Chapter 5. Pricing Options with the LSMC Method (v.A.3)

We have seen in the previous chapter that the arbitrage-free price of a European-style contingent claim can be expressed conditional expected value of its discounted payoff under the risk-neutral probability measure $P*$:

$$V_t = \mathbb{E}_t^*\left(e^{-\int_t^T r(s)\,ds}V_T\right) = \mathbb{E}_t^*\left(e^{(T-t)r}V_T\right)$$

Here V_T is the payoff of the claim at time T and we assume that the risk-free rate is a constant. The first expectation corresponds to the case of stochastic interest rate, and the second one to a constant interest rate (risk-free rate).

Define a stopping time τ on a probability space $(\Omega, \Lambda, \mathrm{P})$ as $\tau: \Omega \to \{t_0, t_1, \dots, t_n\}$, so that $\{\tau = t_k\} \in \Lambda$ for any $k = 0,1, \dots, n$.

To price American-type options, we will take the exercise of the option over all possible stopping times and use the discounted expected future payoff idea of pricing (under the risk-neutral measure). The exercise price of the option will be random – it is the optimal stopping time. The price of the option at time t, V_t, will be given by the following expression:

$$V_t = \sup_{\tau\in[t,T]} \mathbb{E}_t^*\left(e^{-(\tau-t)r}\, Payoff(\tau)|\mathcal{F}_t\right)$$

where $Payoff(\tau)$ is the payoff of the option at the optimal exercise time τ.

For American Put options, in particular, the pricing formula at time t will be:

$$V_t = \sup_{\tau\in[t,T]} \mathbb{E}_t^*\left(e^{-(\tau-t)r}(X - S_\tau)^+|\mathcal{F}_t\right)$$

Assume τ^* is the optimal stopping time that solves the above problem. For options, define EV, CV, and ECV as follows:

$$\boldsymbol{EV_t} = Exercice\ Value\ at\ time\ t$$

$$\boldsymbol{CV_t} = Continuation\ Value\ at\ time\ t$$

$$\mathbb{E}\boldsymbol{CV_t} = Expected\ Conditional\ Continuation\ Value\ at\ time\ t.$$

The optimal stopping/exercise time τ^*can be expressed as follows:

$$\tau^* = The\ first\ time\ that\ the\ Exercice\ Value$$
$$\geq The\ \boldsymbol{Expected}\ Conditional\ Continuation\ Value\ of\ the\ option$$

$$\tau^* = min\{t \geq 0: EV_t \geq \ \mathbb{E}CV_t\} = min\{t \geq 0: (X - S_t)^+ \geq V_t\}$$

There is no closed-form expression for the optimal exercise time τ^*, or for the optimal exercise boundary, (stock prices (as function of time) for which it is optimal to exercise American Put options).

The problems of finding optimal exercise time or exercise boundaries are solved numerically. One of such methods is the Least Square Monte Carlo method, which will be explained in detail below.

The Least-Square-Monte-Carlo Method (LSMC)

The main idea of pricing American-type options via simulation is as follows. Define

$$V_T = (X - S_T)^+ \text{ and } V_t = \max\,(EV_t,\ \mathbb{E}CV_t|\mathcal{F}_t) \text{ for any } t \leq T$$

The goal is to estimate V_0, which is the value of the option at time 0. The estimation will be done recursively, by backward estimation.

Divide the time-interval [0,T] by n equal parts: $0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T$, where $t_k = \frac{T}{n}k = \Delta k$, where $\Delta = \frac{T}{n}$. Simulate *m* paths of the stock prices at times $t_0, t_1, \dots, t_{n-1}, t_n$.

We start at the terminal time $t_n = T$. Compute the exercise value (EV) of the option $(X - S_{t_n}^i)^+$, for every path $i = 1, \dots m$. There is no continuation value (CV) at this time step as this is the last time step, so the option is at its expiration. Therefore, the option values will simply be their exercise values in the final time step $t = t_n$. Now we have the option values for every path, at time $t = t_n$.

Next, we move backwards in time, to time step t_{n-1}, and estimate the exercise value (EV) of the option at every node $(i, n-1)$ for $i = 1, \ldots m$: $(X - S^i_{t_{n-1}})^+$. We also compute[1] the Expected Continuation Value ($\mathbb{E}CV$) of the option at every node $(i, n-1)$ of time step t_{n-1}. Then, we compare the EV to $\mathbb{E}CV$ and take the larger of the two as the value of the option at node $(i, n-1)$ of time step t_{n-1}.

Continuing this process (moving backwards in time, one period at a time) until time $t = t_0$ will lead to computation of V_0 , the value of the option at time $t_0 = 0$.

Examples:

1. In Binomial Framework:

$$V_t = \max\left(EV_t,\ \mathbb{E}CV_t|\mathcal{F}_t\right) = \max\left((X - S_t)^+,\ e^{-r\Delta}(pV_u + (1-p)V_d)\right)$$

Notice that, the Expected Continuation Value ($\mathbb{E}CV$) of the option is given by $e^{-r\Delta}(pV_u + (1-p)V_d)$ in this model.

2. In Trinomial Framework:

$$V_t = \max\left(EV_t,\ \mathbb{E}CV_t|\mathcal{F}_t\right) = \max\left((X - S_t)^+,\ e^{-r\Delta}(p_uV_u + p_mV_m + p_dV_d)\right)$$

Notice that, the Expected Continuation Value ($\mathbb{E}CV$) of the option is given by $e^{-r\Delta}(p_uV_u + p_mV_m + p_dV_d)$ in this model.

[1] The details of this computation will be provided later.

3. In Continuous-Time Setting: LSMC Method

The estimation technique was described above:

$$V_t = \max\left(EV_t,\ \mathbb{E}CV_t|\mathcal{F}_t\right) \text{ for any } t \leq T$$

The challenge here is to estimate the expected continuation value, $\mathbb{E}CV_t$:

$\mathbb{E}CV_t$=$\mathbb{E}_t^*$(Sum of all discounted Cash Flows after time $t|\mathcal{F}_t$).

Define $\Delta = \frac{T}{n}$. Divide the time-interval by n equal parts:

$$0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T, \text{ where } t_k = \frac{T}{n}k = \Delta k.$$

Then, for every *k=n-1,…,0,*

$\mathbb{E}CV_{t_k}$=$\mathbb{E}^*\left(Sum\ of\ all\ discounted\ Cash\ Flows\ after\ time\ t_k \middle| \mathcal{F}_{t_k}\right)$

$$=\mathbb{E}^*\left(\sum_{j=k+1}^{n} e^{-(t_j-t_k)r}\ CashFlow(t_j, t_k, T)\ |\mathcal{F}_{t_k}\right)$$

where $CashFlow(t_j, t_k, T)$ is the payoff of the option at time $t_j > t_k$. Notice that, along each path of the stock price process **at most one of these cash flows can be non-zero**.

Thus, the problem is to estimate the $\mathbb{E}CV$ at any node for the stock price, and at any time. At any fixed time t_k, the $\mathbb{E}CV$ is a function of the stock price at time t_k. The functional form of $\mathbb{E}CV$ (as a function of the stock price) will be different from one time step to another.

The estimation method of $\mathbb{E}CV$ is based on the Least-Square approximation of functions in L^2 spaces.

Assume the $\mathbb{E}CV$ functions are smooth enough to belong to the space L^2. Then, for any orthonormal system of basis functions $\{L_l(x)\}_{l=1}^{\infty}$ of the space L^2, we have the following representation:

$$\mathbb{E}CV(x) = \sum_{l=1}^{\infty} a_l L_l(x)$$

This representation can be approximated by a truncated sum of the above infinite series:

$$\mathbb{E}CV(x) \approx \sum_{l=1}^{k} a_l L_l(x)$$

For illustration purposes, we ASSUME that are able to estimate the scalar coefficients $\{a_{1,} a_2, \ldots, a_k\}$.

Then, at any node (i, j), we can compute the expected continuation value of the option:

$$\mathbb{E}CV(S_j^i) = \sum_{l=1}^{k} a_l L_l(S_j^i)$$

Define the function of the stock price $Y_t(S)$ (at time t) as:

$$Y_t(S) = \mathbb{E}_t CV(S).$$

We need to estimate the functional form of the $Y_k(S)$ function at every time step t_k for $k = (n - 1), (n - 2), \ldots\ 2{,}1$.

Remark: One may wonder, why not use $Y(S_j^i)$ as expected continuation value in node (i, j) (which would be easy as we know the value of the option in the next time-step: $(i, j + 1)$)? The reason for not using $Y(S_j^i)$ as expected continuation value is that it is one observation of CV, but what we need is the Conditional Expected Continuation Value at node (i, j), and not just one realization of CV.

Below we provide more details of the technique.

