuracy.

7.2.2 Sensitivity to moneyness

Table 17 reports prices, standard errors, price differences and early exercise fractions across $S_0 \in \{70, 80, 90, 100, 110, 120, 130\}$, where all four assets are initialized at the same value so that the initial basket equals S_0.

Table 17: Basket put prices and early exercise fractions across moneyness levels. $N_{\text{paths}} =$ 50,000; remaining parameters as in Table 8.

	Univariate		**Multivariate**			**EEF (%)**	
S_0	Price	SE	Price	SE	$\Delta\hat{V}$	Uni	Multi
70	29.9500	0.0072	29.9497	0.0071	−0.0003	100.0	100.0
80	20.0407	0.0239	20.0168	0.0224	−0.0239	92.7	94.0
90	12.1952	0.0397	12.1012	0.0401	−0.0940	61.9	60.3
100	6.9778	0.0359	6.9369	0.0373	−0.0409	38.8	35.0
110	3.7727	0.0284	3.7283	0.0292	−0.0444	22.6	19.5
120	1.9110	0.0203	1.9043	0.0214	−0.0067	12.2	9.3
130	0.9423	0.0148	0.9453	0.0152	+0.0030	6.2	5.4

Option values decrease monotonically from 29.95 at $S_0 = 70$ to 0.94 at $S_0 = 130$, consistent with the standard behaviour of a put option as the underlying moves from deep ITM to deep OTM. The early exercise fraction follows the same declining pattern: at $S_0 = 70$ the estimated early exercise fraction reaches 100%, reflecting the high intrinsic value of the deep ITM option, while at $S_0 = 130$ early exercise occurs on approximately 6% of paths. The two specifications yield broadly similar early exercise fractions across all moneyness levels, with the largest divergence of 3.8 percentage points occurring at $S_0 = 100$.

The price differences $\Delta\hat{V}$ are small throughout the moneyness range. In absolute terms, $|\Delta\hat{V}|$ does not exceed 0.094, attained at $S_0 = 90$; in relative terms, it does not exceed approximately 1.2% of the univariate price, attained at $S_0 = 110$. This is visible in the left panel of Figure 11, where the two price curves and their 95% confidence intervals are essentially indistinguishable across the full range of S_0.

The magnitude of $\Delta\hat{V}$ is not uniform across moneyness levels: it is largest, in absolute terms, in the intermediate region $S_0 \in \{90, 100, 110\}$ ($|\Delta\hat{V}|$ between 0.041 and 0.094) and close to zero at the extremes of the range ($S_0 = 70$ and $S_0 = 130$, where $|\Delta\hat{V}| \leq 0.003$). As noted above, this pattern parallels the variation in the early exercise fraction across moneyness levels: when early exercise occurs on almost all paths or on only a small fraction of them, the estimated continuation value plays a limited role in the exercise decision and the two regression specifications produce nearly identical cashflows. In the intermediate region, where early exercise occurs on a substantial fraction of paths, the exercise decision depends more directly on the estimated continuation value, so that the two specifications are more likely to imply different exercise decisions and consequently different price estimates.

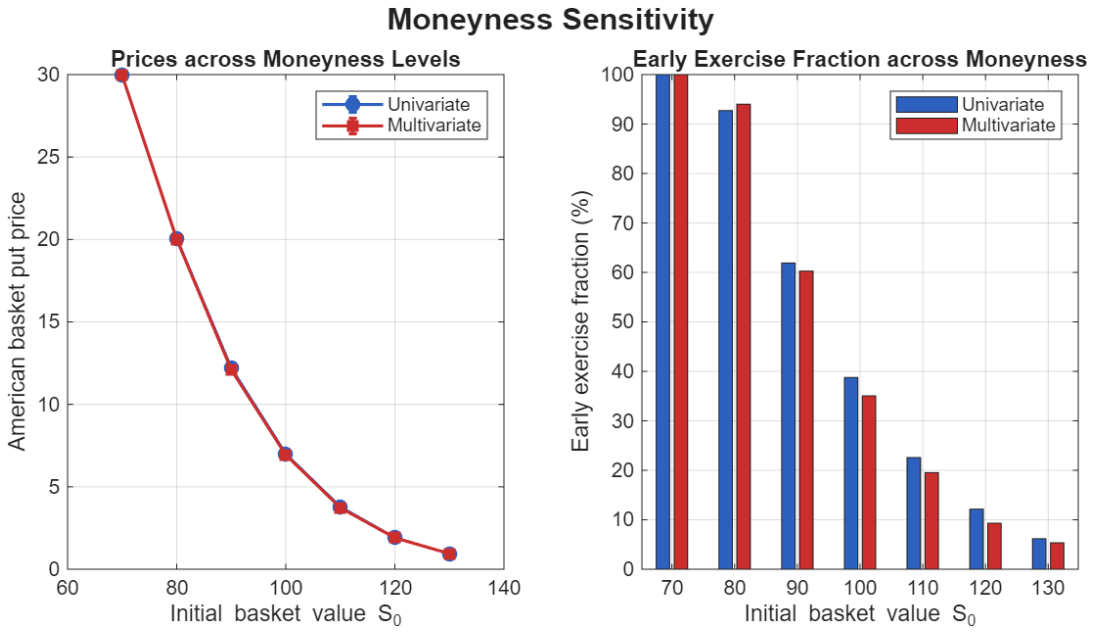


Figure 11: Basket put prices with 95% confidence intervals (*left*) and early exercise fractions (*right*) for the univariate (blue) and multivariate (red) specifications across moneyness levels. $N_{\text{paths}} = 50{,}000$; remaining parameters as in Table 8.

To assess whether the sign of $\Delta\hat{V}$ observed at the baseline seed reflects a systematic feature or a property of that particular Monte Carlo realization, the experiment is repeated under two additional seeds (42 and 999); results are shown in Figure 12. For $S_0 \in \{80, \ldots, 120\}$, $\Delta\hat{V}$ is negative under all three seeds, although its magnitude varies considerably from seed to seed (e.g. at $S_0 = 100$, $\Delta\hat{V}$ ranges from -0.041 to -0.065 across seeds). At the extremes of the range, $S_0 = 70$ and $S_0 = 130$, $\Delta\hat{V}$ is close to zero and its sign is not consistent across seeds. As discussed in Section 7.2.1, neither the sign nor the magnitude of $\Delta\hat{V}$ should be interpreted as evidence of differential accuracy between the two specifications; the robustness check indicates instead that, away from the extremes of the moneyness range, the multivariate price tends to lie slightly below the univariate price, while the exact size of this gap is itself subject to non-negligible Monte Carlo variability.

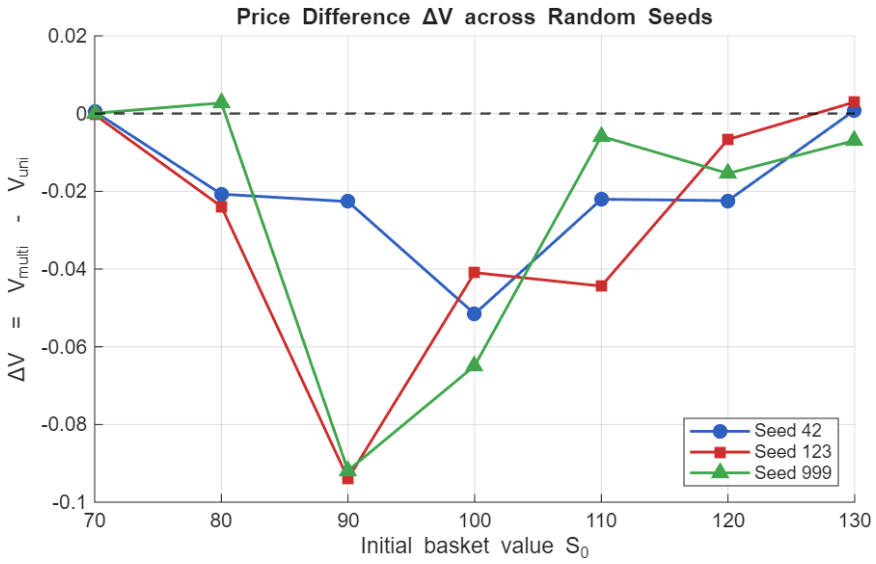


Figure 12: Price difference $\Delta\hat{V} = \hat{V}_{\text{multi}} - \hat{V}_{\text{uni}}$ as a function of S_0 for three independent random seeds (42, 123, 999). The dashed line marks $\Delta\hat{V} = 0$.

7.2.3 Sensitivity to correlation structure

Table 18 reports the estimated prices and standard errors for both regression specifications across five levels of uniform off-diagonal correlation $\rho \in \{0, 0.20, 0.40, 0.60, 0.80\}$, while keeping individual volatilities fixed at $\boldsymbol{\sigma} = (0.25, 0.32, 0.36, 0.28)$.

Table 18: American basket put prices under uniform correlation. $S_0 = 100$, $N_{\text{paths}} = 50{,}000$; remaining parameters as in Table 8.

ρ	**Uni**	**SE**	**Multi**	**SE**	$\Delta\hat{V}$
0.00	4.3529	0.0235	4.2801	0.0246	−0.0728
0.20	5.8069	0.0305	5.7252	0.0317	−0.0817
0.40	7.0292	0.0363	6.9609	0.0373	−0.0682
0.60	8.1007	0.0410	8.0543	0.0417	−0.0464
0.80	9.0834	0.0455	9.0598	0.0456	−0.0236

The basket put price increases monotonically with the correlation parameter, rising from 4.35 at $\rho = 0$ to 9.08 at $\rho = 0.80$ as shown in the left panel of Figure 13. This behaviour reflects the impact of correlation on the dispersion of the basket value B_T. When correlations are low, idiosyncratic movements across assets partially offset each other, leading to a more stable basket process and a lower probability that B_T falls significantly below the strike K. As ρ increases, assets tend to move jointly, reducing diversification effects and increasing the variability of B_T. A more dispersed basket value raises the expected payoff $(K - B_T)^+$, since larger downward deviations of B_T below K generate larger payoffs while upward deviations are bounded below by zero. Consequently, higher correlation leads to a higher option value.

The right panel of Figure 13 shows that the pricing gap $|\Delta\hat{V}|$ decreases as correlation

increases, from approximately 0.07–0.08 at low correlation levels to 0.024 at $\rho = 0.80$. The value at $\rho = 0.20$ (0.082) is slightly larger than at $\rho = 0$ (0.073); given that the Monte Carlo standard errors at these correlation levels are themselves of order 0.02–0.03 (Table 18), this small local deviation from monotonicity does not appear economically meaningful.

Apart from this local deviation, when correlations are low, the individual asset prices retain variation that is not fully reflected in the basket value B_t, so that the additional regressors used in the multivariate specification can lead to a different approximation of the continuation value and consequently to different exercise decisions and price estimates. As ρ increases, asset dynamics become increasingly driven by a common factor, so that B_t captures a larger share of the relevant state information and the price estimates produced by the two specifications become increasingly similar.

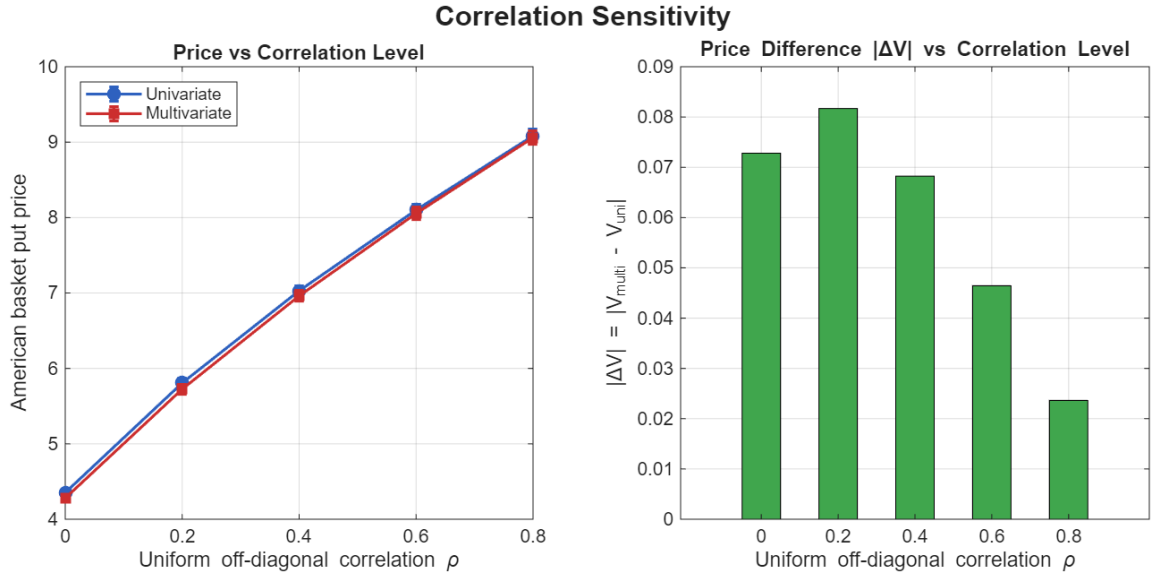


Figure 13: Sensitivity to uniform correlation level ($S_0 = 100$, $N_{\text{paths}} = 50{,}000$; remaining parameters as in Table 8). *Left*: basket put prices with 95% confidence intervals for the univariate (blue) and multivariate (red) specifications. *Right*: absolute price difference $|\Delta\hat{V}|$ across correlation levels.

7.2.4 Dimensionality Scaling

Table 19 reports prices, standard errors, price differences, computation times and number of regressors for both specifications as the number of assets varies over $d \in \{2, 4, 6, 8, 10\}$. For each value of d, a uniform correlation $\rho = 0.35$ is adopted, equal weights $w_j = 1/d$ are assigned to all assets and the volatility vector is drawn cyclically from the baseline vector $\boldsymbol{\sigma} = (0.25, 0.32, 0.36, 0.28)$.

Table 19: Dimensionality scaling results. $S_0 = 100$, $\rho = 0.35$ (uniform), $N_{\text{paths}} = 50{,}000$; remaining parameters as in Table 8.

d	**Uni**	**SE**	**Multi**	**SE**	$\Delta\hat{V}$	t_{uni} (s)	t_{multi} (s)	**nReg**
2	7.3976	0.0379	7.3505	0.0393	−0.0471	0.34	0.38	5
4	6.7342	0.0349	6.6579	0.0359	−0.0764	0.56	0.69	9
6	6.1231	0.0322	6.0936	0.0333	−0.0295	0.79	0.98	13
8	6.0343	0.0313	6.0023	0.0325	−0.0321	0.94	1.24	17
10	5.7683	0.0303	5.7476	0.0313	−0.0207	1.20	1.63	21

The basket put price decreases monotonically as d increases, from 7.40 at $d = 2$ to 5.77 at $d = 10$, as shown in the left panel of Figure 14. This pattern reflects the diversification effect of adding assets to an equally-weighted basket: as d grows, idiosyncratic movements across assets increasingly offset each other, reducing the effective volatility of B_T and therefore the expected payoff $(K - B_T)^+$. The rate of decrease slows for $d \geq 6$, consistent with the diminishing marginal diversification benefit as the basket already contains a large number of assets.