Start at S_0 and use the standard simulation methods to simulate m paths of the stochastic process $\{S_t : 0 \le t \le T\}$ at points $0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T$, where $t_k = \frac{T}{n} k$.

Store all the paths in the computer memory in a matrix form as shown below:

Stock Prices **↓ Time →**	$\boldsymbol{t_0 = 0}$	$\boldsymbol{t_1}$	$\boldsymbol{\dots}$	$\boldsymbol{\dots}$	$\boldsymbol{t_{n-2}}$	$\boldsymbol{t_{n-1}}$	$\boldsymbol{t_n = T}$
Path 1	S_0^1	S_1^1			S_{n-2}^1	S_{n-1}^1	S_n^1
Path 2	S_0^2	S_1^2	$\cdots$	$\cdots$	S_{n-2}^2	S_{n-1}^2	S_n^2
…	$\cdots$	$\cdots$			$\cdots$	$\cdots$	$\cdots$
Path (m-1)	S_0^{m-1}	S_1^{m-1}			S_{n-2}^{m-1}	S_{n-1}^{m-1}	S_n^{m-1}
Path m	S_0^m	S_1^m	$\cdots$	$\cdots$	S_{n-2}^m	S_{n-1}^m	S_n^m

Note: the index j in S_j^i is for time, and index i in S_j^i is for the path of the stock price. Also, $S_0^i = S_0$ for every $i = 1,2, \dots, m$.

We also create an mxn matrix, called **Index**, with the element in (i, j) being denoted by $\boldsymbol{Ind_j^i}$. Initially, we set all $Ind_j^i = 0$ for $j = 1, \dots, n$ and $i = 1, \dots, m$. Having a 1 in any cell of the matrix **Index** means that the option should be exercised at that cell of the stock price/time space.

The details of estimation steps are as follows:

At time $\boldsymbol{t = t_n = T}$

We compute the Exercise Value (EV): $EV_{t_n}^i = EV_{t_n}(S_n^i) = (X - S_n^i)^+$

and

Expected Continuation Value (ECV): $\mathbb{E}CV_{t_n}^i = \mathbb{E}CV_{t_n}(S_n^i) = 0$ for any $i = 1, \dots, m$.

Because $EV_{t_n}^i \geq \mathbb{E}CV_{t_n}^i$ for any $i = 0,1, \dots, m$, then, in those nodes where the option in in-the-money, we will exercise the option. Thus, we have all nodes where we exercise the option, and therefore we can populate the column n of the matrix **Index** the following way:

$$Ind_n^i = \begin{cases} 1, & if \;\; EV_{t_n}^i > 0 \\ 0, & otherwise \end{cases}$$

for any $i = 1, \dots, m$.

Note: Having 1's for certain entries of matrix Index means that the option should be exercised in those nodes, and having a 0 means the option should be kept alive in such nodes.

Now we move one step backwards in time, to time t_{n-1}.

At time $\boldsymbol{t = t_{n-1}}$:

Exercise Value: $EV_{t_{n-1}}^i = EV_{t_{n-1}}(S_{n-1}^i) = (X - S_{n-1}^i)^+$ for any $i = 1, \dots, m$.

Expected Continuation Value: We do not have a formula for this, but let's **ASSUME** that we can estimate the functional form $Y_{n-1}(x) = \mathbb{E}CV_{t_{n-1}}^i = \mathbb{E}CV_{t_{n-1}}(x)$ at this time step (the estimation steps for $Y_{n-1}(x)$ will be provided later).

Then,

$$\mathbb{E}CV_{t_{n-1}}^i = \mathbb{E}CV_{t_{n-1}}(S_{n-1}^i) = Y_{n-1}(S_{n-1}^i) \text{ for any } i = 1, \dots, m.$$

We can compare the $\mathbb{E}CV$ and EV and we have all nodes (at time $\boldsymbol{t_{n-1}}$) where we exercise the option, and therefore we can populate the column $(n-1)$ of the matrix **Index** the following way:

$$Ind_{n-1}^i = \begin{cases} 1, & if \;\; EV_{t_{n-1}}^i \geq \mathbb{E}CV_{t_{n-1}}^i \\ 0, & otherwise \end{cases}$$

for any $i = 0,1, \dots, m$.

Note: In each row of the matrix Index, we can have at most one 1. If $Ind_{n-1}^i = 1$ for any i, then we have to reset $Ind_n^i = 0$ for the same i, even if Ind_n^i was 1 for that i in the previous time-step.

Now we move one step backwards in time, to time t_{n-2}.

At time $t = t_{n-2}$:

Exercise Value: $EV_{t_{n-2}}^i = EV_{t_{n-2}}(S_{n-2}^i) = (X - S_{n-2}^i)^+$ for any $i = 1, \dots, m$.

Expected Continuation Value: We do not have a formula for this, but let's **ASSUME** that we can estimate the functional form $Y_{n-2}(x) = \mathbb{E}CV_{t_{n-2}}^i = \mathbb{E}CV_{t_{n-2}}(x)$ at this time step (the estimation steps for $Y_{n-2}(x)$ will be provided later).

Then,

$$\mathbb{E}CV_{t_{n-2}}^i = \mathbb{E}CV_{t_{n-2}}(S_{n-2}^i) = Y_{n-2}(S_{n-2}^i)$$

Now can compare the $\mathbb{E}CV$ and EV , and populate the column $n-2$ of the matrix Index the following way: for any $i = 0,1, \dots, m$,

$$Ind_{n-2}^i = \begin{cases} 1, & if \;\; EV_{t_{n-2}}^i \geq \mathbb{E}CV_{t_{n-2}}^i \\ 0, & otherwise \end{cases}$$

Note: In each row of matrix Index, we can have at most one 1. If $Ind_{n-2}^i = 1$, then we have to reset $Ind_{n-1}^i = 0$ and $Ind_n^i = 0$ for the same i, even if Ind_{n-1}^i or Ind_n^i were 1 for that i in the previous time-step.

Continuing the above-described steps recursively, we get to time $t = t_1$. At this stage, we have the matrix Index populated with 0 or 1's (each row can have at most one 1, which is the exercise time of the option along that path).

The estimated value of the option is given by:

$$V_0 = \frac{1}{m}\sum_{j=1}^{n}\sum_{i=1}^{m} (Ind_j^i)\, e^{-rj\Delta}\,(X - S_j^i)^+$$

Now, the only remaining question is: how to estimate the functional form of the expected continuation value function at every time-step? That is, how to estimate $Y_{n-l}(.)$ for any $= 1, \dots, n-1$?

The idea of estimation of the functional form of the expected continuation value is based on the Non-linear Least Square method.

Start with time $\boldsymbol{t = t_{n-1}}$.

We would like to estimate the functional form of $Y_{n-1}(S) = \mathbb{E}CV_{t_{n-1}}(S) = \mathbb{E}^*\big(CV(S)\big|\mathcal{F}_{t_{n-1}}\big)$. This is a random variable, and for each starting value of stock at time t_{n-1} we have one realization: $e^{-r\Delta}\,(X - S_n^i)^+$.

For every one of the independent variable X, we have a realization of the dependent variable Y: $X_i = S_{n-1}^i$, $Y_i = e^{-r\Delta}\,(X - S_n^i)^+$ for $i = 0,1, \dots, m$.

Thus, we have m −realizations of (X_i, Y_i):

X	**Y**
S_{n-1}^1	$Ind_n^1\, e^{-r\Delta}\,(X - S_n^1)^+$
S_{n-1}^2	$Ind_n^2\, e^{-r\Delta}\,(X - S_n^2)^+$
	...
	...
S_{n-1}^m	$Ind_n^m e^{-r\Delta}\,(X - S_n^m)^+$

Performance tip: Choose only those observations for which the option is in-the-money since the exercise information is relevant only in those cases. This will make computations more efficient.

Now, having m –realizations of the random variable function, we use the Least Square approach to estimate the functional form:

$$Y_{n-1}(x) \approx \sum_{l=1}^{k} a_l^{n-1} L_l(x)$$

The goal is to estimate the coefficients a_l^{n-1}.

Assume that we have already estimated these coefficients. Then, the expected continuation value at the i-th node of the stock price S_{n-1}^i (and at time t_{n-1}) will be given by

$$Y_{n-1}(S_{n-1}^i) = \sum_{l=1}^{k} a_l^{n-1} L_l(S_{n-1}^i).$$

The task now is to estimate the parameters $(a_1^{n-1}, a_2^{n-1}, \ldots, a_k^{n-1})$. Note that, these k parameters <u>will be different for every time step</u> and should be estimated for every time-step.

The estimation procedure is very similar to the estimation of coefficients in linear regressions.

Define

$$A_{n-1} = \begin{pmatrix} \langle f_1, f_1 \rangle & \cdots & \langle f_k, f_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle f_1, f_k \rangle & \cdots & \langle f_k, f_k \rangle \end{pmatrix}, \quad b_{n-1} = \begin{pmatrix} \langle Y, f_1 \rangle \\ . \\ \langle Y, f_k \rangle \end{pmatrix}, \quad a_{n-1} = \begin{pmatrix} a_1^{n-1} \\ . \\ a_k^{n-1} \end{pmatrix}$$

where

$$\langle f_i, f_j \rangle = L_i(X_1)L_j(X_1) + \cdots + L_i(X_m)L_j(X_m)$$

$$\langle Y, f_j \rangle = Y_1 L_j(X_1) + \cdots + Y_m L_j(X_m)$$

$$X_i = S_{n-1}^i, Y_i = Y_{n-1}(S_{n-1}^i)$$

for any $j = 1, \ldots, k$ and $i = 1, \ldots, k$.

The problem of finding the set of parameters $a_{n-1} = (\, a_1^{n-1}, a_2^{n-1}, \dots, a_k^{n-1})'$ will boil down to solving a system of linear equations

$$A_{n-1} a_{n-1} = b_{n-1}$$

The solution of this system can be obtained by writing

$$a_{n-1} = {A_{n-1}}^{-1} b_{n-1}$$

Thus, we can solve for the parameters a_{n-1} at the time step $t = t_{n-1}$, then, estimate the functional form of the expected continuation value function $Y_{n-1}(X)$, then, for every node make a decision to exercise or to keep the option alive, then, update the entries in the $(n-1)st$ column of the Index matrix. This describes the method for the time step $\boldsymbol{t = t_{n-1}}$.

At time $\boldsymbol{t = t_{n-2}}$.

We would like to estimate the functional form $Y_{n-2}(S) = \mathbb{E}^*\left(CV(S) \middle| \mathcal{F}_{t_{n-2}}\right)$. This is a random variable, for which we have m-realizations. For every realization $X_i = S_{n-2}^i$ of the independent variable X, we have a realization of the dependent variable Y: $Y_i = Ind_{n-1}^i \, e^{-r\Delta} \left(X - S_{n-1}^i\right)^+ + Ind_n^i \, e^{-r2\Delta} \left(X - S_n^i\right)^+$ for $i = 1, \dots, m$.

Note that, at most one of the two terms in Y_i can be non-zero. Thus, we have m −realizations of (X_i, Y_i):

$\boldsymbol{X}$	$\boldsymbol{Y}$
S_{n-2}^1	$Ind_{n-1}^1 \, e^{-r\Delta} (X - S_{n-1}^1)^+ \; + \; Ind_n^1 \, e^{-r2\Delta} (X - S_n^1)^+$
	…
	…
	…
S_{n-2}^m	$Ind_{n-1}^m \, e^{-r\Delta} (X - S_{n-1}^m)^+ \; + \; Ind_n^m \, e^{-r2\Delta} (X - S_n^m)^+$

Note: Choose only those observations for which the option is in-the-money since the exercise information is relevant only in those cases. This will make computations more efficient.

Now, having m –realizations of the function, we use the Least Square approach to estimate the functional form of ECV at this time step:

$$Y_{n-2}(x) \approx \sum_{l=1}^{k} a_l^{n-2} L_l(x)$$

The goal is to estimate the coefficients a_l^{n-2}.

Assume we have already estimated these coefficients. Then, the expected continuation value at any node (of time t_{n-2}) for the stock price S_{n-2}^i will be given by

$$Y_{n-2}(S_{n-2}^i) = \sum_{l=1}^{k} a_l^{n-2} L_l(S_{n-2}^i).$$

The task now is to estimate the parameters $a_{n-2} = (a_1^{n-2}, a_2^{n-2}, \dots, a_k^{n-2})$.

Note that, these k parameters <u>will be different for every time step</u> and should be estimated for every time-step.

Define

$$A_{n-2} = \begin{pmatrix} \langle f_1, f_1 \rangle & \cdots & \langle f_k, f_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle f_1, f_k \rangle & \cdots & \langle f_k, f_k \rangle \end{pmatrix}, \quad b_{n-2} = \begin{pmatrix} \langle Y, f_1 \rangle \\ . \\ \langle Y, f_k \rangle \end{pmatrix}, \quad a_{n-2} = \begin{pmatrix} a_1^{n-2} \\ . \\ a_k^{n-2} \end{pmatrix}$$

where

$$\langle f_i, f_j \rangle = L_i(X_1) L_j(X_1) + \cdots + L_i(X_m) L_j(X_m)$$

$$\langle Y, f_j \rangle = Y_1 L_j(X_1) + \cdots + Y_m L_j(X_m)$$

$$X_i = S_{n-2}^i, \quad Y_i = Y_{n-2}(S_{n-2}^i)$$

for any $j = 1, \dots, k$ and $i = 1, \dots, k$.

The problem of finding the set of parameters $a_{n-2} = (\, a_1^{n-2}, a_2^{n-2}, \dots, a_k^{n-2})'$ will boil down to solving a system of linear equations $A_{n-2} a_{n-2} = b_{n-2}$.

The solution to this system can be obtained by writing $a_{n-2} = {A_{n-2}}^{-1} b_{n-2}$.

We will repeat this process of estimating the vector *a* of *k* coefficients and thus the functional form of the expected continuation value at times $t_{n-3}, t_{n-4}, \dots t_2, t_1$. Thus, we can populate the entire matrix Index when we get to time t_1.

A Numerical Illustration

Consider an American Put Option on a stock that is priced at \$100, the Strike Price of the option is \$97.50, the risk-free rate (continuously compounded for all maturities) is 5%, and the option's expiration is in 3 years.

We divide the time interval into 3 equal parts, and simulate 10 paths of the stock prices. We use Hermite Polynomials as Basis Functions, using k=3 of its functions.

Table 1. Simulated 10 paths of the stock prices. Set all entries of the matrix **Index** equal to 0.

Stock Prices				
Path	**t=0**	**t=1**	**t=2**	**t=3**
1	100	92.8	108.8	121.1
2	100	100.1	94.2	92.1
3	100	98.87	93.11	97.8
4	100	96.34	93.11	90.36
5	100	102.1	100.1	96.43
6	100	98.3	110.2	99.2
7	100	102.9	120.1	128.4
8	100	110.2	98.2	94.5
9	100	89.87	93.8	90
10	100	86.12	90.21	98.34

Index				
Path	**t=0**	**t=1**	**t=2**	**t=3**
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0

0: Do nothing
1: Exercise

At time 3 (option's expiration), compute the exercise values. In Column t=3 of the matrix Index, set those entries equal to 1, in which the option will be exercised (because it is in the money). See Table 2 for the results of this step.

Table 2.

Time t= 3				Option Payoff At Maturity (time 3)
Path	**t=0**	**t=1**	**t=2**	**t=3**
1				0
2				5.4
3				0
4				7.14
5				1.07
6				0
7				0
8				3
9				7.5
10				0

Index				
Path	**t=0**	**t=1**	**t=2**	**t=3**
1	0	0	0	0
2	0	0	0	1
3	0	0	0	0
4	0	0	0	1
5	0	0	0	1
6	0	0	0	0
7	0	0	0	0
8	0	0	0	1
9	0	0	0	1
10	0	0	0	0

Use the Nonlinear Least Square with 3 Hermite Polynomials to estimate the ECV functional form. Of the 10 paths, take only those paths in which the option is in-the-money (at time t=2).

See Table 3 for the results of this step.

Table 3.

Time t= 2 ITM paths ONLY

Path	X		Y
1			
2	94.2		5.136639
3	93.11		0
4	93.11		6.791778
5			
6			
7			
8			
9	93.8		7.134221
10	90.21		0

HERMITE	k=3		A		b	(a1, a2, a3)'
L1(x)	1	3	928.86	172585	19.0626	-0.0001
L2(x)	2x	928.86	172595	3.2E+07	3570.89	-0.7263
L3(x)	4x^2-2	172585	3.2E+07	6E+09	668889	0.0040

ECV (S)=a1+a2*2(S)+a3*(4S^2-2)

After estimating the ECV and comparing them with EV at all nodes of t=2, make an option exercise decision. In Column t=2 of the matrix Index, set those entries equal to 1, in which the option will be exercised (because EV> ECV).

Notice, that in path 4, we set the value of Index at time t=2 to 1, AND reset the value at time t=3 to 0 (from 1). See Table 4 for the results of this step.

Table 4.