The absolute price difference $|\Delta\hat{V}|$ remains small across all values of d, not exceeding 0.077 in absolute value and staying below 1.2% of the corresponding univariate price at every dimension considered. No systematic monotone pattern is apparent, consistent with finite-sample variability in the regression-based approximation of the continuation value.

The right panel of Figure 14 shows that computation time exhibits a generally increasing pattern with d for both specifications. The univariate approach requires 0.34 seconds at $d = 2$ and 1.20 seconds at $d = 10$; the multivariate approach requires 0.38 and 1.63 seconds respectively. The gap between the two grows steadily with d, reflecting the larger regression system solved at each exercise date by the multivariate specification.

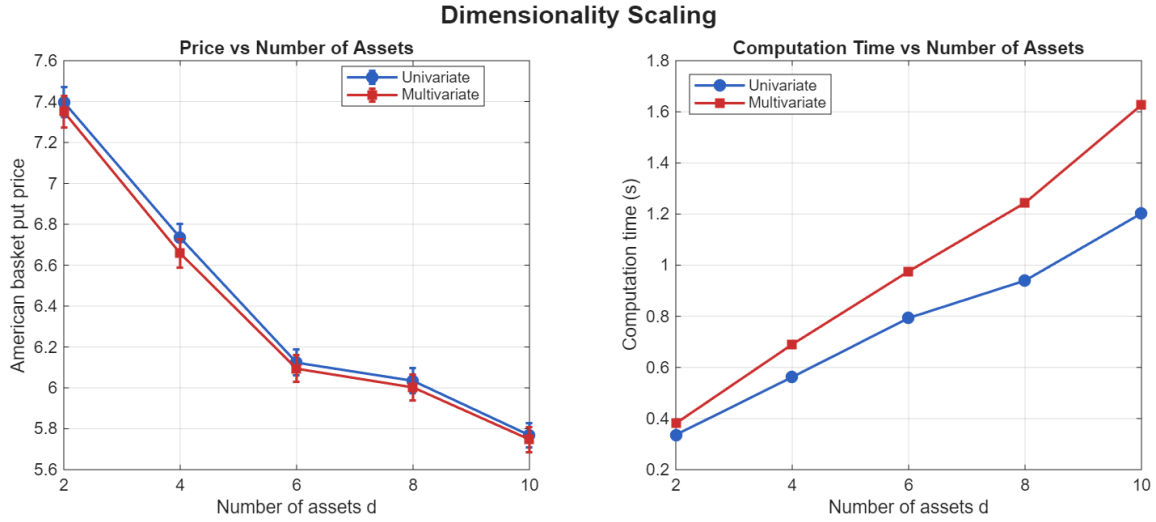


Figure 14: Dimensionality scaling results ($\rho = 0.35$ uniform, $S_0 = 100$, $N_{\text{paths}} = 50{,}000$; remaining parameters as in Table 8). *Left*: basket put prices with 95% confidence intervals. *Right*: computation time in seconds for the univariate (blue) and multivariate (red) specifications.

7.2.5 Early Exercise Analysis

At the ATM initial condition $S_0 = 100$, the univariate specification yields an early exercise fraction of 38.8%, compared to 35.0% for the multivariate specification, where early exercise is defined as exercise at any time step strictly before maturity. As discussed in Section 7.2.1, this difference reflects the different shape of the estimated continuation value surface implied by the two regression specifications and should not be interpreted as evidence that either specification yields a more accurate approximation of the optimal exercise boundary.

Figure 15 displays the empirical distribution of early exercise times for both specifications, restricted to paths exercised strictly before maturity. For the univariate specification, exercise times are distributed across most of the interval $[0, 1)$, with the highest relative frequencies (up to approximately 0.053) occurring roughly between $t = 0.2$ and $t = 0.55$ and somewhat lower but non-negligible frequencies over the rest of the contract life, including a modest increase in the final bins approaching maturity. The multivariate distribution displays an early concentration of comparable magnitude (relative frequency up to approximately 0.052 around $t = 0.2$–0.25), but is additionally characterized by a pronounced spike in the final bin near $t = 1$, reaching a relative frequency of approximately 0.077, higher than any bin observed in the univariate distribution.

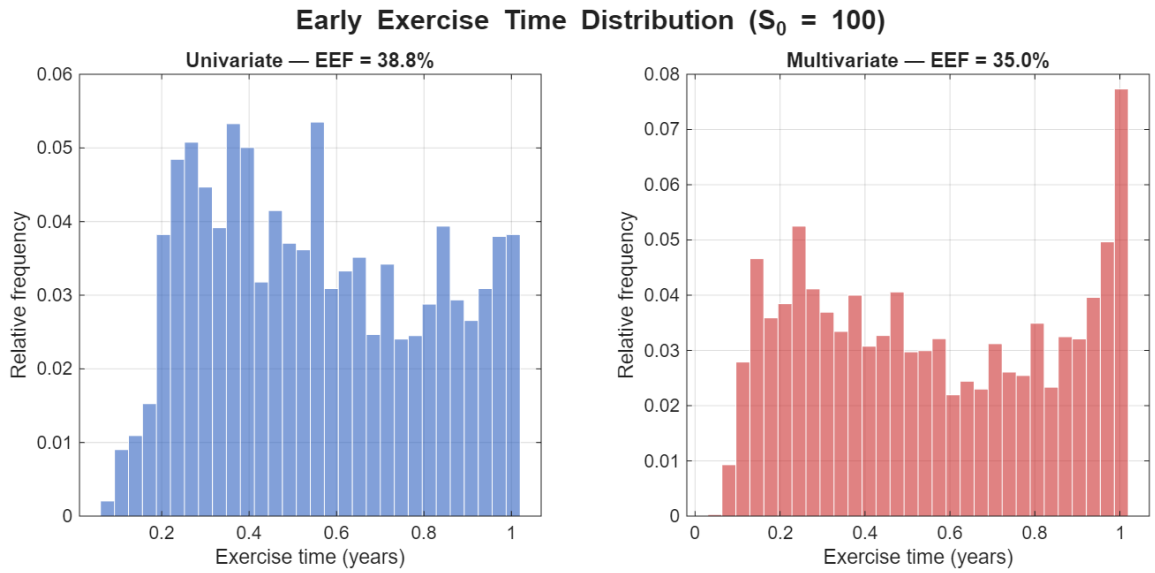


Figure 15: Empirical distribution of early exercise times for the univariate (*left*, EEF = 38.8%) and multivariate (*right*, EEF = 35.0%) specifications at $S_0 = 100$, $N_{\text{paths}} = 50{,}000$. Each histogram includes all paths exercised strictly before maturity.

Figure 16 illustrates the early exercise mechanism under the univariate specification on 60 randomly selected basket paths ($S_0 = 100$, seed = 42). All paths start from the common initial value $B_0 = K = 100$ and subsequently diverge: paths exercised early (blue) remain at or below the strike for most of the horizon, drifting down to approximately 60–90 by $t = 1$, while paths held to maturity (grey) trend upward, reaching values up to approximately 195 by $t = 1$. The filled circles mark the exercise points identified by the LSM backward induction, indicating exercise when the estimated continuation value does not exceed the immediate payoff; these points occur at basket values of approximately 78–87 (below the strike) and are spread across the time axis from approximately $t = 0.14$ to $t = 0.95$, reflecting the path-dependent nature of the stopping decision.

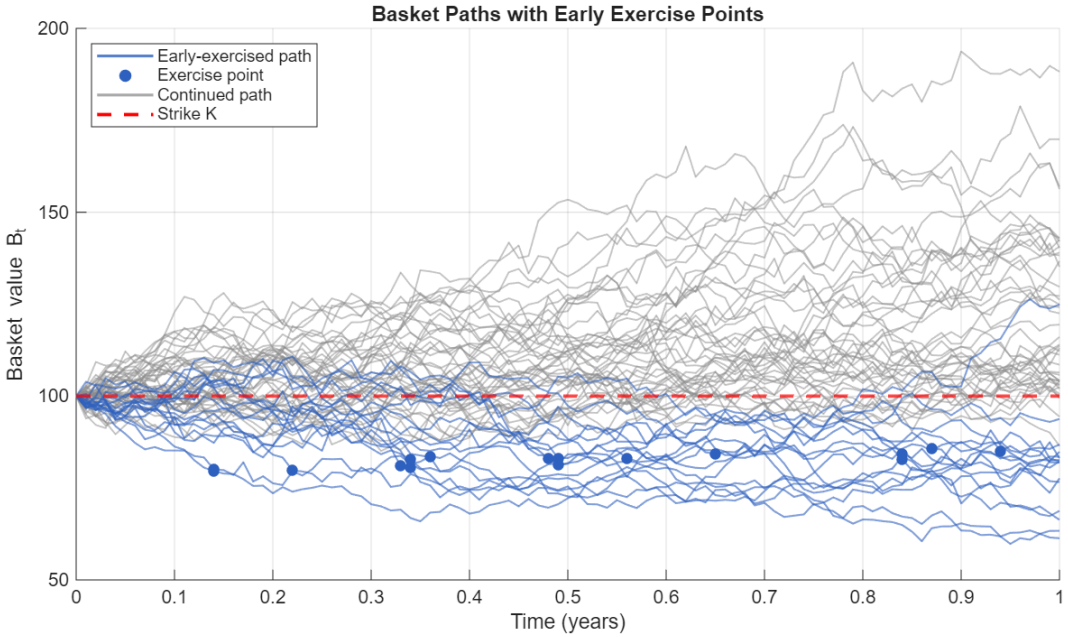


Figure 16: Sample basket paths with early exercise points ($S_0 = 100$, univariate specification, 60 randomly selected paths, seed = 42). Blue lines: paths exercised early; filled circles mark the exercise point. Grey lines: paths held to maturity. Red dashed line: strike $K = 100$.

8 Empirical Application: American Basket Put on Big-Tech Equities

The preceding chapters developed and validated the LSM algorithm in a controlled setting, where model parameters were specified ex ante and benchmark results were either available analytically or reproducible by construction. This chapter extends the analysis to real market data by applying the basket LSM framework of Chapter 6 to a set of technology equities. The objective is twofold: to examine whether the parameters calibrated from historical prices are economically coherent and to assess how the main design choices of the algorithm, i.e. the regression state representation and the basket weighting scheme, affect pricing outcomes in an empirical setting.

8.1 Data and Asset Selection

The analysis is conducted on four equity securities listed on the NASDAQ exchange: Amazon (`AMZN`), Apple (`AAPL`), Microsoft (`MSFT`) and NVIDIA (`NVDA`). All four firms rank among the largest constituents by market capitalization of the NASDAQ-100 index, making the basket a natural proxy for broad technology-sector exposure. At the same time, despite belonging to the same sector, their business activities differ substantially, generating a non-trivial volatility and correlation structure. NVIDIA is primarily exposed to the semiconductor and AI-hardware cycle, Apple is strongly linked to consumer demand and product replacement cycles, while Amazon and Microsoft maintain significant exposure to cloud-computing services. These differences are also reflected in the behaviour of the assets over the sample period: NVIDIA experienced exceptional appreciation during the AI-driven market expansion, whereas the remaining three stocks exhibited comparatively more moderate growth patterns. Daily closing prices were sourced from Investing.com [2026], covering the period from 1 January 2020 to 31 May 2026, for a total of $N_{\text{common}} = 1{,}610$ trading days per asset. The sample window was chosen to yield reliable estimates of annualized volatilities and pairwise correlations from daily log-returns, while remaining close enough to the pricing date to reflect current market conditions. Moreover, it spans several distinct market regimes, including the Covid-19 shock of early 2020, the subsequent recovery and the AI-driven rally of 2023–2026 providing a heterogeneous basis for parameter estimation. To ensure a common observation window, the analysis aligns the four series by retaining, for each asset, the most recent N_{common} observations, producing a rectangular price matrix $P \in \mathbb{R}^{N_{\text{common}} \times d}$ with $d = 4$. In the present dataset, the four series already share the same length and no observations are discarded.

8.2 Parameter Calibration

8.2.1 Volatility and Drift

Let $P_{t,i}$ denote the closing price of asset i on trading day t, for $t = 1, \ldots, N_{\text{common}}$ and $i = 1, \ldots, d$. The daily log-return of asset i on day t is

$$r_{t,i} \;=\; \log\!\left(\frac{P_{t,i}}{P_{t-1,i}}\right), \qquad t = 2, \ldots, N_{\text{common}}, \quad i = 1, \ldots, d. \tag{51}$$

The annualized volatility for asset i is estimated as

$$\hat{\sigma}_i \;=\; \text{std}(\{r_{t,i}\}_{t=2}^{N_{\text{common}}}) \cdot \sqrt{252}, \tag{52}$$

where std$(\cdot)$ denotes the sample standard deviation and 252 is the conventional number of trading days per year. Since volatility is invariant to the change of measure from $\mathbb{P}$ to $\mathbb{Q}$, the estimate $\hat{\sigma}_i$ is used directly in the risk-neutral simulation of (37).
The annualized historical drift $\hat{\mu}_i = \bar{r}_i \cdot 252$, where $\bar{r}_i$ denotes the sample mean of $\{r_{t,i}\}_{t=2}^{N_{\text{common}}}$, is reported for completeness but plays no role in pricing: under $\mathbb{Q}$ the drift of every asset is replaced by the risk-free rate r.
The calibrated parameters are summarized in Table 20.

Table 20: Calibrated parameters for the four-asset basket. S_0 is the last closing price in the common window (May 2026), $\hat{\sigma}$ and $\hat{\mu}$ are annualized estimates from daily log-returns over the full 1,610-day window.

Ticker	S_0 **(USD)**	$\hat{\sigma}$	$\hat{\mu}$
AMZN	270.64	35.3%	+16.4%
AAPL	312.06	31.3%	+22.3%
MSFT	450.24	29.8%	+16.1%
NVDA	211.14	52.0%	+55.8%

The most prominent feature is the substantially higher volatility of NVIDIA ($\hat{\sigma}_{\text{NVDA}} = 52.0\%$), approximately 1.5 to 1.7 times the volatility of the other three assets. As visible in Figure 17, over the estimation window NVIDIA appreciated by a factor of approximately 35 relative to its price at the start of the sample, while AAPL, AMZN and MSFT gained factors of roughly 4.2, 2.9 and 2.8 respectively. The daily log-return series in Figure 18 confirm this pattern: NVIDIA exhibits markedly larger return swings throughout, with occasional daily moves exceeding $\pm 20\%$, while periods of elevated volatility appear simultaneously across all four series in the early part of the sample, consistent with a common market factor.