OPTIMAL EXERCISE DECISION

Time t= 2		
Path	**EV**	**CV**
1		
2	3.3	5.82454
3	4.39	4.12536
4	4.39	4.12536
5		
6		
7		
8		
9	3.7	5.19655
10	7.29	-0.2094

Index				
Path	**t=0**	**t=1**	**t=2**	**t=3**
1			0	0
2			0	1
3			1	0
4			1	0
5			0	1
6			0	0
7			0	0
8			0	1
9			0	1
10			1	0

Use the Nonlinear least square with 3 Hermite Polynomials to estimate the ECV functional form at time t=1. Of the 10 paths, take only those 4 in which the option is in-the-money (at time t=1). See Table 5 for the results of this step.

Table 5.

Time t= 1 ITM paths ONLY

Path	X	Y
1	92.8	0
2		
3		
4	96.34	4.1759
5		
6		
7		
8		
9	89.87	6.7863
10	86.12	6.9345

HERMITE	k=3
L1(x)	1
L2(x)	2x
L3(x)	4x^2-2

A		
3	922.94	133538
922.94	133546	2E+07
133538	2.4E+07	4E+09

b
17.9
3219
6E+05

(a1, a2, a3)'
0.2910
0.0047
0.0001

ECV (S)=a1+a2*2(S)+a3*(4S^2-2)

X: stock Price at time t=1

Y: pv (at time t=1) of option payoff along the path

After estimating the ECV and comparing them with EV at all nodes of t=1, make an option-exercise decision. In Column t=1 of the matrix Index, set those entries equal to 1, in which the option will be exercised (because EV> ECV).

Notice that, in paths 9 and 10, we set the value of Index at time t=1 to 1, AND we reset the value from 1 to 0 at time t=3 and in path 9 and, from 1 to 0 at time t=2 and in path 10. See Table 6 for the results of this step.

Table 6.

OPTIMAL EXERCISE DECISION		
Time t= 1		
Path	**EV**	**CV**
1	4.7	4.434
2		
3		
4	1.16	4.721
5		
6		
7		
8		
9	7.63	4.203
10	11.38	3.917

Index				
Path	**t=0**	**t=1**	**t=2**	**t=3**
1		1	0	0
2		0	0	1
3		0	1	0
4		0	1	0
5		0	0	1
6		0	0	0
7		0	0	0
8		0	0	1
9		1	0	0
10		1	0	0

The Matrix Index has 1's in certain rows (no more than one in each row, because the option can be exercised only once along any path). Those are the exercise times. At those nodes we can compute the option exercise values. See Table 7 for the results of this step.

Table 7.

EXERCISE VALUES

Path	t=0	t=1	t=2	t=3
1		4.7		
2				5.4
3			4.39	
4			4.39	
5				1.07
6				
7				
8				3
9		7.63		
10		11.38		

PRICE(0) = $3.86

Path	t=0	t=1	t=2	t=3
1	0	1	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	0
5	0	0	0	1
6	0	0	0	0
7	0	0	0	0
8	0	0	0	1
9	0	1	0	0
10	0	1	0	0

Then, find the sum of the present value of each of the 8 exercise values and divide it by 10 to estimate the value of the option: c=$3.86.

Comments:

1. *How to solve the linear system of equations $Ax = b$?*

The are many well-studied methods for solving linear systems of equations such as $Ax = b$. Gaussian elimination method is one of them. The basic idea of the method is the following:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + \ a_{1k}x_k = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + \ a_{2k}x_k = b_2 \\ \qquad\qquad\qquad \dots \\ \qquad\qquad\qquad \dots \\ a_{k1}x_1 + a_{k2}x_2 + \cdots + \ a_{kk}x_k = b_k \end{cases}$$

Use linear combinations of rows to eliminate x_1 in every row from 2 to k (just keep it in the first row):

$$New\ Row_i = Old\ Row_i - \frac{a_{i1}}{a_{11}} Row_1$$

Now we have x_1 only in the first equation (first row).

Do the same to eliminate the terms that contain x_2, from row 3 to k. After successful elimination of x_2, from row 3 to k, we will have x_2 only in row 2.

Repeat the procedure until we have a set of k equations, which can be written in a matrix form as

$$\tilde{A}x = b,$$

where $\tilde{A}$ is a diagonal matrix. That is, all elements of the matrix below the main diagonal are 0. Sometimes we may need to permute certain rows to make sure all the operations (dividing by numbers) are valid.

Now, it is easy to solve for x's: start from the last row first and solve it for x_k. Then, move to row (k-1), use the found value of x_k and solve for x_{k-1}. Repeat this procedure recursively to solve for all x's.

LU-decomposition or Cholesky-decomposition (among many others, such as Gauss-Seidel, SOR) are other methods for solving the above system of linear equation.

2. *Below we will provide some choices of basis functions for the least square estimation.*

Some choices for basis functions: two orthogonal function families and monomials:

	Hermite	**Laguerre**	**Monomials**
I-term	1	$e^{-x/2}$	1
II-term	$2x$	$e^{-\frac{x}{2}}(1-x)$	x
III-term	$4x^2-2$	$e^{-\frac{x}{2}}(1-2x+\frac{x^2}{2})$	x^2
IV-term	$8x^3-12x$	$e^{-\frac{x}{2}}(1-3x+\frac{3x^2}{2}-\frac{x^3}{6})$	x^3
V-term	$16x^4-56x^2+16$	$e^{-\frac{x}{2}}(1-4x+3x^2-\frac{2x^3}{3}+\frac{x^4}{24})$	x^4
n-th term	$L_n = 2xL_{n-1}-2(n-1)L_{n-2}$	$L_n(x)=e^{-\frac{x}{2}}\frac{e^x}{n!}\frac{d^n}{dx^n}(x^n e^{-x})$	x^n

Comment: Stentoft (2004) suggests that using ordinary monomials in the Least Square approximation (that is functions of type $L_j(x)=x^{j-1}$) is computationally preferable than the choice of (some) orthogonal basis functions, such as Laguerre polynomials.

3. *What choice of* ***k*** *is reasonable?*

Stentoft (2004) studies the trade-off between the precision of convergence (higher k) and the computational time. The study suggests that the best specification uses $k = 2$ or 3 with simple polynomial functions).

4. *How to price options on stocks with stochastic volatility?*

   That would be a 2-factor model and the method would apply in the pricing of an option in such a framework. We would use two sets of basis functions (for the two factors) and their cross-terms in the least square estimation. Everything else will carry over from the method described earlier.

5. For stability of faster convergence of the method, use only in-the-money paths in the least squares estimation step as the goal is to estimate the tptimal option exercise time.
6. Scaling all prices in the simulation by the exercise price has been shown to improve the stability of the algorithm.

Below we derive the solutions for the Nonlinear Least Square method.

Non-Linear Least Square Problem

Assume we have m −realizations of pairs (X_i, Y_i), $i = 0,1, \dots, m$ and we would like to find the best fit (to data) by linear combinations of nonlinear functions $L_j(x)$, for $j = 0,1, \dots, k$.

$$Y(X, \boldsymbol{a}) = \sum_{j=1}^{k} a_j L_j(x)$$

The goal is to estimate the vector of optimal coefficients $\boldsymbol{a} = (a_1, a_2, \dots, a_k)'$ that solves the following optimization problem:

$$\min_{\boldsymbol{a}} \{ L(\boldsymbol{a}) = \sum_{i=1}^{m} (Y_i - Y(X, \boldsymbol{a}))^2 \}$$

which can be rewritten as

$$\min_{\boldsymbol{a}} L(\boldsymbol{a}), \quad or \quad \min_{\boldsymbol{a}} \sum_{i=1}^{m} \left(Y_i - \sum_{j=1}^{k} a_j L_j(X_i) \right)^2$$

We apply the usual solution techniques to solve for $\boldsymbol{a}$:

1. **Find the First Order Conditions (F.O.C.) and solve them for the parameters.**
2. **Verify that the Second Order Conditions (S.O.C.) are satisfied.**
3. **The solution to the F.O.C. is the solution to the problem.**

The F.O.C are: $\frac{\partial L(\boldsymbol{a})}{\partial a_j} = 0$ for $j = 0,1, \dots, k$.

We can write them in an expanded form as a system of k equations as follows:

$$\begin{cases} \sum_{j=1}^{k} a_j \sum_{i=1}^{m} L_j(X_i) L_1(X_i) = \sum_{i=1}^{m} Y_i \, L_1(X_i) \\ \sum_{j=1}^{k} a_j \sum_{i=1}^{m} L_j(X_i) L_2(X_i) = \sum_{i=1}^{m} Y_i \, L_2(X_i) \\ \dots \\ \dots \\ \sum_{j=1}^{k} a_j \sum_{i=1}^{m} L_j(X_i) L_k(X_i) = \sum_{i=1}^{m} Y_i \, L_k(X_i) \end{cases}$$

This system can be modified into a matrix form as follows:

$$\begin{pmatrix} \sum_{i=1}^{m} L_1(X_i)L_1(X_i) & \sum_{i=1}^{m} L_1(X_i)L_2(X_i) & \dots\dots & \sum_{i=1}^{m} L_1(X_i)L_k(X_i) \\ & \dots & & \\ & \dots & & \\ \sum_{i=1}^{m} L_k(X_i)L_1(X_i) & \sum_{i=1}^{m} L_k(X_i)L_2(X_i) & \dots\dots & \sum_{i=1}^{m} L_k(X_i)L_k(X_i) \end{pmatrix} \begin{pmatrix} a_1 \\ \dots \\ \dots \\ a_k \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} Y_i \, L_1(X_i) \\ \dots \\ \dots \\ \sum_{i=1}^{m} Y_i \, L_k(X_i) \end{pmatrix}$$

Make some notations:

$f_l = (L_l(X_1), L_l(X_2), \dots, L_l(X_m))$, Y$= (Y_1, Y_2, \dots, Y_m)$

and the scalar products

$$\langle f_l, f_v \rangle = L_l(X_1)L_v(X_1) + \cdots + L_l(X_m)L_v(X_m), \langle Y, f_l \rangle = Y_1 L_l(X_1) + \cdots + Y_m L_l(X_m)$$

Also, denote by $A = \begin{pmatrix} \langle f_1, f_1 \rangle & \cdots & \langle f_k, f_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle f_1, f_k \rangle & \cdots & \langle f_k, f_k \rangle \end{pmatrix}$, and by $b = \begin{pmatrix} \langle Y, f_1 \rangle \\ . \\ \langle Y, f_k \rangle \end{pmatrix}$, $a = \begin{pmatrix} a_1 \\ \cdot \\ a_k \end{pmatrix}$

We have

$$A\boldsymbol{a} = b$$

The solution can be written as

$$\widehat{\boldsymbol{a}} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

References.

Longstaff, F.A. and E.S. Schwartz , 2001,"Valuing American options by simulation: a simple least-squares approach". *Review of Financial Studies.* Volume 14, Number 1, pages 113-147.

Stentoft, Lars , 2004. "Assessing the Least Squares Monte-Carlo Approach to American Option Valuation," *Review of Derivatives Research*.

Multidimensional case.

We will demonstrate the application of the LSMC method in a multidimensional case by the following example of pricing spread options.

Example: Pricing an American Spread Option between ***a* shares of AMZN stock, *b* shares of JD stock, and strike price of *K*.**

Assume that the risk-neutral dynamics of JD and AMZN stock prices follow (correlated) Geometric Brownian Motion processes, given by the following stochastic differential equations, respectively:

$dS_{1,t} = rS_{1,t}dt + \sigma_1 S_{1,t}dW_t^1$ and $dS_{2,t} = rS_{2,t}dt + \sigma_2 S_{2,t}dW_t^2$, where $dW_t^1 dW_t^2 = \rho dt$,

The goal is to price of the following **American Spread Option between** a shares of AMZN stock, b shares of JD stock, and strike price of K. The payoff of the option, if it is exercised at time t, is

$$max\left(aS_{2,t} - bS_{1,t} - K, 0\right)$$

Default parameters: $a = 1, b = 40, K = \$100, r = 3\%,\ \rho = 0.8,\ \sigma_1 = 38\%,\ \sigma_2 = 27\%,\ S_{1,0} = \$75,\ S_{2,0} = \$3{,}225,\ T = 0.5$.

The pricing formula for this American-style spread option is given by:

$$\boldsymbol{V_0 = \sup_{\tau\in[0,T]} \mathbb{E}_t^*\left(e^{-\tau r}(aS_{2,\tau} - bS_{1,\tau} - K)^+\right)}$$

The estimation technique is as follows.

$$V_t = \max\,(EV_t,\ \mathbb{E}CV_t|\mathcal{F}_t) \text{ for any } t \le T$$

The challenge here is to estimate the expected continuation value, $\mathbb{E}CV_t$:

$\mathbb{E}CV_t$=$\mathbb{E}_t^*$(Sum of all discounted Cash Flows after time $t|\mathcal{F}_t$).

Define Δ= $\frac{T}{n}$. Divide the time-interval by n equal parts:

$$0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T, \text{ where } t_j = \frac{T}{n}j = \Delta j.$$

Then, for every $j = n-1, n-2, \dots, 2,1$:

$\mathbb{E}CV_{t_k}$=$\mathbb{E}^*\left(Sum\ of\ all\ discounted\ Cash\ Flows\ after\ time\ t_k \middle| \mathcal{F}_{t_k}\right)$

$$=\mathbb{E}^*\left(\sum_{j=k+1}^{n} e^{-(t_j-t_k)r}\ CashFlow(t_j, t_k, T)\ |\mathcal{F}_{t_k}\right)$$

where $CashFlow(t_j, t_k, T)$ is the payoff of the option at time $t_j > t_k$. Notice that, along each path of the underlying processes **at most one of these cash flows can be non-zero**.

Thus, the problem is to estimate the $\mathbb{E}CV$ at any node for the stock price, and at any time. At any fixed time t_k, the $\mathbb{E}CV$ is a function of the stock price at time t_k.

The functional form of $\mathbb{E}CV$ (as a function of underlyings) will be different from one time-step to another.

The estimation method of $\mathbb{E}CV$ is based on the Least-Square approximation of functions in L^2 spaces.

Assume the $\mathbb{E}CV$ functions are smooth enough to belong to the space L^2.

Then, for any orthonormal system of basis functions of 2 variables $\{L_l(x_1, x_2)\}_{l=1}^{\infty}$ of the space L^2, we have the following representation:

$$\mathbb{E}CV(x_1, x_2) = \sum_{l=1}^{\infty} a_l L_l(x_1, x_2)$$

This representation can be approximated by a truncated sum of the above infinite series:

$$\mathbb{E}CV(x_1, x_2) \approx \sum_{l=1}^{k} a_l L_l(x_1, x_2)$$

For illustration purposes, we ASSUME that are able to estimate the scalar coefficients $\{a_{1,} a_2, \dots, a_k\}$.

Then, at any node (i, j), we can compute the expected continuation value of the option:

$$\mathbb{E}CV\left(S_{1,j}^{i}, S_{2,j}^{i}\right) = \sum_{l=1}^{k} a_l L_l\left(S_{1,j}^{i}, S_{2,j}^{i}\right)$$

Define the function of the stock price $Y_t\left(S_{1,t}, S_{2,t}\right)$ (at time t) as:

$$Y_t\left(S_{1,t}, S_{2,t}\right) = \mathbb{E}_t CV\left(S_{1,t}, S_{2,t}\right).$$

We need to estimate the functional form of the $Y_k(S_1, S_2)$ function at every time step t_k for $k = (n - 1), (n - 2), \ldots\ 2,1$.

Remark: One may wonder, why not use $Y\left(S_{1,j}^{i}, S_{2,j}^{i}\right)$ as expected continuation value in node (i, j) (which would be easy as we know the value of the option in the next time-step: $(i, j + 1)$)?

The reason for not using $Y\left(S_{1,j}^{i}, S_{2,j}^{i}\right)$ as the expected continuation value is that it is only <u>one observation of Continuation Value</u> <u>(CV)</u>, but what we need is the Conditional <u>Expected Continuation Value</u> at node (i, j), and not just one realization of CV.

Below we provide more details of the technique.

Start at $S_{1,0}, S_{2,0}$ initial stock prices (underlyings) and use the standard simulation methods to simulate *m* paths of the stochastic process $\left\{S_{1,t}: 0 \le t \le T\right\}$ and $\left\{S_{2,t}: 0 \le t \le T\right\}$ at points $0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T$, where $t_j = \frac{T}{n} j$.

Store all the paths in the computer memory in a matrix form as shown below:

Stock Prices ↓ Time →		$t_0 = 0$	t_1	…	…	t_{n-2}	t_{n-1}	$t_n = T$
Path 1	*Stock 1*	$S_{1,0}^1$	$S_{1,1}^1$			$S_{1,n-2}^1$	$S_{1,n-1}^1$	$S_{1,n}^1$
	Stock 2	$S_{2,0}^1$	$S_{2,1}^1$			$S_{2,n-2}^1$	$S_{2,n-1}^1$	$S_{2,n}^1$
Path 2	*Stock 1*	$S_{1,0}^2$	$S_{1,1}^2$			$S_{1,n-2}^2$	$S_{1,n-1}^2$	$S_{1,n}^2$
	Stock 2	$S_{2,0}^2$	$S_{2,1}^2$			$S_{2,n-2}^2$	$S_{2,n-1}^2$	$S_{2,n}^2$
…		…	…			…	…	…
Path i	*Stock 1*	$S_{1,0}^i$	$S_{1,1}^i$			$S_{1,n-2}^i$	$S_{1,n-1}^i$	$S_{1,n}^i$
	Stock 2	$S_{2,0}^i$	$S_{2,1}^i$			$S_{2,n-2}^i$	$S_{2,n-1}^i$	$S_{2,n}^i$
…		…	…			…	…	…
Path m	*Stock 1*	$S_{1,0}^m$	$S_{1,1}^m$			$S_{1,n-2}^m$	$S_{1,n-1}^m$	$S_{1,n}^m$
	Stock 2	$S_{2,0}^m$	$S_{2,1}^m$			$S_{2,n-2}^m$	$S_{2,n-1}^m$	$S_{2,n}^m$

Note: The index j in $S_{1,j}^i$ or in $S_{2,j}^i$ is for time, and index i is for the path of the stock price.