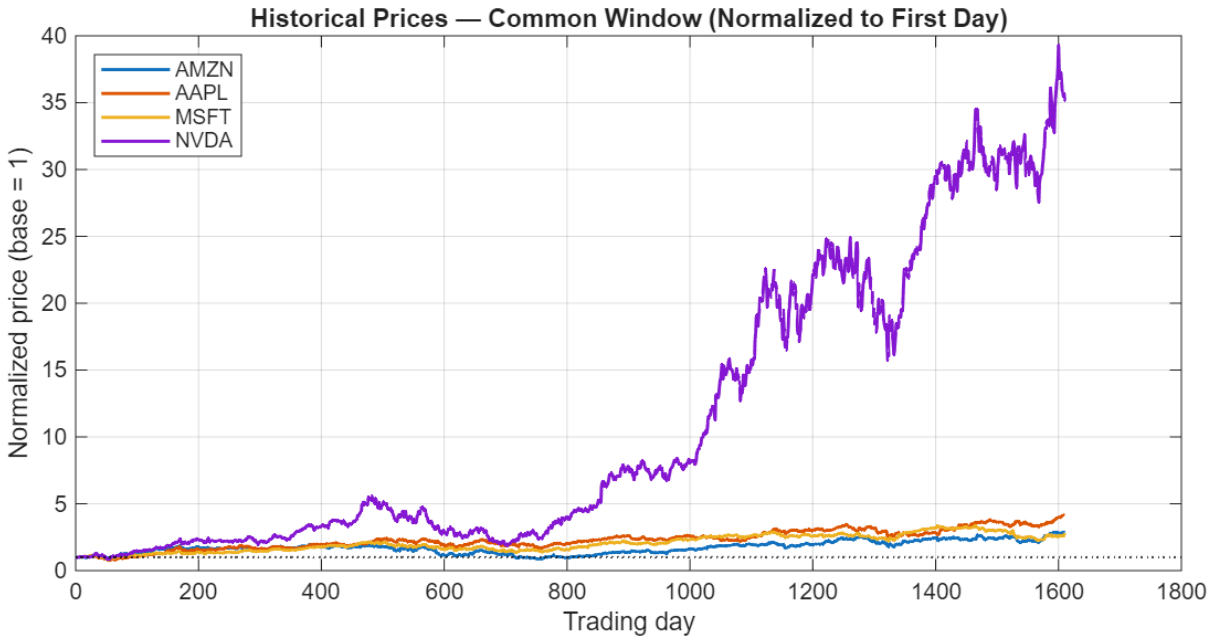


Figure 17: Normalized historical prices for the four constituents over the common window of 1,610 trading days (January 2020 – May 2026). Each series is divided by its value on the first trading day to allow direct comparison of cumulative returns across assets.

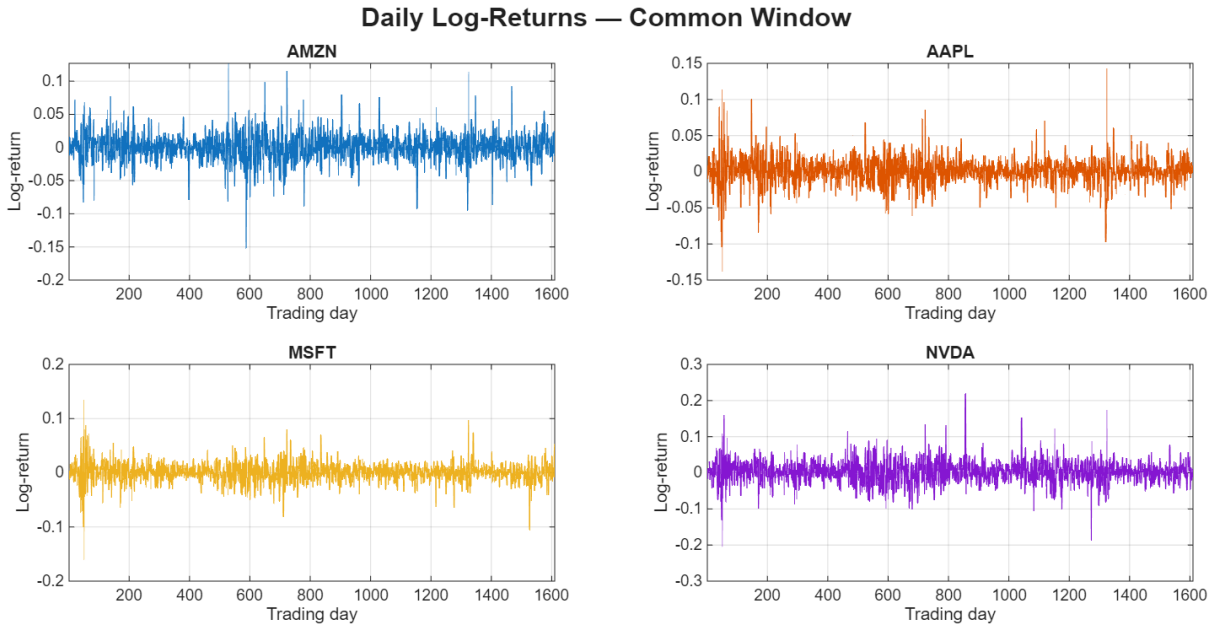


Figure 18: Daily log-returns for the four constituents over the common window of 1,610 trading days. Note that the vertical axis range differs across panels (from approximately ± 0.15 for AAPL to approximately ± 0.3 for NVDA), reflecting the markedly higher return volatility of NVIDIA relative to the other three constituents.

8.2.2 Correlation Matrix and Cholesky Factorization

Let $R \in \mathbb{R}^{(N_{\text{common}}-1)\times d}$ denote the matrix of daily log-returns, with entries $R_{t,i} = r_{t,i}$ as defined in (51). The sample Pearson correlation between assets i and j is estimated as

$$\hat{\rho}_{ij} \;=\; \frac{\displaystyle\sum_{t=2}^{N_{\text{common}}} (r_{t,i} - \bar{r}_i)(r_{t,j} - \bar{r}_j)}{\sqrt{\displaystyle\sum_{t=2}^{N_{\text{common}}} (r_{t,i} - \bar{r}_i)^2 \;\cdot\; \sum_{t=2}^{N_{\text{common}}} (r_{t,j} - \bar{r}_j)^2}}, \qquad i, j = 1, \ldots, d, \tag{53}$$

where $\bar{r}_i = \frac{1}{N_{\text{common}}-1} \sum_{t=2}^{N_{\text{common}}} r_{t,i}$. The resulting estimate is

$$\hat{\rho} \;=\; \begin{pmatrix} & \text{AMZN} & \text{AAPL} & \text{MSFT} & \text{NVDA} \\ \text{AMZN} & 1.000 & 0.576 & 0.643 & 0.571 \\ \text{AAPL} & 0.576 & 1.000 & 0.676 & 0.571 \\ \text{MSFT} & 0.643 & 0.676 & 1.000 & 0.657 \\ \text{NVDA} & 0.571 & 0.571 & 0.657 & 1.000 \end{pmatrix} \tag{54}$$

All pairwise correlations are positive and lie in the interval [0.571, 0.676], reflecting the common sectoral exposure of the four assets to the broad technology cycle. The most correlated pair is AAPL–MSFT ($\hat{\rho} = 0.676$); the two lowest correlations are both observed for NVDA, with AMZN and with AAPL ($\hat{\rho} = 0.571$ in both cases).
The eigenvalues of $\hat{\rho}$ are approximately $\{0.294, 0.427, 0.431, 2.849\}$, all strictly positive, confirming that the matrix is positive definite. The Cholesky factorization $\hat{\Sigma} = AA^\top$, where $\hat{\Sigma} = \text{diag}(\hat{\boldsymbol{\sigma}})\,\hat{\rho}\,\text{diag}(\hat{\boldsymbol{\sigma}})$, therefore exists and is unique, and the lower-triangular factor A is used to generate correlated GBM increments as described in (38).
The average off-diagonal correlation is $\bar{\rho} = 0.616$, which serves as a reference marker in the correlation sensitivity analysis of Section 8.6.

8.2.3 Weighting Schemes and Implied Basket Volatility

Two weighting schemes are considered throughout. The **equal-weight** (EW) scheme assigns $w_i^{\text{ew}} = 1/d = 0.25$ to each asset, consistent with the synthetic experiments of Chapter 6. The **value-weight** (VW) scheme sets weights proportional to approximate market capitalization[12] at the end of the estimation window, yielding

$$\mathbf{w}^{\text{vw}} \;=\; (0.184,\; 0.286,\; 0.208,\; 0.321)^\top,$$

where the entries correspond to AMZN, AAPL, MSFT, NVDA respectively. NVIDIA receives the largest weight (0.321), reflecting its dominant market capitalization at the time of writing.

The implied annualized basket volatility under each scheme is

$$\hat{\sigma}_B^{(\cdot)} \;=\; \sqrt{\mathbf{w}^{(\cdot)\top} \hat{\Sigma}\, \mathbf{w}^{(\cdot)}}, \tag{55}$$

[12]The approximate market capitalizations used are: AMZN \$2,950 bn, AAPL \$4,590 bn, MSFT \$3,340 bn, NVDA \$5,150 bn (May 2026) taken from Investing.com [2026]. Weights are computed as $w_i^{\text{vw}} = m_i / \sum_j m_j$.

giving $\hat{\sigma}_B^{\text{ew}} = 31.3\%$ and $\hat{\sigma}_B^{\text{vw}} = 32.6\%$. As shown in Figure 19, both figures lie well below NVIDIA's individual volatility of 52.0%: even though all pairwise correlations are high, combining four assets attenuates the basket volatility relative to its most volatile constituent due to the diversification effect, as discussed in Section 7.2.4. The 1.3 percentage point differential between the two schemes reflects the higher weight assigned to NVIDIA under VW.

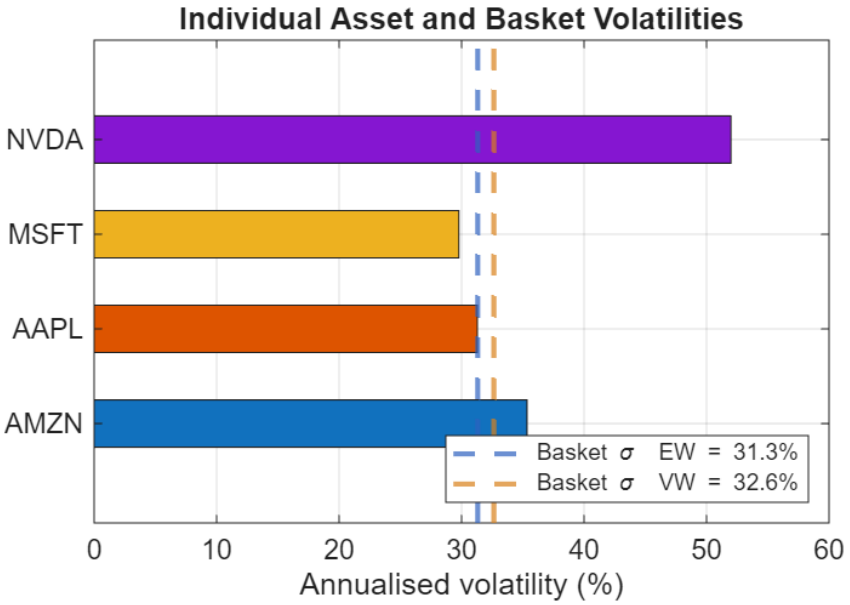


Figure 19: Annualized volatilities for the four individual assets and implied basket volatilities under the equal-weight and value-weight schemes (dashed lines).

Strike and risk-free rate. The strike is set to $K = 310$ USD, close to the initial equal-weight basket value $B_0^{\text{ew}} = \mathbf{w}^{\text{ew}\top} S_0 = 311.02$ USD, so that neither immediate exercise nor continuation is a priori dominant. The risk-free rate is $r = 3.8\%$, corresponding to the annualized yield on the 1-year US Treasury bill at the end of the estimation window [Street Stats, 2026] and the option maturity is $T = 1$ year.

8.3 Baseline Results: Univariate vs. Multivariate Regression

Table 21 reports the estimated American basket put prices under both weighting schemes and both regression specifications, together with Monte Carlo standard errors (SE) and early exercise fractions (EEF), obtained with $N_{\text{paths}} = 50{,}000$, $N_{\text{steps}} = 100$ and antithetic variates.

Table 21: Baseline American basket put prices under equal-weight (EW) and value-weight (VW) schemes, for univariate and multivariate regression specifications. SE: Monte Carlo standard error; EEF: early exercise fraction. Parameters: $K = 310$, $r = 3.8\%$, $T = 1$ year, $N_{\text{paths}} = 50{,}000$, $N_{\text{steps}} = 100$, seed = 123.

Configuration	**Price (USD)**	**SE**	**EEF**
EW — Univariate	31.1855	0.1541	39.40%
EW — Multivariate	31.0311	0.1559	36.70%
VW — Univariate	36.6550	0.1640	44.51%
VW — Multivariate	36.4419	0.1655	41.75%

Multi−Uni (EW)	$= -0.1544$	Multi−Uni (VW)	$= -0.2131$
VW−EW (Uni)	$= +5.4695$	VW−EW (Multi)	$= +5.4108$

Univariate vs. multivariate. The price difference between the multivariate and univariate specifications is −0.154 USD for EW and −0.213 USD for VW, representing less than 0.5% and 0.6% of the respective estimated prices. These results are consistent with the finding of Section 7.2.1 for the synthetic experiments: the scalar basket value B_t captures most of the information relevant for the exercise decision and expanding the state space to the full price vector $\mathbf{S}_t$ does not yield a meaningful improvement in the estimated option price.
The EEF is slightly higher under the univariate specification for both weighting schemes (39.4% vs. 36.7% for EW; 44.5% vs. 41.8% for VW), although the effect is modest. This pattern is consistent with the observation of Section 7.2.5 that conditioning on additional state variables tends to assign higher estimated continuation values to certain ITM paths, deferring early exercise.

Equal weight vs. value weight. The VW basket is priced approximately 5.41–5.47 USD higher than the EW basket across the two regression specifications, a premium consistent with the higher effective basket volatility generated by the value-weighting scheme (32.6% vs. 31.3%). Assigning a larger weight to NVIDIA, the most volatile constituent, raises the basket's effective volatility and, for a put option, increases both the probability of the basket falling below the strike and the expected depth of ITM paths at exercise. The EEF is correspondingly higher for the VW basket under both specifications, reinforcing this interpretation.

Robustness of ΔV across seeds. To verify that the near-equivalence of the two regression specifications is not an artefact of the baseline seed, $\Delta V = V^{\text{multi}} - V^{\text{uni}}$ is computed across all seven moneyness levels for three independent seeds (42, 123, 999); results are shown in Figure 20. ΔV is negative for all three seeds throughout the range $s \in [0.70, 1.10]$, with the largest absolute value, $\Delta V \approx -0.235$ USD, occurring for seed 999 at $s = 0.90$, corresponding to approximately 0.51% of the univariate price at that level (46.14 USD). The single exception across all 21 seed–moneyness combinations is seed 999 at $s = 1.20$, where $\Delta V \approx +0.008$ USD, a value indistinguishable from zero relative to the corresponding option price of approximately 13 USD. At $s = 1.30$, ΔV returns to

small negative values for all three seeds (-0.034, -0.017 and -0.018 USD, respectively). The sign and magnitude of ΔV are therefore stable across independent simulation draws, confirming that the near-equivalence of the two specifications is a structural feature of this estimation problem rather than a seed-specific artefact.