Also, $S_{1,0}^i = S_{1,0}$ and $S_{2,0}^i = S_{2,0}$for every $i = 1,2, \dots, m$.

We also create an mxn matrix, called **Index**, with the element in (i, j) being denoted by $\boldsymbol{Ind_j^i}$.

Initially, we set all $Ind_j^i = 0$ for $j = 1, \dots, n$ and $i = 1, \dots, m$. Having a 1 in any cell of the matrix **Index** means that the option should be exercised at that cell of the stock price/time space.

The details of estimation steps are as follows:

At time $\boldsymbol{t = t_n = T}$

- We compute the Exercise Value (EV): $EV_{t_n}^i = EV_{t_n}(S_{1,n}^i, S_{2,n}^i) = (aS_{2,n}^i - bS_{1,n}^i - K)^+$

- Expected Continuation Value (ECV): $\mathbb{E}CV_{t_n}^i = \mathbb{E}CV_{t_n}(S_{1,n}^i, S_{2,n}^i) = 0$ for any $i = 1, \dots, m$.

Because $EV_{t_n}^i \geq \mathbb{E}CV_{t_n}^i$ for any $i = 0,1, \dots, m$, then, in those nodes where the option in in-the-money, we will exercise the option. Thus, we have all nodes where we exercise the option, and therefore we can populate the column n of the matrix **Index** the following way:

$$Ind_n^i = \begin{cases} 1, & if \;\; EV_{t_n}^i > 0 \\ 0, & otherwise \end{cases}$$

for any $i = 1, \dots, m$.

Note: Having 1's for certain entries of matrix Index means that the option should be exercised in those nodes, and having a 0 means the option should be kept alive in such nodes.

Now we move one step backwards in time, to time t_{n-1}.

At time $\boldsymbol{t = t_{n-1}}$:

- Exercise Value: $EV_{t_{n-1}}^i = EV_{t_{n-1}}(S_{1,n-1}^i, S_{2,n-1}^i) = (aS_{2,n-1}^i - bS_{1,n-1}^i - K)^+$ for any $i = 1, \dots, m$.
- Expected Continuation Value: We do not have a formula for this, but let's **ASSUME** that we can estimate the functional form $Y_{n-1}(x_1, x_2) = \mathbb{E}CV_{t_{n-1}}^i = \mathbb{E}CV_{t_{n-1}}(x_1, x_2)$ at this time step (the estimation steps for $Y_{n-1}(x_1, x_2)$ will be provided later).

Then,

$$\mathbb{E}CV_{t_{n-1}}^i = \mathbb{E}CV_{t_{n-1}}(S_{1,n-1}^i, S_{2,n-1}^i) = Y_{n-1}(S_{1,n-1}^i, S_{2,n-1}^i) \text{ for any } i = 1, \dots, m.$$

We can compare the $\mathbb{E}CV$ and EV and we have all nodes (at time $\boldsymbol{t_{n-1}}$) where we exercise the option, and therefore we can populate the column $(n-1)$ of the matrix **Index** the following way:

$$Ind_{n-1}^{i}=\begin{cases}1, & if\ \ EV_{t_{n-1}}^{i}\geq \mathbb{E}CV_{t_{n-1}}^{i}\\ 0, & otherwise\end{cases}$$

for any $i=0,1,\dots,m$.

Note: In each row of the matrix Index, we can <u>have at most one 1</u>. If $Ind_{n-1}^{i}=1$ for any i, then we have to reset $Ind_{n}^{i}=0$ for the same i, even if Ind_{n}^{i} was 1 for that i in the previous time-step.

Now we move one step backwards in time, to time t_{n-2}.

At time $\boldsymbol{t=t_{n-2}}$:

- Exercise Value: $EV_{t_{n-2}}^{i}=EV_{t_{n-2}}(S_{1,n-2}^{i},S_{2,n-2}^{i})=(aS_{2,n-2}^{i}-bS_{1,n-2}^{i}-K)^{+}$ for any $i=1,\dots,m$.
- Expected Continuation Value: We do not have a formula for this, but let's **ASSUME** that we can estimate the functional form $Y_{n-2}(x_1,x_2)=\mathbb{E}CV_{t_{n-2}}^{i}=\mathbb{E}CV_{t_{n-2}}(x_1,x_2)$ at this time step (the estimation steps for $Y_{n-2}(x_1,x_2)$ will be provided later).

Then,

$$\mathbb{E}CV_{t_{n-2}}^{i}=\mathbb{E}CV_{t_{n-2}}(S_{1,n-2}^{i},S_{2,n-2}^{i})=Y_{n-1}(S_{1,n-2}^{i},S_{2,n-2}^{i})\text{ for any } i=1,\dots,m.$$

Now can compare the $\mathbb{E}CV$ and EV , and populate the column $n-2$ of the matrix Index the following way: for any $i=0,1,\dots,m$,

$$Ind_{n-2}^{i}=\begin{cases}1, & if\ \ EV_{t_{n-2}}^{i}\geq \mathbb{E}CV_{t_{n-2}}^{i}\\ 0, & otherwise\end{cases}$$

Note: In each row of matrix Index, we can have at most one 1. If $Ind_{n-2}^{i}=1$, then we have to reset $Ind_{n-1}^{i}=0$ and $Ind_{n}^{i}=0$ for the same i, even if Ind_{n-1}^{i} or Ind_{n}^{i} were 1 for that i in the previous time-step.

Continuing the above-described steps recursively, we get to time $t = t_1$. At this stage, we have the matrix Index populated with 0 or 1's (each row can have at most one 1, which is the exercise time of the option along that path).

The estimated value of the option is given by:

$$V_0 = \frac{1}{m}\sum_{j=1}^{n}\sum_{i=1}^{m} (Ind_j^i)\, e^{-rj\Delta}\,(aS_{2,j}^i - bS_{1,j}^i - K)^+$$

Now, the only remaining question is:

How to estimate the functional form of the Expected Continuation Value function at every time-step?

That is, how to estimate $Y_{n-j}(.,.)$ for any $= 1, \dots, n-1$?

The idea of estimation of the functional form of the Expected Continuation Value is based on the Non-Linear Least Square method. Below we describe the application of the method in our case.

Start with time $\boldsymbol{t = t_{n-1}}$.

We would like to estimate the functional form of $Y_{n-1}(S_{1,} S_2) = \mathbb{E}CV_{t_{n-1}}(S_{1,} S_2) =$ $\mathbb{E}^*\left(CV(S_{1,} S_2)\middle|\mathcal{F}_{t_{n-1}}\right)$. This is a random variable, and for each starting value of stock at time t_{n-1} we have one realization: $e^{-r\Delta}\,(aS_{2,n}^i - bS_{1,n}^i - K)^+$.

For every pair of the independent variables (X_1, X_2), we have a realization of the dependent variable Y:

$X_1^i = S_{1,n-1}^i$, $X_2^i = S_{2,n-1}^i$, $Y^i = e^{-r\Delta}\,(aS_{2,n}^i - bS_{1,n}^i - K)^+$ for $i = 1, \dots, m$.

Thus, we have m −realizations of (X_1^i, X_2^i, Y^i):

$\boldsymbol{X_1}$	$\boldsymbol{X_2}$	**Y**
$S_{1,n-1}^1$	$S_{2,n-1}^1$	$Ind_n^1\ e^{-r\Delta}\,(aS_{2,n}^1 - bS_{1,n}^1 - K)^+$
$S_{1,n-1}^2$	$S_{2,n-1}^2$	$Ind_n^2\ e^{-r\Delta}\,(aS_{2,n}^2 - bS_{1,n}^2 - K)^+$
		...
		...
$S_{1,n-1}^m$	$S_{2,n-1}^m$	$Ind_n^m\ e^{-r\Delta}\,(aS_{2,n}^m - bS_{1,n}^m - K)^+$

Performance tip: Choose only those observations for which the option is in-the-money since the exercise information is relevant only in those cases. This will make computations more efficient.