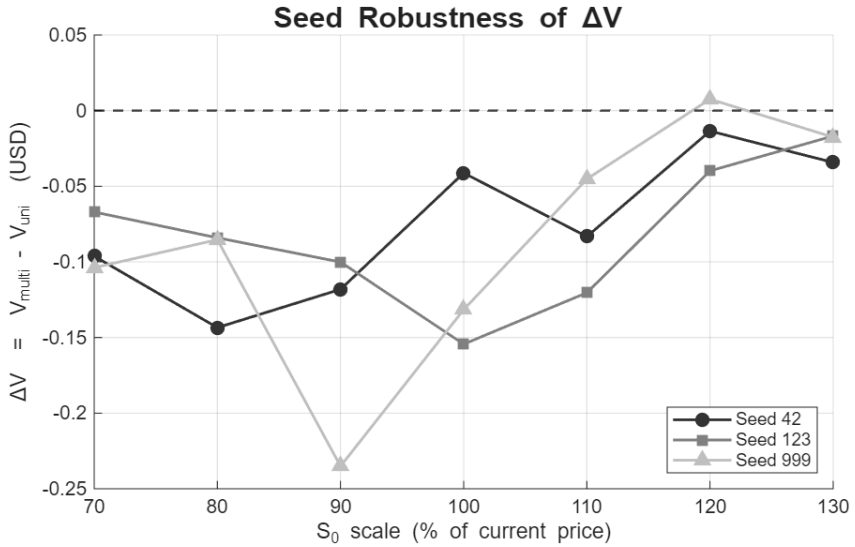


Figure 20: Price difference $\Delta V = V^{\text{multi}} - V^{\text{uni}}$ across moneyness levels for three independent seeds (42, 123, 999). The dashed line marks $\Delta V = 0$. ΔV is consistently negative and small across seeds and moneyness levels.

8.4 Moneyness Sensitivity

To examine how the basket put price varies with the initial level of the underlying, all four asset prices are scaled uniformly by a factor $s \in \{0.70, 0.80, 0.90, 1.00, 1.10, 1.20, 1.30\}$. Uniform scaling preserves the cross-sectional price ratios across assets, ensuring that the basket moneyness relative to $K = 310$ varies monotonically with s without distorting the relative position of the four constituents. Results are reported in Table 22 and Figure 21.

Table 22: Moneyness sensitivity under the equal-weight scheme. Scale s is applied uniformly to all four asset prices. Diff = Multi − Uni. $N_{\text{paths}} = 50{,}000$.

Scale	Uni	SE	Multi	SE	Diff	EEF Uni	EEF Multi
70%	92.694	0.103	92.627	0.101	−0.067	93.2%	93.8%
80%	66.539	0.161	66.455	0.161	−0.084	73.6%	73.0%
90%	46.140	0.168	46.040	0.169	−0.100	54.7%	53.4%
100%	31.186	0.154	31.031	0.156	−0.154	39.4%	36.7%
110%	20.484	0.132	20.363	0.134	−0.120	27.5%	24.4%
120%	12.999	0.108	12.959	0.110	−0.040	18.6%	15.8%
130%	8.191	0.087	8.174	0.089	−0.017	11.7%	9.7%

The price curve is monotonically decreasing in s, as expected: a higher initial basket level implies a lower probability of the basket finishing below the strike and hence a

lower put value. The EEF declines sharply as the basket moves from deep-ITM to OTM: at $s = 0.70$ virtually all paths are exercised early (EEF $> 93\%$), while at $s = 1.30$ approximately $10 - 12\%$ paths trigger early exercise. This pattern is qualitatively consistent with the finding documented in Table 17 for the synthetic experiments, although the specific EEF values differ due to the distinct parameter configurations of the two settings.

The pricing gap $|\Delta\hat{V}|$ remains small across all moneyness levels, peaking near the at-the-money region ($s = 1.00$, $|\Delta\hat{V}| = 0.154$ USD) and becoming negligible at both extremes. This is intuitive: near the money, continuation and exercise values are hardest to distinguish, so the accuracy of the continuation value approximation becomes most relevant. The EEF pattern recalls the baseline results of Section 8.3, with the multivariate specification producing slightly lower exercise frequencies at most moneyness levels. The only exception occurs at $s = 0.70$, where the two values are nearly identical (93.2% vs. 93.8%); because virtually all paths exercise early at this level under both specifications, the difference is unlikely to reflect a systematic economic effect.

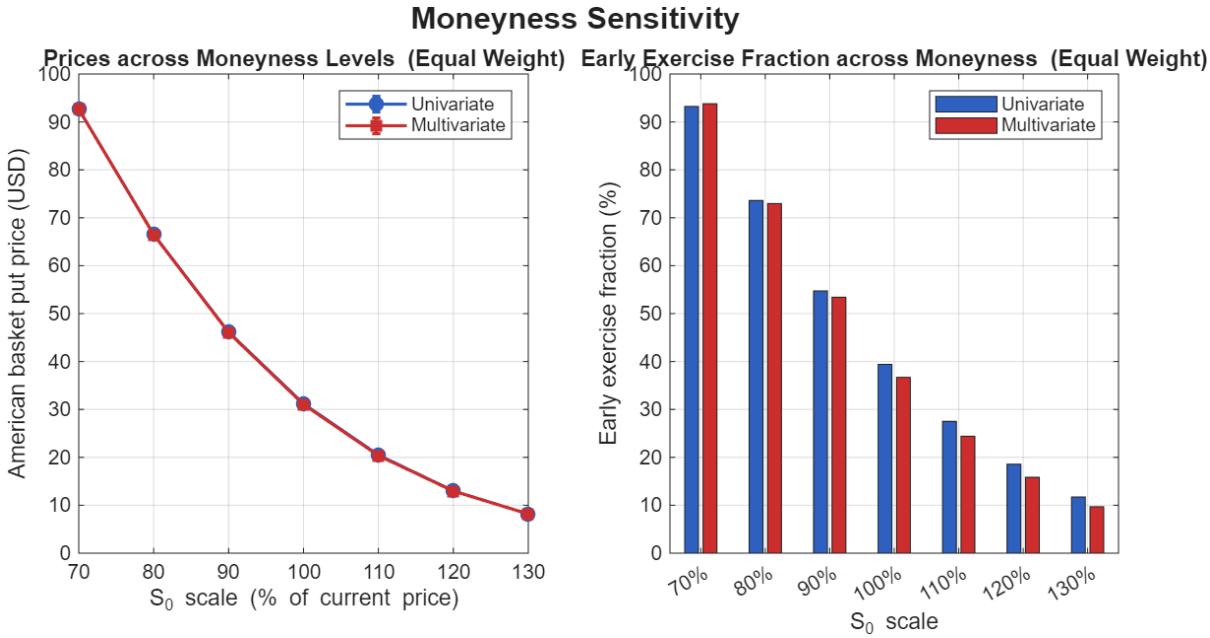


Figure 21: American basket put prices (*left*) and early exercise fractions (*right*) across moneyness levels under the equal-weight scheme, for the univariate (blue) and multivariate (red) specifications. Error bars denote 95% confidence intervals (± 1.96 SE).

8.5 Equal-Weight vs. Value-Weight Across Moneyness

Figure 22 compares the price curves of the two weighting schemes across all seven moneyness levels, for both regression methods. The value-weight basket commands a consistently higher price at every scale s, with the price gap ranging from approximately 6.9 USD at $s = 0.70$ to approximately 2.5 USD at $s = 1.30$. The gap therefore decreases as the basket moves out-of-the-money, reflecting the fact that changes in volatility have a larger impact on option value when the payoff is more sensitive to downward movements in the underlying. The ordering is stable across both regression methods, as confirmed by Table 23: the two panels of Figure 22 are visually indistinguishable, indicating that the choice of regression specification has negligible impact on the EW–VW pricing differen-

tial. The result is consistent with the volatility analysis of Section 8.2.3 where the higher weight assigned to NVIDIA under the VW scheme raises the implied basket volatility, shifting the entire put price curve upward through the increased probability of large downward movements in B_t.

Table 23: Equal-weight vs. value-weight prices across moneyness levels, for both regression methods. $N_{\text{paths}} = 50{,}000$.

Scale	EW Uni	VW Uni	EW Multi	VW Multi
70%	92.694	99.584	92.627	99.655
80%	66.539	73.437	66.455	73.422
90%	46.140	52.548	46.040	52.449
100%	31.186	36.655	31.031	36.442
110%	20.484	24.863	20.363	24.734
120%	12.999	16.522	12.959	16.445
130%	8.191	10.682	8.174	10.706

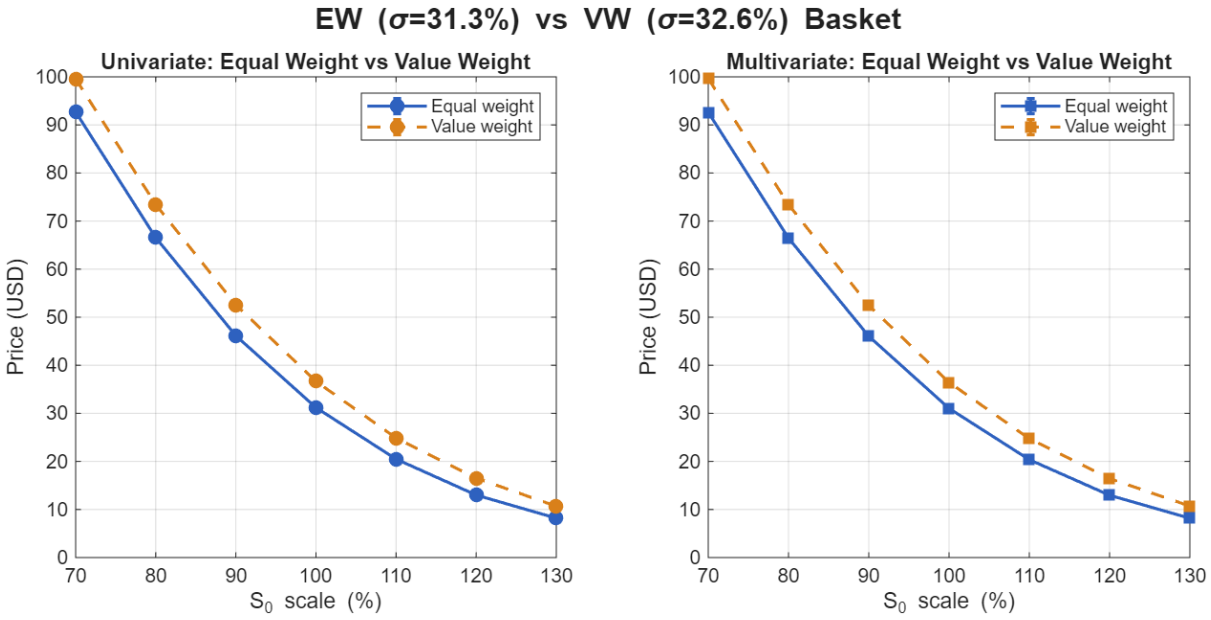


Figure 22: American basket put prices under equal-weight (EW, blue) and value-weight (VW, orange dashed) schemes, for univariate (*left*) and multivariate (*right*) regression. The VW curve lies uniformly above the EW curve at all moneyness levels; the gap narrows from approximately 6.9 USD at $s = 70\%$ to 2.5 USD at $s = 130\%$. The two panels are visually indistinguishable, reflecting the negligible influence of the regression specification on the EW–VW differential.

8.6 Correlation Sensitivity

To quantify the sensitivity of the basket put price to the assumed dependence structure, the off-diagonal correlation is varied uniformly over the grid $\rho \in \{0, 0.20, 0.40, 0.60, 0.80\}$,

while keeping the volatility vector $\hat{\boldsymbol{\sigma}}$ and initial prices S_0 at their calibrated values. For each scenario, the correlation matrix takes the form

$$\rho_{ij} \;=\; \rho \cdot \mathbf{1}_{i \neq j} + \mathbf{1}_{i=j}, \tag{56}$$

and the covariance matrix is reconstructed as $\Sigma = \mathrm{diag}(\hat{\boldsymbol{\sigma}})\, \rho\, \mathrm{diag}(\hat{\boldsymbol{\sigma}})$ before recomputing the Cholesky factor. The historical average off-diagonal correlation $\bar{\rho} = 0.616$ is marked as a reference on the figures but does not constitute a separate pricing scenario. Results are reported in Table 24 and Figure 23.

Table 24: Correlation sensitivity under the equal-weight scheme. Prices are computed for a uniform off-diagonal correlation ρ. Diff = Multi − Uni. S_0 at baseline (scale = 100%), $N_{\text{paths}} = 50{,}000$.

ρ	**Uni**	**SE**	**Multi**	**SE**	**Diff**
0.00	17.264	0.092	17.038	0.096	−0.226
0.20	22.470	0.116	22.250	0.120	−0.220
0.40	26.896	0.136	26.648	0.139	−0.248
0.60	30.807	0.153	30.690	0.155	−0.118
0.80	34.375	0.168	34.272	0.168	−0.103

The basket put price increases monotonically with ρ, rising from 17.26 USD under independence to 34.38 USD at $\rho = 0.80$. This reflects the standard diversification effect: higher correlation reduces cross-asset risk dispersion, increases the variance of the basket and raises the probability of large negative movements in B_t. The historical average $\bar{\rho} = 0.616$ lies between the $\rho = 0.60$ and $\rho = 0.80$ grid points and the corresponding baseline EW price of 31.19 USD is consistent with this positioning, as shown in the left panel of Figure 23.

The absolute pricing gap $|\Delta V|$ is larger at low correlation levels (0.220–0.248 USD for $\rho \leq 0.40$) and smaller at high ones (0.103–0.118 USD for $\rho \geq 0.60$). The pattern is not strictly monotone: $|\Delta V|$ rises slightly from $\rho = 0.20$ to $\rho = 0.40$ before dropping sharply, as visible in the right panel of Figure 23. The overall decline is consistent with the finding of Section 7.2.3: as correlation increases, asset dynamics become increasingly driven by a common factor, making B_t a more sufficient state variable for the exercise decision and reducing the marginal value of the multivariate specification.

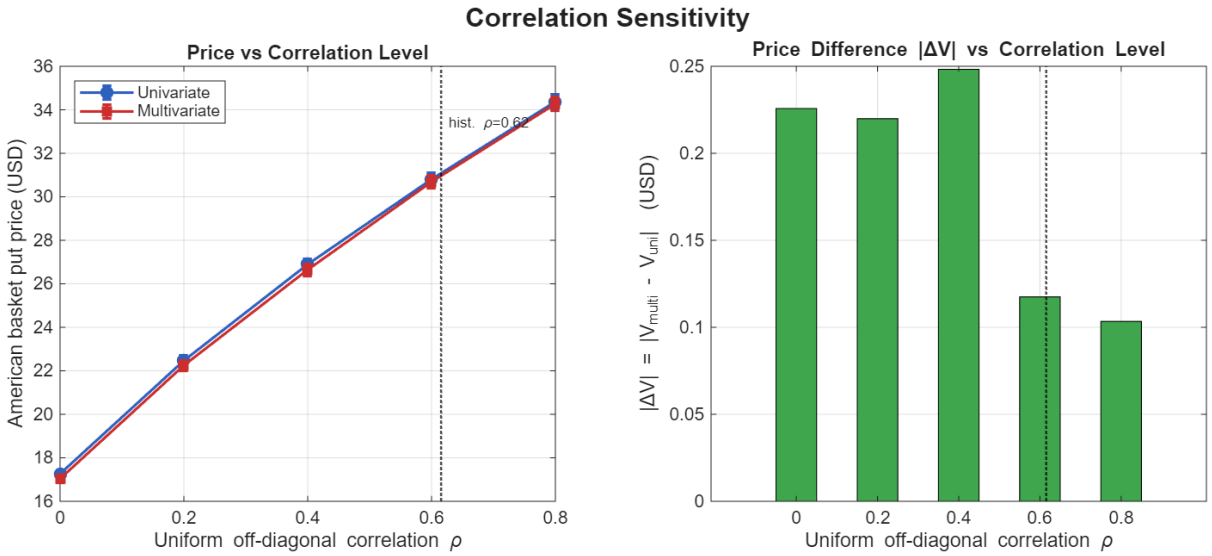


Figure 23: Left panel: basket put price under the univariate (blue) and multivariate (red) specifications as a function of uniform off-diagonal correlation ρ. Right panel: absolute price difference $|\Delta V|$ across correlation levels. The dotted vertical line marks the historical average $\bar{\rho} = 0.616$ in both panels.

8.7 Early Exercise Analysis

Figure 24 displays the distribution of early exercise times for the ATM equal-weight basket ($s = 1.00$, $K = 310$), separately for the two regression specifications. The EEF is 39.4% for the univariate and 36.7% for the multivariate specification, as reported in Table 21. The two distributions differ in their temporal structure. Under the univariate specification, exercise events are spread across the life of the contract without a clear temporal trend, with an irregular pattern and a local peak around $t \approx 0.55$. Under the multivariate specification, the distribution exhibits a clear increasing trend toward maturity, with the highest concentration of exercise events near $t = 1$. Both distributions show very little exercise activity in the first few months of the contract ($t < 0.15$), consistent with the near-ATM initial condition: with $B_0 = 311$ USD and $K = 310$ USD, few paths have moved sufficiently below the strike in the early part of the simulation to make immediate exercise optimal.

The difference in temporal structure is consistent with the interpretation developed in Section 8.3: the multivariate regression assigns higher estimated continuation values to a subset of ITM paths at intermediate dates, making it optimal to defer exercise toward later times. The EEF difference of 2.7 percentage points therefore reflects not only a reduction in the number of early-exercised paths, but also a shift in the timing of exercise decisions under the multivariate specification.

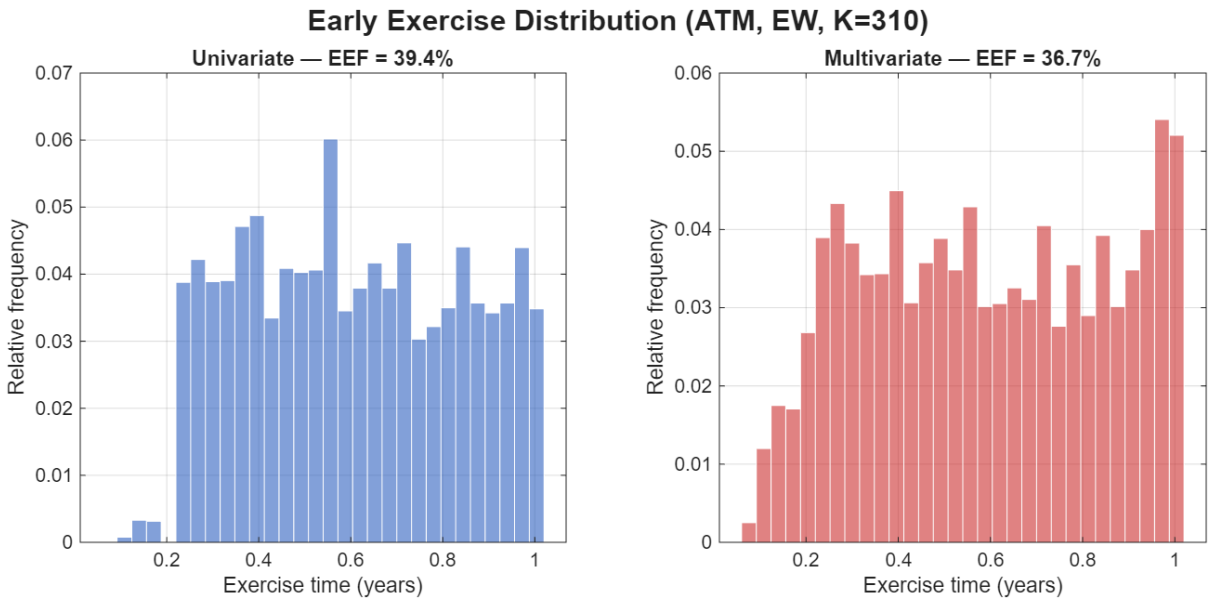


Figure 24: Distribution of early exercise times for the ATM equal-weight basket ($K = 310$, $s = 1.00$). Left: univariate regression (EEF = 39.4%). Right: multivariate regression (EEF = 36.7%). The multivariate distribution exhibits a more pronounced concentration near maturity.

Figure 25 displays 60 randomly selected basket paths under the equal-weight univariate specification. Blue paths are those for which the LSM algorithm triggers early exercise, with filled circles marking the exercise point; grey paths are held to maturity.
All exercise points lie strictly below the strike $K = 310$ USD, confirming that exercise occurs only when the option is ITM. No exercise is triggered in the first few months of the contract, consistent with the near-ATM initial condition discussed above. Most notably, the exercise points are concentrated between approximately 210 and 260 USD, well below the strike and among the exercised paths several appear to have entered the ITM region before the exercise point is reached. This indicates that crossing below K is not sufficient to trigger exercise: the LSM algorithm acts only when the estimated immediate payoff exceeds the estimated continuation value, which in the present calibration tends to occur only when the basket is sufficiently below the strike. This observation is consistent with the estimated early exercise boundary of Figure 26 , which lies between 215 and 257 USD throughout the contract life.

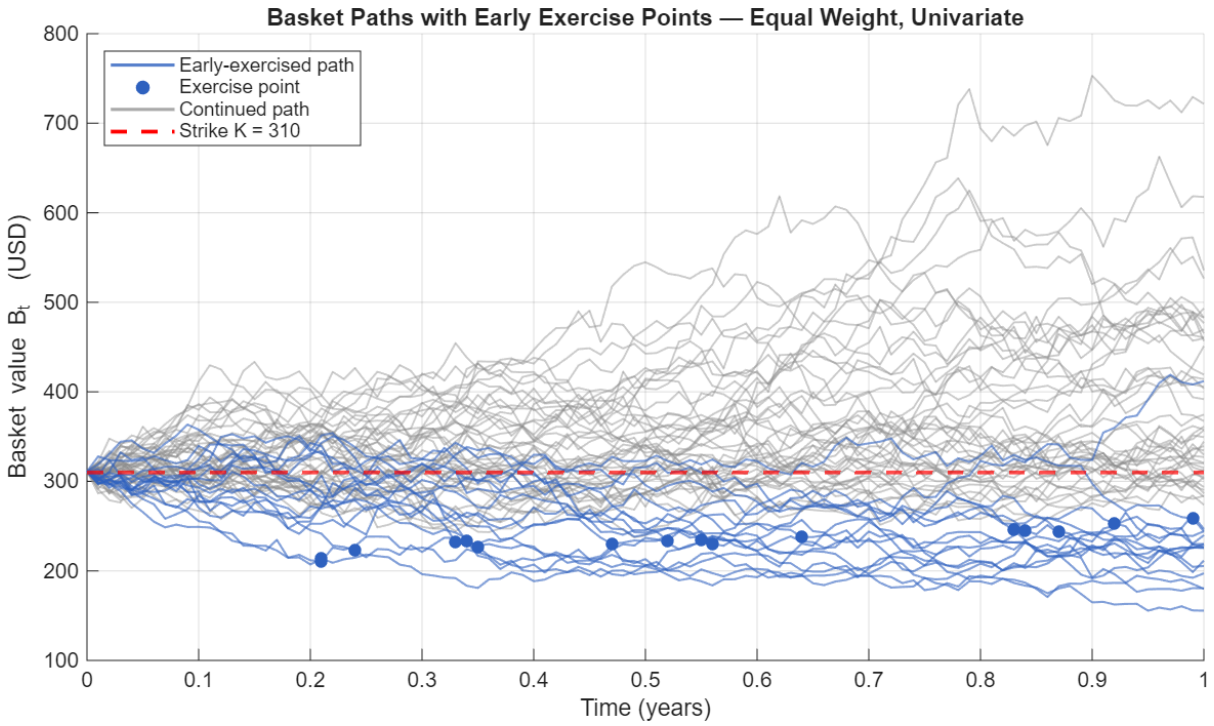


Figure 25: Sixty simulated basket paths (equal weight, univariate regression, seed = 42). Blue paths: exercised early; filled circles: exercise points. Grey paths: held to maturity. The dashed red line marks the strike $K = 310$. Early exercise occurs exclusively in the region below the strike.

Estimated Early Exercise Boundary Figure 26 displays the estimated early exercise boundary $B^*(t)$, computed at each discrete time step as the average basket value over all paths exercised at that step under the univariate specification. This quantity is the basket analogue of the optimal exercise boundary $b^*(t)$ introduced in Section 3.4, where exercise occurs at the first time the underlying falls below the boundary: in the present multi-asset setting, the relevant state variable is the basket value B_t rather than the individual asset price S_t, so the boundary is defined in the (t, B_t) space as

$$\tau^* = \inf\{t \geq 0 : B_t \leq B^*(t)\}.$$

It is important to note that $B^*(t)$ is an empirical estimate rather than the true optimal stopping boundary: paths exercised at step t are precisely those for which the LSM algorithm estimates the immediate payoff to exceed the continuation value, so their average basket value provides a proxy for the location of the exercise boundary at each date.
The estimated boundary increases from approximately 215 USD in the early part of the contract to approximately 257 USD near maturity, with a generally increasing trend that becomes smoother after $t \approx 0.25$. Consistent with the theoretical behaviour of American put options, the boundary is expected to approach the strike as maturity is reached, since the remaining time value shrinks and exercise becomes optimal at basket levels progressively closer to K. In the present discretization with $N_{\text{steps}} = 100$, the last estimated point reaches approximately 257 USD, still below $K = 310$, reflecting the residual time value at the final discrete exercise date.

Throughout the contract life $B^*(t)$ lies substantially below the strike, between approximately 69% and 83% of K. Given the substantial basket volatility ($\hat{\sigma}_B^{\text{ew}} = 31.3\%$) and

the remaining time to maturity, the continuation value remains significant even for moderately ITM paths, so immediate exercise is optimal only when the basket has fallen sufficiently deep below the strike. This is consistent with the exercise points visible in Figure 25, which are concentrated well below the strike.

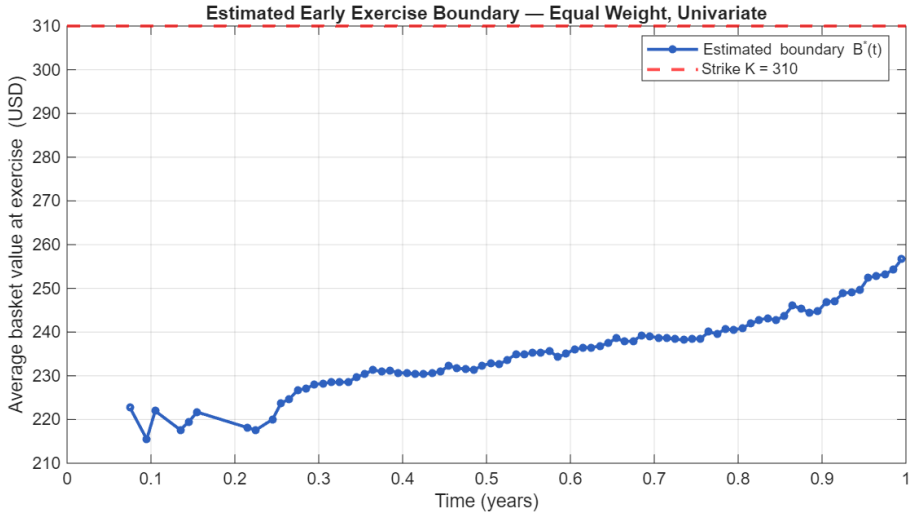


Figure 26: Estimated early exercise boundary $B^*(t)$ for the ATM equal-weight basket (univariate regression), computed as the mean basket value at exercise across all paths exercised at each time step. The notation $B^*(t)$ parallels the single-asset boundary $b^*(t)$ of Section 4, with the basket value B_t replacing the individual asset price S_t as the relevant state variable.

9 Conclusion

This thesis has examined the Least Squares Monte Carlo (LSM) method of Longstaff and Schwartz [2001] as a tool for pricing American-style options, moving from its theoretical foundations to a single-asset implementation, an extension to multi-asset basket options and an application calibrated on real equity data. Taken together, the results support a coherent picture of both the strengths and the practical boundaries of the method.

The theoretical chapters established that the difficulty of pricing American options lies not in simulating the underlying dynamics, but in approximating the continuation value that governs the optimal stopping decision. The LSM method addresses this by recasting the continuation value as a regression problem, which reduces the dimensionality of the computation and avoids the curse of dimensionality that limits finite-difference and tree-based methods in multi-factor settings. The numerical experiments carried out in this thesis are consistent with this theoretical picture. The implemented LSM estimator reproduces the expected lower-bound property relative to the European price and converges at the Monte Carlo rate predicted by theory as the number of simulated paths increases. It also exhibits the bias–variance trade-off in the number of basis functions described by Stentoft [2004], with a small number of regressors already sufficient to capture the relevant features of the exercise boundary, in line with the original recommendation of Longstaff and Schwartz [2001] and with the robustness results of Moreno and Navas [2003].

The extension to basket options preserved this overall behaviour while introducing a new methodological question: whether the continuation value regression should be specified on the basket value alone or on the full vector of constituent prices. A central finding of this thesis is that, for the basket options considered, a regression specification based solely on the basket value achieves pricing results comparable to those obtained from a substantially richer state representation based on the full vector of underlying asset prices, including in the empirical application calibrated on historical equity data. This points to a broader theme that recurs throughout the thesis: in regression-based Monte Carlo pricing, the binding constraint is rarely the amount of state information made available to the regression, but rather how efficiently that information is summarized. A parsimonious regression specification that captures the economically relevant driver of the payoff (here, the basket value itself) appears to perform as well as a specification with a much larger state space, while remaining computationally lighter and less exposed to estimation noise.