Now, having m –realizations of the random variable function, we use the Least Square approach to estimate the functional form:

$$Y_{n-1}(x_1, x_2) \approx \sum_{l=1}^{k} a_l^{n-1} L_l(x_1, x_2)$$

The goal is to estimate the vector of *k* coefficients: $a^{n-1} = (a_1^{n-1}, \dots, a_l^{n-1}, \dots, a_k^{n-1})'$.

Assume that we have already estimated these *k* coefficients by the nonlinear least square method (to be demonstrated below).

Then, the expected continuation value along the i-th path (and at time t_{n-1}) will be given by

$$Y_{n-1}(S_{1,n-1,}^i\, S_{2,n-1}^i) = \sum_{l=1}^{k} a_l^{n-1} L_l(S_{1,n-1,}^i S_{2,n-1}^i).$$

The task now is to estimate the parameters $a^{n-1} = (a_1^{n-1}, a_2^{n-1}, \dots, a_k^{n-1})'$.

Note that, these k parameters <u>will be different for every time step</u> and they should be estimated for every time-step.

The estimation procedure is very similar to the estimation of coefficients in linear regressions.

Define

$$F^{n-1} = \begin{pmatrix} \langle f_1, f_1 \rangle & \cdots & \langle f_k, f_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle f_1, f_k \rangle & \cdots & \langle f_k, f_k \rangle \end{pmatrix}, \quad b^{n-1} = \begin{pmatrix} \langle Y, f_1 \rangle \\ . \\ \langle Y, f_k \rangle \end{pmatrix}, \quad a^{n-1} = \begin{pmatrix} a_1^{n-1} \\ . \\ a_k^{n-1} \end{pmatrix}$$

where

$$\langle f_i, f_j \rangle = L_i(S_{1,n-1,}^1 S_{2,n-1}^1)\, L_j(S_{1,n-1,}^1 S_{2,n-1}^1) + L_i(S_{1,n-1,}^2 S_{2,n-1}^2)\, L_j(S_{1,n-1,}^2 S_{2,n-1}^2) + \cdots +$$

$$L_i(S_{1,n-1,}^m S_{2,n-1}^m)\, L_j(S_{1,n-1,}^m S_{2,n-1}^m)$$

$$\langle Y, f_j \rangle = Y_1 L_j(S_{1,n-1,}^1 S_{2,n-1}^1) + Y_2 L_j(S_{1,n-1,}^2 S_{2,n-1}^2) + \cdots + Y_m L_j(S_{1,n-1,}^m S_{2,n-1}^m)$$

$$Y_i = Y_{n-1}(S_{1,n-1,}^i\, S_{2,n-1}^i)$$

for any $j = 1, \dots, k$ and $i = 1, \dots, k$.

The problem of finding the set of parameters $a^{n-1} = (\, a_1^{n-1}, a_2^{n-1}, \dots, a_k^{n-1})'$ will boil down to solving a system of linear equations

$$F^{n-1} a^{n-1} = b^{n-1}$$

The solution of this system is given by:

$$a^{n-1} = (F^{n-1})^{-1} b^{n-1}$$

Thus, we can solve for the parameters a^{n-1} at the time step $t = t_{n-1}$, then, estimate the functional form of the expected continuation value function $Y_{n-1}(.,.)$, then, for every node make a decision to exercise or to keep the option alive, then, update the entries in the $(n-1)st$ column of the Index matrix. This describes the method for the time step $\boldsymbol{t = t_{n-1}}$.

At time $\boldsymbol{t = t_{n-2}}$.

We would like to estimate the functional form $Y_{n-2}(S_{1,} S_2) = \mathbb{E}CV_{t_{n-2}}(S_{1,} S_2) =$

$\mathbb{E}^*\left(CV(S_{1,} S_2)\middle|\mathcal{F}_{t_{n-2}}\right)$.

This is a random variable, for which we have m-realizations. For every for $i = 1, \dots, m$ we have:

$$X_1^i = S_{1,n-2}^i, \;\; X_2^i = S_{2,n-2}^i,$$

$$Y^i = Ind_{n-1}^i e^{-r\Delta}\left(aS_{2,n-1}^i - bS_{1,n-1}^i - K\right)^+ + \; Ind_n^i e^{-r2\Delta}\left(aS_{2,n}^i - bS_{1,n}^i - K\right)$$

Remark: Note that, at most one of the two terms in Y_i above can be non-zero. In numerical implementation, it is important not to consume computational time in such unnecessary calculations.

Thus, we have m –realizations of (X_1^i, X_2^i, Y^i) for $i = 1, \dots, m$.

$\boldsymbol{X_1}$	$\boldsymbol{X_2}$	**Y**
$S_{1,n-2}^1$	$S_{2,n-2}^1$	$Ind_{n-1}^1\, e^{-r\Delta}\,(aS_{2,n-1}^1 - bS_{1,n-1}^1 - K)^+ + Ind_n^1\, e^{-r2\Delta}\,(aS_{2,n}^1 - bS_{1,n}^1 - K)^+$
$S_{1,n-2}^2$	$S_{2,n-2}^2$	$Ind_{n-1}^2\, e^{-r\Delta}\,(aS_{2,n-1}^2 - bS_{1,n-1}^2 - K)^+ + Ind_n^2\, e^{-r2\Delta}\,(aS_{2,n}^2 - bS_{1,n}^2 - K)^+$
		…
$S_{1,n-2}^m$	$S_{2,n-2}^m$	$Ind_{n-1}^m\, e^{-r\Delta}\left(aS_{2,n-1}^m - bS_{1,n-1}^m - K\right)^+ + Ind_n^m\, e^{-r2\Delta}\left(aS_{2,n}^m - bS_{1,n}^m - K\right)^+$

Remark: Choose only those observations for which the option is in-the-money since the exercise information is relevant only in those cases. This will make computations more efficient.

Now, having m –realizations of the function, we use the (nonlinear) Least Square approach to estimate the functional form:

$$Y_{n-2}(x_1, x_2) \;\approx \sum_{l=1}^{k} a_l^{n-2} L_l(x_1, x_2)$$

The goal is to estimate the vector of coefficients $a^{n-2} = (\, a_1^{n-2}, a_2^{n-2}, \ldots, a_k^{n-2})'$.

Assume we have already estimated these coefficients. Then, the expected continuation value at any node (of time t_{n-2}) for the stock price S_{n-2}^{i} will be given by

$$Y_{n-2}(S_{1,n-2,}^{i}\, S_{2,n-2}^{i}) = \sum_{l=1}^{k} a_l^{n-2} L_l(S_{1,n-2,}^{i} S_{2,n-2}^{i}).$$

The task now is to estimate the parameters $a^{n-2} = (\, a_1^{n-2}, a_2^{n-2}, \ldots, a_k^{n-2})'$

Note that, these k parameters <u>will be different for every time step</u> and they should be estimated for every time-step.

Define

$$F^{n-2} = \begin{pmatrix} \langle f_1, f_1 \rangle & \cdots & \langle f_k, f_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle f_1, f_k \rangle & \cdots & \langle f_k, f_k \rangle \end{pmatrix}, \quad b^{n-2} = \begin{pmatrix} \langle Y, f_1 \rangle \\ . \\ \langle Y, f_k \rangle \end{pmatrix}, \quad a^{n-2} = \begin{pmatrix} a_1^{n-2} \\ . \\ a_k^{n-2} \end{pmatrix}$$

where

$$\langle f_i, f_j \rangle = L_i(S_{1,n-2,}^{1} S_{2,n-2}^{1})\, L_j(S_{1,n-2,}^{1} S_{2,n-2}^{1}) + L_i(S_{1,n-2,}^{2} S_{2,n-2}^{2})\, L_j(S_{1,n-2,}^{2} S_{2,n-2}^{2}) + \cdots$$

$$+ L_i(S_{1,n-2,}^{m} S_{2,n-2}^{m})\, L_j(S_{1,n-2,}^{m} S_{2,n-2}^{m})$$

$$\langle Y, f_j \rangle = Y_1 L_j(S_{1,n-2,}^{1} S_{2,n-2}^{1}) + Y_2 L_j(S_{1,n-2,}^{2} S_{2,n-2}^{2}) + \cdots + Y_m L_j(S_{1,n-2,}^{m} S_{2,n-2}^{m})$$

$$Y_i = Y_{n-1}(S_{1,n-2,}^{i}\, S_{2,n-2}^{i})$$

for any $j = 1, \ldots, k$ and $i = 1, \ldots, k$.

The problem of finding the set of parameters a^{n-2}will boil down to solving the following system:

$$F^{n-2} a^{n-2} = b^{n-2}$$

The solution of this system is given by:

$$a^{n-2} = (F^{n-2})^{-1}b^{n-2}$$

We will repeat this process (of estimating the vector a of k coefficients and thus the functional form of the expected continuation value) for times $t_{n-3}, t_{n-4}, \dots t_2, t_1$, and populate the entire matrix Index.