The analysis carried out here remains confined to a Geometric Brownian Motion setting with constant parameters, to baskets of modest size and to a specific family of basis functions; relaxing each of these restrictions defines a natural direction for further work. A natural extension concerns the dynamics of the underlying assets. Replacing the Geometric Brownian Motion with a stochastic volatility model such as Heston [1993], where variance follows its own mean-reverting diffusion, removes the closed-form simulation scheme used throughout this thesis and requires a numerical discretization (e.g. Euler–Maruyama) of the joint price-variance system, introducing an additional discretization bias. More importantly, it would test whether the parsimony of the univariate regression specification still holds when paths sharing the same basket value can differ in their instantaneous variance and thus in their continuation value.

This trade-off between the dimension of the regression and the information it captures is, ultimately, the central thread of this work: the most effective state-space representation for LSM is not necessarily the richest one available, but the one that retains what is relevant to the exercise decision.

A Matlab Code

A.1 Monte Carlo pricing of a European call option

This section reports the MATLAB script implementing the Monte Carlo estimator for the price of a European call option under the Black–Scholes model. The code is consistent with the theoretical framework developed in Section 3.2 and is included for reproducibility purposes.

Non-essential elements such as convergence diagnostics, intermediate outputs and graphical visualizations have been deliberately omitted in order to improve clarity and readability.

```
%% Model parameters
S0     = 120;   % Initial stock price
K      = 100;   % Strike price
r      = 0.05;  % Risk-free rate (continuously compounded)
sigma  = 0.2;   % Volatility
T      = 1.0;   % Maturity (years)

%% Black-Scholes closed-form price
d1   = (log(S0/K) + (r + 0.5*sigma^2)*T) / (sigma*sqrt(T));
d2   = d1 - sigma*sqrt(T);
C_BS = S0 * normcdf(d1) - K * exp(-r*T) * normcdf(d2);

%% Monte Carlo convergence analysis (first plot)
N_samples = round(logspace(2, 6, 30));
MC_est    = zeros(length(N_samples), 1);

rng(42);
Z_all = randn(max(N_samples), 1);

for k = 1:length(N_samples)
    n       = N_samples(k);
    Z       = Z_all(1:n);
    S_T     = S0 * exp((r - 0.5*sigma^2)*T + sigma*sqrt(T).*Z);
    payoff  = exp(-r*T) * max(S_T - K, 0);
    MC_est(k) = mean(payoff);
end

%% Monte Carlo estimation with confidence interval
N_final      = 1e6;
rng(99);
Z_final      = randn(N_final, 1);
S_T_final    = S0 * exp((r - 0.5*sigma^2)*T + sigma*sqrt(T).*Z_final);
payoff_final = exp(-r*T) * max(S_T_final - K, 0);

C_MC   = mean(payoff_final);
sC     = std(payoff_final, 0);
SE     = sC / sqrt(N_final);
z      = 1.96;
CI     = [C_MC - z*SE , C_MC + z*SE];

%% Empirical RMSE over independent replications (second plot)
k_rep = 200;
RMSE  = zeros(length(N_samples), 1);

```

```
rng(123);

for idx = 1:length(N_samples)
    n = N_samples(idx);
    errors_sq = zeros(k_rep, 1);
    for j = 1:k_rep
        Z_rep        = randn(n, 1);
        S_T_rep      = S0 * exp((r-0.5*sigma^2)*T + sigma*sqrt(T).*
           Z_rep);
        payoff_rep   = exp(-r*T) * max(S_T_rep - K, 0);
        C_hat        = mean(payoff_rep);
        errors_sq(j) = (C_hat - C_BS)^2;
    end
    RMSE(idx) = sqrt(mean(errors_sq));
end
```

Listing 1: Monte Carlo convergence analysis and final estimation for a European call option under the Black–Scholes model.

Code description. The analytical price computed via the Black–Scholes formula (Section 2.6) serves as a benchmark for all subsequent numerical comparisons.

Monte Carlo convergence analysis. To study convergence of the price estimator, 30 sample sizes n are generated logarithmically spaced between 10^2 and 10^6 using `logspace`. A single vector `Z_all` of 10^6 standard normal draws is generated once with a fixed seed (`rng(42)`) for reproducibility. For each n, the estimator uses the first n entries of `Z_all`: this nested construction ensures that larger samples are extensions of smaller ones, producing a more stable convergence curve and avoiding the extra noise that would arise from regenerating independent samples at each step. For each n, the terminal stock price is simulated via the exact solution (8), the discounted payoff is computed and the sample mean $\widehat{C}_n$ is stored.

Monte Carlo estimation with confidence interval. A separate run with $N = 10^6$ simulations and an independent seed (`rng(99)`) is used to produce the final point estimate $\widehat{C}_{\mathrm{MC}}$ and the asymptotic 95% confidence interval

$$\widehat{C}_n \pm 1.96 \, \frac{s_C}{\sqrt{n}},$$

where s_C is the sample standard deviation (9). A new seed is used here to ensure that this block is statistically independent from the nested samples generated in the convergence analysis above.

Empirical RMSE. The theoretical convergence rate $\mathcal{O}(n^{-1/2})$ [Glasserman, 2004] describes the decay of the standard deviation of the estimator error, not of a single realisation. To verify it empirically, the script estimates the Root Mean Square Error:

$$\widehat{\mathrm{RMSE}}(n) = \sqrt{\frac{1}{k} \sum_{j=1}^{k} \left(\widehat{C}_n^{(j)} - C_{\mathrm{BS}} \right)^2},$$

using $k = 200$ independent replications for each value of n. A separate seed (`rng(123)`) is used to keep this block statistically independent from the previous ones. For each

replication j, a new vector of n standard normal draws is generated, the terminal price and discounted payoff are computed and the squared error $(\widehat{C}_n^{(j)} - C_{\mathrm{BS}})^2$ is recorded. The RMSE is then obtained as the square root of the mean of these squared errors.

A.2 Covariance Structure in Multidimensional GBM

This appendix provides a detailed derivation of the covariance structure for the Multidimensional Geometric Brownian Motion defined in Section 6, showing how the Cholesky decomposition ensures that the covariance between log-returns matches the desired covariance matrix.

Proposition 5. *If the asset prices* $\{S_t^{(j)}\}_{t\geq 0}$, $j = 1, \ldots, d$, *satisfy*

$$\mathrm{d}S_t^{(j)} = rS_t^{(j)}\,\mathrm{d}t + S_t^{(j)} \sum_{k=1}^{d} a_{jk}\,\mathrm{d}W_t^{(k)}, \qquad j = 1, \ldots, d, \tag{57}$$

where $\mathbf{W}_t$ *has independent components and* $\Sigma = AA^\top$, *then*

$$\mathrm{Cov}\Big[\mathrm{d}\log S_t^{(i)},\, \mathrm{d}\log S_t^{(j)}\Big] = \Sigma_{ij}\,\mathrm{d}t.$$

Proof. Applying Itô's Lemma to $f(S) = \log S$, with $f'(S) = 1/S$ and $f''(S) = -1/S^2$,

$$\mathrm{d}\log S_t^{(j)} = \frac{1}{S_t^{(j)}}\,\mathrm{d}S_t^{(j)} - \frac{1}{2(S_t^{(j)})^2}\,(\mathrm{d}S_t^{(j)})^2.$$

Substituting (57) into the first term yields

$$r\,\mathrm{d}t + \sum_{k=1}^{d} a_{jk}\,\mathrm{d}W_t^{(k)}.$$

For the quadratic variation term, expanding the square of (57) and using the Itô multiplication table introduced in Section 2.2,

$$(\mathrm{d}t)^2 = 0, \qquad \mathrm{d}t\,\mathrm{d}W_t^{(k)} = 0, \qquad \mathrm{d}W_t^{(k)}\,\mathrm{d}W_t^{(\ell)} = \delta_{k\ell}\,\mathrm{d}t,$$

only the diffusion term contributes, giving

$$(\mathrm{d}S_t^{(j)})^2 = (S_t^{(j)})^2 \sum_{k=1}^{d}\sum_{\ell=1}^{d} a_{jk}\,a_{j\ell}\,\mathrm{d}W_t^{(k)}\,\mathrm{d}W_t^{(\ell)} = (S_t^{(j)})^2 \sum_{k=1}^{d} a_{jk}^2\,\mathrm{d}t,$$

where the last equality follows from $\mathrm{d}W_t^{(k)}\,\mathrm{d}W_t^{(\ell)} = \delta_{k\ell}\,\mathrm{d}t$, so that only the diagonal terms $k = \ell$ contribute. The second term in Itô's Lemma therefore becomes

$$-\frac{1}{2(S_t^{(j)})^2}\,(\mathrm{d}S_t^{(j)})^2 = -\frac{1}{2}\sum_{k=1}^{d} a_{jk}^2\,\mathrm{d}t,$$

and combining both contributions

$$\mathrm{d}\log S_t^{(j)} = \left(r - \frac{1}{2}\sum_{k=1}^{d} a_{jk}^2\right)\mathrm{d}t + \sum_{k=1}^{d} a_{jk}\,\mathrm{d}W_t^{(k)}. \tag{58}$$

Since the drift in (58) is deterministic,

$$\mathbb{E}[\mathrm{d}\log S_t^{(j)}] = \left(r - \frac{1}{2}\sum_{k=1}^{d} a_{jk}^2\right)\mathrm{d}t,$$

subtracting the mean from both sides of (58) gives

$$\mathrm{d}\log S_t^{(j)} - \mathbb{E}[\mathrm{d}\log S_t^{(j)}] = \sum_{k=1}^{d} a_{jk}\,\mathrm{d}W_t^{(k)},$$

and analogously for asset i. The covariance is therefore

$$\begin{aligned}\mathrm{Cov}\Big[\mathrm{d}\log S_t^{(i)},\,\mathrm{d}\log S_t^{(j)}\Big] &= \mathbb{E}\left[\Big(\sum_{k=1}^{d} a_{ik}\,\mathrm{d}W_t^{(k)}\Big)\Big(\sum_{\ell=1}^{d} a_{j\ell}\,\mathrm{d}W_t^{(\ell)}\Big)\right] \\ &= \sum_{k=1}^{d}\sum_{\ell=1}^{d} a_{ik}\,a_{j\ell}\,\mathbb{E}\Big[\mathrm{d}W_t^{(k)}\,\mathrm{d}W_t^{(\ell)}\Big] \\ &= \sum_{k=1}^{d} a_{ik}\,a_{jk}\,\mathrm{d}t,\end{aligned}$$

where the last equality uses $\mathbb{E}[\mathrm{d}W_t^{(k)}\,\mathrm{d}W_t^{(\ell)}] = \delta_{k\ell}\,\mathrm{d}t$ once more, so only the diagonal terms $k = \ell$ survive.

The sum $\sum_{k=1}^{d} a_{ik}\,a_{jk}$ is the (i,j)-th entry of $AA^\top$, so

$$\mathrm{Cov}\Big[\mathrm{d}\log S_t^{(i)},\,\mathrm{d}\log S_t^{(j)}\Big] = (AA^\top)_{ij}\,\mathrm{d}t = \Sigma_{ij}\,\mathrm{d}t,$$

which completes the proof. □

Computational Implications. The key insight from this derivation is the role of the independence structure of the Brownian motions. Starting from d independent standard Brownian motions $\{W_t^{(1)}, \ldots, W_t^{(d)}\}$, the linear transformation

$$\tilde{W}_t^{(j)} := \sum_{k=1}^{d} a_{jk} W_t^{(k)}, \quad j = 1, \ldots, d, \tag{59}$$

produces correlated Brownian motions satisfying $\mathbb{E}[\mathrm{d}\tilde{W}_t^{(i)}\,\mathrm{d}\tilde{W}_t^{(j)}] = \Sigma_{ij}\,\mathrm{d}t$, reproducing exactly the prescribed covariance structure. This construction is particularly advantageous for Monte Carlo simulation. The Cholesky decomposition is numerically stable whenever Σ is positive definite. The lower-triangular structure of A contains only $d(d+1)/2$ non-zero entries, approximately halving the number of multiplications per time step compared to a full matrix transformation, which is significant for large d. Finally, independent random draws can be generated first with the correlation structure introduced subsequently via matrix multiplication, simplifying both implementation and testing.

A.3 LSM Pricing of an American Put Option

This section reports the complete MATLAB implementation of the Longstaff–Schwartz algorithm for pricing American put options, as described in Chapter 5. The code is organised into a main script and three functions: `lsm_american_put`, `lsm_american_put_outofsample` and `basis_function`. Non-essential elements such as convergence diagnostics, sensitivity analyses and graphical visualizations have been omitted for brevity.

```
clc; clear;

%% Parameters
K       = 100;
r       = 0.05;
sigma   = 0.2;
T       = 1;
Nsteps  = 100;
Npaths  = 10000;
M       = 4;
seed    = 123;
basis   = 'Polynomial';   % 'Polynomial', 'Hermite', or 'Chebyshev'

S0_vec          = [80 90 100 110 120];
put_A_prices    = zeros(size(S0_vec));
put_E_prices    = zeros(size(S0_vec));
early_frac      = zeros(size(S0_vec));
results_struct_all = cell(size(S0_vec));
exercised_time_all = cell(size(S0_vec));

%% Chebyshev pre-processing (fixed bounds across all dates and S0)
x_min_cheb = [];
x_max_cheb = [];
if strcmpi(basis, 'Chebyshev')
    rng(seed);
    dt_tmp  = T / Nsteps;
    Nhalf   = Npaths / 2;
    Z_tmp   = randn(Nhalf, Nsteps);
    Z_tmp   = [Z_tmp; -Z_tmp];
    inc_tmp = (r - 0.5*sigma^2)*dt_tmp + sigma*sqrt(dt_tmp).*Z_tmp;
    S_tmp   = [min(S0_vec)*ones(Npaths,1), zeros(Npaths, Nsteps)];
    for tt = 1:Nsteps
        S_tmp(:,tt+1) = S_tmp(:,tt) .* exp(inc_tmp(:,tt));
    end
    x_min_cheb = min(S_tmp(:)) / K;
    x_max_cheb = max(S_tmp(:)) / K;
    clear S_tmp Z_tmp inc_tmp dt_tmp Nhalf tt
end

%% Pricing loop over initial stock prices
for i = 1:length(S0_vec)
    params.S0         = S0_vec(i);
    params.K          = K;
    params.r          = r;
    params.sigma      = sigma;
    params.T          = T;
    params.Nsteps     = Nsteps;
    params.Npaths     = Npaths;
    params.M          = M;
    params.seed       = seed;
```

```
    params.basis      = basis;
    params.antithetic = true;
    params.x_min_cheb = x_min_cheb;
    params.x_max_cheb = x_max_cheb;

    [V0, se, stats, results_struct] = lsm_american_put(params);
    put_A_prices(i)      = V0;
    early_frac(i)        = stats.frac_exercised;
    put_E_prices(i)      = european_put_bs(params.S0, K, r, sigma, T);
    results_struct_all{i} = results_struct;
    exercised_time_all{i} = results_struct.exercised_time;
end

EEV = put_A_prices - put_E_prices;

results_table = table(S0_vec', put_A_prices', put_E_prices', ...
    EEV', early_frac', ...
    'VariableNames', {'S0','AmericanPut','EuropeanPut', ...
                      'EEV','EarlyExerciseFraction'});
disp(results_table);

%% Out-of-sample validation
nS       = length(S0_vec);
Vals_in  = zeros(nS,1);  SEs_in  = zeros(nS,1);  FracEx_in  = zeros(nS
    ,1);
Vals_out = zeros(nS,1);  SEs_out = zeros(nS,1);  FracEx_out = zeros(nS
    ,1);

for i = 1:nS
    params_in       = params;
    params_in.S0    = S0_vec(i);
    params_in.seed  = 123;

    params_out      = params;
    params_out.S0   = S0_vec(i);
    params_out.seed = 999;

    [Vals_in(i), SEs_in(i), stats_in, regModel] = ...
        lsm_american_put(params_in);
    FracEx_in(i) = stats_in.frac_exercised;

    [Vals_out(i), SEs_out(i), stats_out] = ...
        lsm_american_put_outofsample(params_out, regModel);
    FracEx_out(i) = stats_out.frac_exercised;
end

for i = 1:nS
    fprintf(['S0=%6.1f | In-sample: %.4f +/- %.4f ' ...
             '(early=%5.2f%%) | Out-sample: %.4f +/- %.4f ' ...
             '(early=%5.2f%%)\n'], ...
        S0_vec(i), Vals_in(i), SEs_in(i), 100*FracEx_in(i), ...
        Vals_out(i), SEs_out(i), 100*FracEx_out(i));
end
```

Listing 2: Main script: parameter setup, pricing loop and out-of-sample validation.

```
function [V0, se, stats, results_struct] = lsm_american_put(p)

```

```
    S0 = p.S0; K = p.K; r = p.r; sigma = p.sigma;
    T = p.T; Nsteps = p.Nsteps; Npaths = p.Npaths;
    M = p.M; basisType = p.basis;

    x_min_cheb = [];
    x_max_cheb = [];
    if isfield(p, 'x_min_cheb'), x_min_cheb = p.x_min_cheb; end
    if isfield(p, 'x_max_cheb'), x_max_cheb = p.x_max_cheb; end

    if isfield(p,'seed'), rng(p.seed); end

    if p.antithetic && mod(Npaths,2) ~= 0
        error('Npaths must be even when using antithetic variates.');
    end

    dt   = T / Nsteps;
    disc = exp(-r*dt);

    %% 1. Simulate stock paths (exact GBM discretization)
    if p.antithetic
        Nhalf  = Npaths / 2;
        Z_half = randn(Nhalf, Nsteps);
        Z      = [Z_half; -Z_half];
    else
        Z = randn(Npaths, Nsteps);
    end

    increments = (r - 0.5*sigma^2)*dt + sigma*sqrt(dt).*Z;
    S = [S0*ones(Npaths,1), zeros(Npaths, Nsteps)];
    for t = 1:Nsteps
        S(:,t+1) = S(:,t) .* exp(increments(:,t));
    end

    %% 2. Payoff matrix
    payoff   = max(K - S, 0);
    cashflow = payoff(:,end);
    exercised_time = (Nsteps+1)*ones(Npaths,1);

    beta_coeff       = cell(Nsteps,1);
    cont_value_avg   = nan(Nsteps,1);
    continuation_value = nan(Npaths, Nsteps);

    %% 3. Backward induction
    for t = Nsteps:-1:2
        cashflow  = cashflow * disc;
        itm       = payoff(:,t) > 0;
        idx_itm   = find(itm);
        if isempty(idx_itm), continue; end

        X   = S(idx_itm,t) / K;
        Y   = cashflow(idx_itm) / K;
        Phi = basis_function(X, M, basisType, x_min_cheb, x_max_cheb);

        beta               = Phi \ Y;
        beta_coeff{t}      = beta;
        cont               = Phi * beta * K;
        cont_value_avg(t) = mean(cont);
        continuation_value(idx_itm,t) = cont;
```

```
61
62          exercise_now    = payoff(idx_itm,t) > cont;
63          exc_idx_global  = idx_itm(exercise_now);
64          cashflow(exc_idx_global)      = payoff(exc_idx_global,t);
65          exercised_time(exc_idx_global) = t;
66      end
67
68      %% 4. Discount to time 0
69      cashflow = cashflow * disc;
70      V0 = mean(cashflow);
71      se = std(cashflow) / sqrt(Npaths);
72      stats.frac_exercised = mean(exercised_time < (Nsteps+1));
73
74      %% 5. Output structure
75      results_struct = struct( ...
76          'exercised_time',    exercised_time,    ...
77          'S',                 S,                 ...
78          'payoff',            payoff,            ...
79          'beta_coeff',        {beta_coeff},      ...
80          'Nsteps',            Nsteps,            ...
81          'T',                 T,                 ...
82          'M',                 M,                 ...
83          'basis',             basisType,         ...
84          'K',                 K,                 ...
85          'antithetic',        p.antithetic,      ...
86          'V0',                V0,                ...
87          'se',                se,                ...
88          'stats',             stats,             ...
89          'cont_value_avg',    cont_value_avg,    ...
90          'continuation_value',continuation_value,...
91          'x_min_cheb',        x_min_cheb,        ...
92          'x_max_cheb',        x_max_cheb);
93  end
```

Listing 3: Function `lsm_american_put`: LSM backward induction for American put pricing.

```
1   function [V0, se, stats] = lsm_american_put_outofsample(p, regModel)
2
3       S0 = p.S0; K = p.K; r = p.r; sigma = p.sigma;
4       T = p.T; Nsteps = p.Nsteps; Npaths = p.Npaths;
5       M = p.M; basisType = p.basis;
6
7       dt   = T / Nsteps;
8       disc = exp(-r*dt);
9
10      x_min_cheb = [];
11      x_max_cheb = [];
12      if isfield(regModel,'x_min_cheb'), x_min_cheb = regModel.x_min_cheb
            ; end
13      if isfield(regModel,'x_max_cheb'), x_max_cheb = regModel.x_max_cheb
            ; end
14
15      if isfield(p,'seed'), rng(p.seed); end
16
17      useAntithetic = false;
18      if isfield(regModel,'antithetic') && regModel.antithetic
19          useAntithetic = true;
```

```
20          if mod(Npaths,2) ~= 0
21              error('Npaths must be even when using antithetic variates.'
                    );
22          end
23      end
24
25      %% 1. Simulate out-of-sample paths
26      if useAntithetic
27          Nhalf  = Npaths / 2;
28          Z_half = randn(Nhalf, Nsteps);
29          Z      = [Z_half; -Z_half];
30      else
31          Z = randn(Npaths, Nsteps);
32      end
33
34      increments = (r - 0.5*sigma^2)*dt + sigma*sqrt(dt).*Z;
35      S = [S0*ones(Npaths,1), zeros(Npaths, Nsteps)];
36      for t = 1:Nsteps
37          S(:,t+1) = S(:,t) .* exp(increments(:,t));
38      end
39
40      %% 2. Payoff matrix
41      payoff   = max(K - S, 0);
42      cashflow = payoff(:,end);
43      exercised_time = (Nsteps+1)*ones(Npaths,1);
44
45      %% 3. Backward induction with stored coefficients
46      for t = Nsteps:-1:2
47          cashflow = cashflow * disc;
48          itm      = payoff(:,t) > 0;
49          idx_itm  = find(itm);
50          if isempty(idx_itm), continue; end
51
52          X_out   = S(idx_itm,t) / K;
53          Phi_out = basis_function(X_out, M, basisType, ...
54                                   x_min_cheb, x_max_cheb);
55
56          if ~iscell(regModel.beta_coeff) || ...
57                  isempty(regModel.beta_coeff{t})
58              continue;
59          end
60          beta = regModel.beta_coeff{t};
61
62          cont           = Phi_out * beta * K;
63          exercise_now   = payoff(idx_itm,t) > cont;
64          exc_idx_global = idx_itm(exercise_now);
65          cashflow(exc_idx_global)       = payoff(exc_idx_global,t);
66          exercised_time(exc_idx_global) = t;
67      end
68
69      %% 4. Price and standard error
70      cashflow = cashflow * disc;
71      V0 = mean(cashflow);
72      se = std(cashflow) / sqrt(Npaths);
73      stats.frac_exercised = mean(exercised_time < (Nsteps+1));
74  end
```

Listing 4: Function `lsm_american_put_outofsample`: out-of-sample valuation using stored regression coefficients.

```
function Phi = basis_function(x, M, type, x_min, x_max)

    x = x(:);
    n = numel(x);
    Phi = zeros(n, M);

    switch lower(type)

        case 'laguerre'
            L      = zeros(n, M);
            L(:,1) = ones(n,1);
            if M > 1, L(:,2) = 1 - x; end
            for j = 2:M-1
                L(:,j+1) = ((2*j-1-x).*L(:,j) - (j-1)*L(:,j-1)) / j;
            end
            w = exp(-x/2);
            for j = 1:M
                Phi(:,j) = w .* L(:,j);
            end

        case 'hermite'
            Phi(:,1) = ones(n,1);
            if M > 1, Phi(:,2) = x; end
            for j = 2:M-1
                Phi(:,j+1) = x.*Phi(:,j) - (j-1)*Phi(:,j-1);
            end

        case 'polynomial'
            for j = 0:M-1
                Phi(:,j+1) = x .^ j;
            end

        case 'chebyshev'
            if nargin < 4 || isempty(x_min) || isempty(x_max)
                error('Chebyshev requires x_min and x_max.');
            end
            if x_max <= x_min
                error('x_max must be strictly greater than x_min.');
            end
            x_scaled = 2*(x - x_min)/(x_max - x_min) - 1;
            x_scaled = max(-1, min(1, x_scaled));
            Phi(:,1) = ones(n,1);
            if M > 1, Phi(:,2) = x_scaled; end
            for j = 3:M
                Phi(:,j) = 2*x_scaled.*Phi(:,j-1) - Phi(:,j-2);
            end

        otherwise
            error(['Unknown basis type: %s. ' ...
                   'Choose: laguerre, hermite, ' ...
                   'polynomial, chebyshev.'], type);
    end
end
```

Listing 5: Function `basis_function`: evaluation of the regression basis (Polynomial, Laguerre, Hermite, Chebyshev).

```
function price = european_put_bs(S0, K, r, sigma, T)
    d1    = (log(S0/K) + (r + 0.5*sigma^2)*T) / (sigma*sqrt(T));
    d2    = d1 - sigma*sqrt(T);
    price = K*exp(-r*T)*normcdf(-d2) - S0*normcdf(-d1);
end
```

Listing 6: Function `european_put_bs`: Black–Scholes closed-form price for a European put option.

A.4 LSM Pricing of an American Basket Put Option

This section reports the MATLAB implementation of the Longstaff–Schwartz algorithm for pricing American basket put options, as described in Chapter 6. The code is organised into four functions: `make_params`, which assembles the parameter structure; `build_uniform_rho`, which constructs a uniform correlation matrix; `lsm_basket_uni`, which implements the univariate regression on the scalar basket value; and `lsm_basket_multi`, which implements the multivariate regression on the full asset price vector.

```
function params = make_params(r, T, K, Nsteps, Npaths, seed, ...
                              d, S0, w, sigma, A, antithetic)
    params.r          = r;
    params.T          = T;
    params.K          = K;
    params.Nsteps     = Nsteps;
    params.Npaths     = Npaths;
    params.seed       = seed;
    params.d          = d;
    params.S0         = S0;
    params.w          = w;
    params.sigma      = sigma;
    params.A          = A;
    params.antithetic = antithetic;
end

function rho = build_uniform_rho(d, rho_val)
    rho = rho_val * ones(d) + (1 - rho_val) * eye(d);
end
```

Listing 7: Helper functions: parameter structure assembly and uniform correlation matrix construction.

```
function [V0, se, stats] = lsm_basket_uni(p)

    r = p.r; T = p.T; K = p.K;
    Nsteps = p.Nsteps; Npaths = p.Npaths;
    d = p.d; S0 = p.S0; w = p.w;
    sigma = p.sigma; A = p.A;

    if isfield(p, 'seed'), rng(p.seed); end
    if p.antithetic && mod(Npaths, 2) ~= 0
        error('Npaths must be even when using antithetic variates.');
    end

    dt   = T / Nsteps;
    disc = exp(-r * dt);

    %% 1. Simulate correlated GBM paths
    S = zeros(Npaths, d, Nsteps+1);
    S(:,:,1) = repmat(S0, Npaths, 1);

    if p.antithetic
        Nhalf  = Npaths / 2;
        Z_half = randn(Nhalf, d, Nsteps);
        Z      = cat(1, Z_half, -Z_half);
    else
        Z = randn(Npaths, d, Nsteps);
    end
```

```
27
28      for t = 1:Nsteps
29          Zcorr      = Z(:,:,t) * A';
30          S(:,:,t+1) = S(:,:,t) .* exp( ...
31              (r - 0.5*sigma.^2)*dt + sqrt(dt)*Zcorr);
32      end
33
34      %% 2. Basket value and payoff
35      Basket = zeros(Npaths, Nsteps+1);
36      for t = 1:Nsteps+1
37          Basket(:,t) = S(:,:,t) * w;
38      end
39
40      payoff         = max(K - Basket, 0);
41      cashflow       = payoff(:,end);
42      exercised_time = (Nsteps+1) * ones(Npaths, 1);
43
44      %% 3. Backward induction
45      for t = Nsteps:-1:2
46          cashflow = cashflow * disc;
47
48          idx = find(payoff(:,t) > 0);
49          if isempty(idx), continue; end
50
51          % Quadratic basis in normalised basket value B_t/K
52          Y   = cashflow(idx) / K;
53          X   = Basket(idx, t) / K;
54          Phi = [ones(size(X)), X, X.^2];
55
56          beta = Phi \ Y;
57          C    = Phi * beta * K;
58
59          ex_mask  = payoff(idx, t) > C;
60          exc_idx  = idx(ex_mask);
61          cashflow(exc_idx)       = payoff(exc_idx, t);
62          exercised_time(exc_idx) = t;
63      end
64
65      %% 4. Discount to time 0 and compute price
66      cashflow = cashflow * disc;
67      V0 = mean(cashflow);
68      se = std(cashflow) / sqrt(Npaths);
69
70      stats.frac_exercised = mean(exercised_time < (Nsteps+1));
71      stats.exercised_time = exercised_time;
72      stats.Basket         = Basket;
73  end
```