Non-Linear Least Square Problem (two independent variables)

Assume we have m −realizations of pairs $(X_i^1, X_i^2\ Y_i)$, $i = 0,1,\dots,m$ and we would like to find the best fit (to data) by linear combinations of nonlinear functions $L_j(x_1, x_2)$, for $j = 0,1,\dots,k$.

$$Y(X_1, X_2, \boldsymbol{a}) = \sum_{j=1}^{k} a_j L_j(x_1, x_2)$$

The goal is to estimate the vector of optimal coefficients $\boldsymbol{a} = (\, a_1, a_2, \dots, a_k)'$ that solves the following optimization problem:

$$\min_{\boldsymbol{a}} \{\, L(\boldsymbol{a}) = \sum_{i=1}^{m} (Y_i - Y(X_i^1, X_i^2, \boldsymbol{a}))^2\}$$

which can be rewritten as

$$\min_{\boldsymbol{a}} L(\boldsymbol{a}), \quad \text{or} \quad \min_{\boldsymbol{a}} \sum_{i=1}^{m} \left(Y_i - \sum_{j=1}^{k} a_j L_j(X_i^1, X_i^2) \right)^2$$

We apply the usual solution techniques to solve for $\boldsymbol{a}$:

4. Find the First Order Conditions (F.O.C.) and solve them for the parameters.
5. Verify that the Second Order Conditions (S.O.C.) are satisfied.
6. Those solution to the F.O.C. that satisfy the S.O.C. are the solutions to the problem.

The F.O.C of the above problem are: $\frac{\partial L(\boldsymbol{a})}{\partial a_j} = 0$ for $j = 0,1,\dots,k$.

We can write them in an expanded form as a system of k equations as follows:

$$\begin{cases} \sum_{j=1}^{k} a_j \sum_{i=1}^{m} L_j(X_i^1, X_i^2) L_1(X_i^1, X_i^2) = \sum_{i=1}^{m} Y_i \, L_1(X_i^1, X_i^2) \\ \sum_{j=1}^{k} a_j \sum_{i=1}^{m} L_j(X_i^1, X_i^2) L_2(X_i^1, X_i^2) = \sum_{i=1}^{m} Y_i \, L_2(X_i^1, X_i^2) \\ \quad \ldots \\ \quad \ldots \\ \sum_{j=1}^{k} a_j \sum_{i=1}^{m} L_j(X_i^1, X_i^2) L_k(X_i^1, X_i^2) = \sum_{i=1}^{m} Y_i \, L_k(X_i^1, X_i^2) \end{cases}$$

Denote by $\boldsymbol{X_i} = (X_i^1, X_i^2)$. The above system can be modified into a matrix form as follows:

$$\begin{pmatrix} \sum_{i=1}^{m} L_1(\boldsymbol{X_i}) L_1(\boldsymbol{X_i}) & \sum_{i=1}^{m} L_1(\boldsymbol{X_i}) L_2(\boldsymbol{X_i}) & \ldots\ldots & \sum_{i=1}^{m} L_1(\boldsymbol{X_i}) L_k(\boldsymbol{X_i}) \\ & \ldots & & \\ & \ldots & & \\ \sum_{i=1}^{m} L_k(\boldsymbol{X_i}) L_1(\boldsymbol{X_i}) & \sum_{i=1}^{m} L_k(\boldsymbol{X_i}) L_2(\boldsymbol{X_i}) & \ldots\ldots & \sum_{i=1}^{m} L_k(\boldsymbol{X_i}) L_k(\boldsymbol{X_i}) \end{pmatrix} \begin{pmatrix} a_1 \\ \ldots \\ \ldots \\ a_k \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} Y_i \, L_1(\boldsymbol{X_i}) \\ \ldots \\ \ldots \\ \sum_{i=1}^{m} Y_i \, L_k(\boldsymbol{X_i}) \end{pmatrix}$$

Make some notations:

$f_l = (L_l(\boldsymbol{X_1}), L_l(\boldsymbol{X_2}), \ldots, L_l(\boldsymbol{X_m}))$, Y$= (Y_1, Y_2, \ldots, Y_m)$

and define the following scalar products

$$\langle f_l, f_v \rangle = L_l(\boldsymbol{X_1}) L_v(\boldsymbol{X_1}) + \cdots + L_l(\boldsymbol{X_m}) L_v(\boldsymbol{X_m}), \langle Y, f_l \rangle = Y_1 L_l(\boldsymbol{X_1}) + \cdots + Y_m L_l(\boldsymbol{X_m})$$

Also, denote by $A = \begin{pmatrix} \langle f_1, f_1 \rangle & \cdots & \langle f_k, f_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle f_1, f_k \rangle & \cdots & \langle f_k, f_k \rangle \end{pmatrix}$, $\quad b = \begin{pmatrix} \langle Y, f_1 \rangle \\ . \\ \langle Y, f_k \rangle \end{pmatrix}$, $\quad a = \begin{pmatrix} a_1 \\ \cdot \\ a_k \end{pmatrix}$

The above system of equations can be written in a matrix form as follows:

$$A\boldsymbol{a} = b$$

The solution of the system can be written as $\quad \boldsymbol{\hat{a} = A^{-1} b.}$

Note: It can be shown that the SOC are satisfied, therefore the solution to the SOC above is the solution to the optimization problem.

Exercises

1. Consider the following situation on the stock of company XYZ: The current stock price is \$40, and the volatility of the stock price is $\sigma = 20\%$ per annum. Assume the prevailing risk-free rate is $r = 6\%$ per annum. Use the following method to price the specified option:

(a) Use the LSMC method with 100,000 paths simulations (50,000 plus 50,000 antithetic) to price an American put option with strike price of $X = \$40$, maturity of 0.5-years, 1-year, 2-years, and current stock prices of \$36, \$40, \$44. Use Laguerre polynomials for $k = 2, 3, 4$.

(b) Use the LSMC method with 100,000 paths simulations (50,000 plus 50,000 antithetic) to price an American put option with strike price of $X = \$40$, maturity of 0.5-years, 1-year, 2-years, and current stock prices of \$36, \$40, \$44. Use Hermite polynomials for $k = 2, 3, 4$.

(c) Use the LSMC method with 100,000 paths simulations (50,000 plus 50,000 antithetic) to price an American put option with strike price of $X = \$40$, maturity of 0.5-years, 1-year, 2-years, and current stock prices of \$36, \$40, \$44. Use simple monomials for $k = 2, 3, 4$.

(d) Compare all your findings above and comment.

2. Consider the following 2-factor model for stock prices with stochastic volatility:

$$\begin{cases} dS_t = rS_t dt + \sqrt{V_t}\, S_t dW_t^1 \\ dV_t = a(b - V_t)dt + c\sqrt{V_t} dW_t^2 \end{cases}$$

where the Brownian Motion processes above are correlated: $dW_t^1 dW_t^2 = \rho dt$.

(a) Compute the price of an American Call option (via Least Square Monte Carlo simulation) that has a strike price of K and matures in T years.

Use Hermite polynomials for $k = 2, 3$.

Use the following parameters of the model: $= -0.6,\ r = 0.03,\ S_0 = \$48,\ V_0 = 0.05,$

$\sigma = 0.42,\ \alpha = 5.8,\ \ \beta = 0.0625.$

(b) Compute the price of an American Put option (via Least Square Monte Carlo simulation) that has a strike price of K and matures in T years.

Use simple monomials for $k = 2, 3$.

Use the following parameters of the model: $= -0.6,\ r = 0.03,\ S_0 = \$48,\ V_0 = 0.05,$

$\sigma = 0.42,\ \alpha = 5.8,\ \ \beta = 0.0625.$

3. Compute the prices of American Call options on the same stock with same specifications as in part (c) of the previous problem. Compare with the exact (Black-Scholes) formula and comment.

4. Forward start options are path dependent options that have strike prices to be determined at a future date. For example, a forward start put option payoff at maturity is

$$max(S_t - S_T, 0)$$

where the strike price of the put option is S_t. Here $0 \le t \le T$.

(a) Estimate the value of the forward-start European put option on a stock with these characteristics: $S_0 = \$65, X = \$60, \sigma = 20\%$ per annum, risk-free rate is $r = 6\%$ per annum, $t = 0.2$ and $T = 1$.

(b) Estimate the value of the forward-start American put option on a stock with these characteristics: $S_0 = \$65, X = \$60, \sigma = 20\%$ per annum, risk-free rate is $r = 6\%$ per annum, $t = 0.2$ and $T = 1$. The continuous exercise starts at time $t = 0.2$.