Listing 8: Function `lsm_basket_uni`: LSM backward induction with univariate regression on the basket value B_t.

```
1   function [V0, se, stats] = lsm_basket_multi(p)
2
3       r = p.r; T = p.T; K = p.K;
4       Nsteps = p.Nsteps; Npaths = p.Npaths;
5       d = p.d; S0 = p.S0; w = p.w;
6       sigma = p.sigma; A = p.A;
7
```

```
    if isfield(p, 'seed'), rng(p.seed); end
    if p.antithetic && mod(Npaths, 2) ~= 0
        error('Npaths must be even when using antithetic variates.');
    end

    dt   = T / Nsteps;
    disc = exp(-r * dt);

    %% 1. Simulate correlated GBM paths
    S = zeros(Npaths, d, Nsteps+1);
    S(:,:,1) = repmat(S0, Npaths, 1);

    if p.antithetic
        Nhalf  = Npaths / 2;
        Z_half = randn(Nhalf, d, Nsteps);
        Z      = cat(1, Z_half, -Z_half);
    else
        Z = randn(Npaths, d, Nsteps);
    end

    for t = 1:Nsteps
        Zcorr      = Z(:,:,t) * A';
        S(:,:,t+1) = S(:,:,t) .* exp( ...
            (r - 0.5*sigma.^2)*dt + sqrt(dt)*Zcorr);
    end

    %% 2. Basket value and payoff
    Basket = zeros(Npaths, Nsteps+1);
    for t = 1:Nsteps+1
        Basket(:,t) = S(:,:,t) * w;
    end

    payoff         = max(K - Basket, 0);
    cashflow       = payoff(:,end);
    exercised_time = (Nsteps+1) * ones(Npaths, 1);

    %% 3. Backward induction
    for t = Nsteps:-1:2
        cashflow = cashflow * disc;

        idx = find(payoff(:,t) > 0);
        if isempty(idx), continue; end

        Y   = cashflow(idx) / K;
        % Normalised individual asset prices: n_itm x d
        S_t = reshape(S(idx,:,t), length(idx), d) / K;
        % Linear-quadratic basis without cross terms: 2d+1 regressors
        Phi = [ones(length(idx), 1), S_t, S_t.^2];

        beta = Phi \ Y;
        C    = Phi * beta * K;

        ex_mask  = payoff(idx, t) > C;
        exc_idx  = idx(ex_mask);
        cashflow(exc_idx)       = payoff(exc_idx, t);
        exercised_time(exc_idx) = t;
    end

```

```
66      %% 4. Discount to time 0 and compute price
67      cashflow = cashflow * disc;
68      V0 = mean(cashflow);
69      se = std(cashflow) / sqrt(Npaths);
70
71      stats.frac_exercised = mean(exercised_time < (Nsteps+1));
72      stats.exercised_time = exercised_time;
73      stats.Basket         = Basket;
74  end
```

Listing 9: Function `lsm_basket_multi`: LSM backward induction with multivariate regression on the full asset price vector $\mathbf{S}_t$. The basis is linear-quadratic without cross terms, yielding $2d + 1$ regressors.

A.5 Empirical Application: Big Tech Basket — Main Script

This section presents the main script for the empirical application of Chapter 8. The helper functions `make_params`, `build_uniform_rho`, `lsm_basket_uni` and `lsm_basket_multi` are listed in Section A.4 and are not repeated here. The script is organised into the following blocks: configuration and data import, the analysis sections and the two auxiliary functions specific to this script.

```
%% STEP 0     CONFIGURATION
% Approximate market capitalisations (bn USD, May 2026)
% Order: Amazon | Apple | Microsoft | NVIDIA
mktcap_approx = [2950, 4590, 3340, 5150];

r      = 0.038;   % risk-free rate: 1Y US Treasury yield (31-May-2026)
T      = 1;       % maturity (years)
Nsteps = 100;     % exercise dates
Npaths = 50000;   % Monte Carlo paths
seed   = 123;

%% STEP 1     READ DATA FROM EXCEL
dataDir   = 'C:\...\DATASET';
excelFile = fullfile(dataDir, 'Big_Tech_Data.xlsx');

sheets  = {'Amazon', 'Apple', 'Microsoft', 'NVIDIA'};
tickers = {'AMZN',   'AAPL',  'MSFT',      'NVDA'};
d       = numel(sheets);

prices_cell = cell(d, 1);

for i = 1:d
    Tbl   = readtable(excelFile, 'Sheet', sheets{i}, ...
                      'VariableNamingRule', 'preserve');
    price = Tbl{:, 2};   % column B = Price
    % Prices stored as strings with decimal comma (321,02)
    if iscell(price) || isstring(price)
        price = string(price);
        price = strrep(price, ',', '.');
        price = str2double(price);
    end
    price = price(~isnan(price));
    prices_cell{i} = flipud(price(:));   % chronological order
end

%% STEP 2     ALIGN ON COMMON TRADING WINDOW
lengths  = cellfun(@numel, prices_cell);
N_common = min(lengths);

for i = 1:d
    prices_cell{i} = prices_cell{i}(end - N_common + 1 : end);
end

P = cell2mat(prices_cell');   % N_common x d
```

Listing 10: Configuration, data import from Excel, and alignment on the common trading window.

```
%% STEP 3     PARAMETER CALIBRATION
log_ret   = diff(log(P));
```

```
sigma_vec = std(log_ret)  * sqrt(252);   % annualized volatilities
rho_mat   = corr(log_ret);               % historical correlation
    matrix
Sigma_mat = diag(sigma_vec) * rho_mat * diag(sigma_vec);
A         = chol(Sigma_mat, 'lower');    % lower Cholesky factor

S0_real = P(end, :);
K       = round(mean(S0_real) / 10) * 10;   % ATM strike (rounded)

%% STEP 4     BASKET WEIGHTS
w_equal = ones(d, 1) / d;
w_value = (mktcap_approx / sum(mktcap_approx))';

sigma_basket_eq = sqrt(w_equal' * Sigma_mat * w_equal);
sigma_basket_vw = sqrt(w_value' * Sigma_mat * w_value);

%% STEP 5     BUILD PARAMETER STRUCTS
params_eq = make_params(r, T, K, Nsteps, Npaths, seed, d, ...
                        S0_real, w_equal, sigma_vec, A, true);
params_vw = make_params(r, T, K, Nsteps, Npaths, seed, d, ...
                        S0_real, w_value, sigma_vec, A, true);
```

Listing 11: Parameter calibration from historical log-returns, basket weight construction, and parameter struct assembly.

```
%% BASELINE PRICES
[V_eq_uni,   se_eq_uni,   stats_eq_uni  ] = lsm_basket_uni(params_eq);
[V_eq_multi, se_eq_multi, stats_eq_multi] = lsm_basket_multi(params_eq)
    ;
[V_vw_uni,   se_vw_uni,   stats_vw_uni  ] = lsm_basket_uni(params_vw);
[V_vw_multi, se_vw_multi, stats_vw_multi] = lsm_basket_multi(params_vw)
    ;

%% MONEYNESS SENSITIVITY  (equal weight)
% S0 scaled uniformly across all assets: 70% to 130% of S0_real
scale_vec = [0.70, 0.80, 0.90, 1.00, 1.10, 1.20, 1.30];
nMoney    = length(scale_vec);

V_uni_mon    = zeros(nMoney, 1);   se_uni_mon    = zeros(nMoney, 1);
V_multi_mon  = zeros(nMoney, 1);   se_multi_mon  = zeros(nMoney, 1);
eef_uni_mon  = zeros(nMoney, 1);   eef_multi_mon = zeros(nMoney, 1);

for i = 1:nMoney
    p    = params_eq;
    p.S0 = S0_real * scale_vec(i);
    [V_uni_mon(i),   se_uni_mon(i),   st_u] = lsm_basket_uni(p);
    [V_multi_mon(i), se_multi_mon(i), st_m] = lsm_basket_multi(p);
    eef_uni_mon(i)   = st_u.frac_exercised;
    eef_multi_mon(i) = st_m.frac_exercised;
end
```

Listing 12: Baseline prices for univariate and multivariate LSM under equal and value weighting, and moneyness sensitivity obtained by a uniform scaling of the initial asset prices.

```
%% EQUAL WEIGHT vs VALUE WEIGHT ACROSS MONEYNESS
V_vw_uni_mon   = zeros(nMoney, 1);   se_vw_uni_mon   = zeros(nMoney, 1)
    ;
```

```
V_vw_multi_mon = zeros(nMoney, 1);   se_vw_multi_mon = zeros(nMoney, 1)
    ;

for i = 1:nMoney
    p    = params_vw;
    p.S0 = S0_real * scale_vec(i);
    [V_vw_uni_mon(i),   se_vw_uni_mon(i)  ] = lsm_basket_uni(p);
    [V_vw_multi_mon(i), se_vw_multi_mon(i)] = lsm_basket_multi(p);
end

%% CORRELATION SENSITIVITY
% Uniform grid; historical average used as reference marker only.
off_diag     = ~logical(eye(d));
rho_hist_avg = mean(rho_mat(off_diag));

rho_grid = [0, 0.20, 0.40, 0.60, 0.80];
nCorr    = length(rho_grid);

V_uni_corr   = zeros(nCorr, 1);   se_uni_corr   = zeros(nCorr, 1);
V_multi_corr = zeros(nCorr, 1);   se_multi_corr = zeros(nCorr, 1);
delta_corr   = zeros(nCorr, 1);

for i = 1:nCorr
    rho_i   = build_uniform_rho(d, rho_grid(i));
    Sigma_i = diag(sigma_vec) * rho_i * diag(sigma_vec);
    A_i     = chol(Sigma_i, 'lower');
    p       = params_eq;
    p.A     = A_i;
    [V_uni_corr(i),   se_uni_corr(i),  ~] = lsm_basket_uni(p);
    [V_multi_corr(i), se_multi_corr(i),~] = lsm_basket_multi(p);
    delta_corr(i) = V_multi_corr(i) - V_uni_corr(i);
end
```

Listing 13: Equal-weight versus value-weight comparison across moneyness levels, and correlation sensitivity on a uniform grid $\rho \in \{0, 0.20, 0.40, 0.60, 0.80\}$, with the historical average marked as a reference.

```
%% EARLY EXERCISE ANALYSIS  (ATM, equal weight)
tau_uni   = stats_eq_uni.exercised_time;
tau_multi = stats_eq_multi.exercised_time;

tau_uni_yr   = tau_uni   * (T / Nsteps);
tau_multi_yr = tau_multi * (T / Nsteps);

early_uni   = tau_uni_yr(tau_uni   < Nsteps+1);
early_multi = tau_multi_yr(tau_multi < Nsteps+1);

eef_uni_pct   = 100 * numel(early_uni)   / Npaths;
eef_multi_pct = 100 * numel(early_multi) / Npaths;
```

Listing 14: Early exercise analysis at the ATM level under equal weighting, using exercise times stored in `stats_eq_uni` and `stats_eq_multi` from the baseline pricing step.

```
%% ESTIMATED EARLY EXERCISE BOUNDARY
boundary = nan(Nsteps, 1);
dt       = T / Nsteps;

for t = 1:Nsteps
```

```
    idx_ex = find(tau_uni == t);
    if ~isempty(idx_ex)
        boundary(t) = mean(stats_eq_uni.Basket(idx_ex, t));
    end
end

%% SEED ROBUSTNESS OF DeltaV ACROSS MONEYNESS
% Verifies that the sign and magnitude of DeltaV = V_multi - V_uni
% are stable across three independent seeds.
seeds_rob    = [42, 123, 999];
nSeeds       = length(seeds_rob);
DeltaV_seeds = zeros(nMoney, nSeeds);

for s = 1:nSeeds
    for i = 1:nMoney
        p      = params_eq;
        p.S0   = S0_real * scale_vec(i);
        p.seed = seeds_rob(s);
        [V_u, ~] = lsm_basket_uni(p);
        [V_m, ~] = lsm_basket_multi(p);
        DeltaV_seeds(i, s) = V_m - V_u;
    end
end
```

Listing 15: Empirical early exercise boundary, estimated as the mean basket value over paths exercising at each step, and seed robustness of $\Delta V = V_{\text{multi}} - V_{\text{uni}}$ across moneyness levels for three independent seeds.

References

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, 1964.

Patrick Billingsley. *Probability and Measure*. John Wiley & Sons, New York, 3rd edition, 1995.

Tomas Björk. *Arbitrage Theory in Continuous Time*. Oxford University Press, 3 edition, 2009.

Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

Bruno Bouchard and Nizar Touzi. Discrete-time approximation and monte carlo simulation of backward stochastic differential equations. *Stochastic Processes and their Applications*, 111(2):175–206, 2004. doi: 10.1016/j.spa.2004.01.001.

Michael J. Brennan and Eduardo S. Schwartz. The valuation of American put options. *Journal of Finance*, 32(2):449–462, 1977. doi: 10.2307/2326779.

Marek Capinski and Tomasz Zastawniak. *Mathematics for Finance: An Introduction to Financial Engineering*. Springer Undergraduate Mathematics Series. Springer, London, 2 edition, 2011. ISBN 978-0-85729-081-6.

Eric Clément, Damien Lamberton, and Philip Protter. An analysis of a least squares regression method for american option pricing. *Finance and Stochastics*, 6(4):449–471, 2002.

John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979. doi: 10.1016/0304-405X(79)90015-1.

Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 5th edition, 2019.

Paul Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer, New York, 2004.

Paul Glasserman and Bin Yu. Number of paths versus number of basis functions in american option pricing. *Annals of Applied Probability*, 14(4):2090–2119, 2004.

Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2):327–343, 1993. doi: 10.1093/rfs/6.2.327.

John C. Hull. *Options, Futures, and Other Derivatives*. Pearson, 11 edition, 2021.

Investing.com. Historical price data. `https://www.investing.com`, 2026. Accessed: May 2026.

Ioannis Karatzas and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer, New York, 2 edition, 1991.

Yue-Kuen Kwok. *Mathematical Models of Financial Derivatives*. Springer, 2nd edition, 2008.

Francis A. Longstaff and Eduardo S. Schwartz. Valuing american options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.

Robert C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

Manuel Moreno and Javier F. Navas. On the robustness of least-squares monte carlo (lsm) for pricing american derivatives. *Review of Derivatives Research*, 6(2):107–128, 2003.

National Institute of Standards and Technology. Nist digital library of mathematical functions, 2023. https://dlmf.nist.gov/.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer, Berlin, 6 edition, 2003. doi: 10.1007/978-3-642-14394-6.

Goran Peskir and Albert N. Shiryaev. *Optimal Stopping and Free-Boundary Problems*. Lectures in Mathematics. ETH Zürich. Birkhäuser Verlag, Basel, 2006. doi: 10.1007/978-3-7643-7391-0.

Steven E. Shreve. *Stochastic Calculus for Finance I: The Binomial Asset Pricing Model*. Springer, 2004a.

Steven E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer, 2004b.

Lars Stentoft. Convergence of the least squares monte carlo approach to american option valuation. *Management Science*, 50(9):1193–1203, 2004.

Street Stats. Us treasury rates. `https://streetstats.finance/rates/treasuries`, 2026. Accessed: May 2026.