Valuing American Options by Simulation

- Least Square and Machine learning Approaches

12/2/19

Presented by: Jinyang Li

ACTSC 906 Final Project

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS

AMERICAN OPTION

OVERVIEW

- A general class of continuous-time American option pricing problems can be formulated by specifying a process $U(t)$, $0 < t < T$, representing the discounted payoff from exercise at time t, and a class of admissible stopping times τ_0 with values in $[0, T]$. The problem, then, is to find the optimal expected discounted payoff

$$sup_{\{\tau \in \tau_0\}} E[U(\tau)]$$

- We focus on American put options. In this case, consider a put with strike price K on a single underlying asset $S(t)$. The risk-neutral dynamics of S are modeled as geometric Brownian motion $GBM(r, \sigma^2)$, with r a constant risk-free interest rate. Suppose the option expires at T, its value at time 0 is then

$$sup_{\{\tau \in \tau_0\}} E[e^{-r\tau} \left(K\text{-}S(\tau)\right)^+]$$

AMERICAN OPTION

OVERVIEW

- Traditional ways:
  1. Black-Scholes: Due to the optimal stopping problem, this is technically impossible.
  2. Lattice Tree
  3. Finite difference
  4. Monte Carlo
- Machine Learning methods:
  1. Gaussian Process Regression
  2. Lease Square Policy Iteration
  3. Fitted Q-iteration

OPTIMAL STOPPING PROBLEM

OVERVIEW

- Optimal Stopping problem for Stochastic Process X_t

$$V(x) = sup_{\{\tau \in \tau_0\}} E[U(X_\tau) | X_0 = x]$$

- Where τ is a set of stopping times of X_t, V_t is called the Value function, and G is the Reward function.

- American option pricing is an optimal stopping problem.
  - X_t is the stochastic process for underlying security’s price
  - x is underlying security’s current price
  - τ is set of exercise times corresponding to various stopping policies
  - $V(\cdot)$ is American option price as function of underlying current price
  - $U(\cdot)$ is the discounted option payoff.

MARKOV DECISION PROCESSES

OVERVIEW

American Option Pricing is Optimal Stopping, and hence an MDP , So can be tackled with Dynamic Programming or RL algorithms.

- We formulate Stopping Time problems as Markov Decision Processes
- State is a suitable function of the history of Stochastic Process X_t
- Action is Boolean: Stop or Continue
- Reward always 0, except upon Stopping (when it is $= \ G\ (X_\tau\ )$)
- State-transitions governed by Underlying Price Stochastic Process

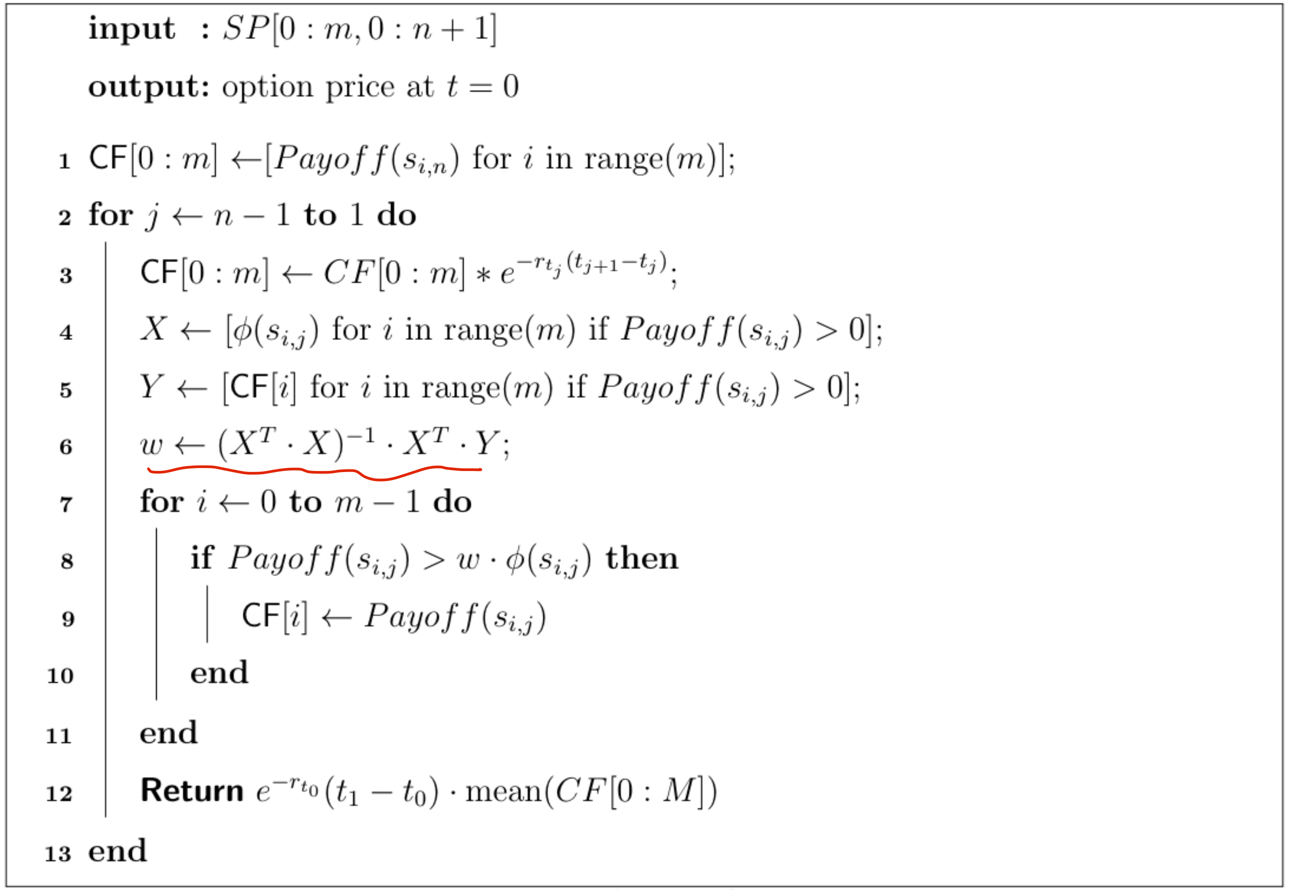
INGREDIENTS FOR APPROACHES

LSA

Before showing the pseudo-codes, below are the symbols used:

- m Monte-Carlo paths indexed $i = 0, 1, \ldots, m-1$
- $n+1$ time steps indexed $j = n, n-1, \ldots, 1, 0$
- Infinitesimal risk-free rate at time t_j denoted r_{t_j}
- Simulation paths of prices of underlying as input 2-dim array $SP[i, j]$
- At each time step, $CF[i]$ is PV of current+future cashflow for path i
- $s_{i,j}$ denotes state for $(i, j) :=$ (time t_j, price history $SP[i, :(j+1)]$)
- $Payoff(s_{i,j})$ denotes Option payoff at (i, j)
- $\phi_0(s_{i,j}), \ldots, \phi_{r-1}(s_{i,j})$ represents feature functions (of state $s_{i,j}$)
- $w_0, \ldots, w_{r-1}$ are the regression weights
- Regression function $f(s_{i,j}) = w \cdot \phi(s_{i,j}) = \sum_{I=0}^{r-1} w_I \cdot \phi_I(s_{i,j})$
- $f(\cdot)$ is estimate of continuation value for in-the-money states

LONGSTAFF-SCHWARTZ ALGORITHM



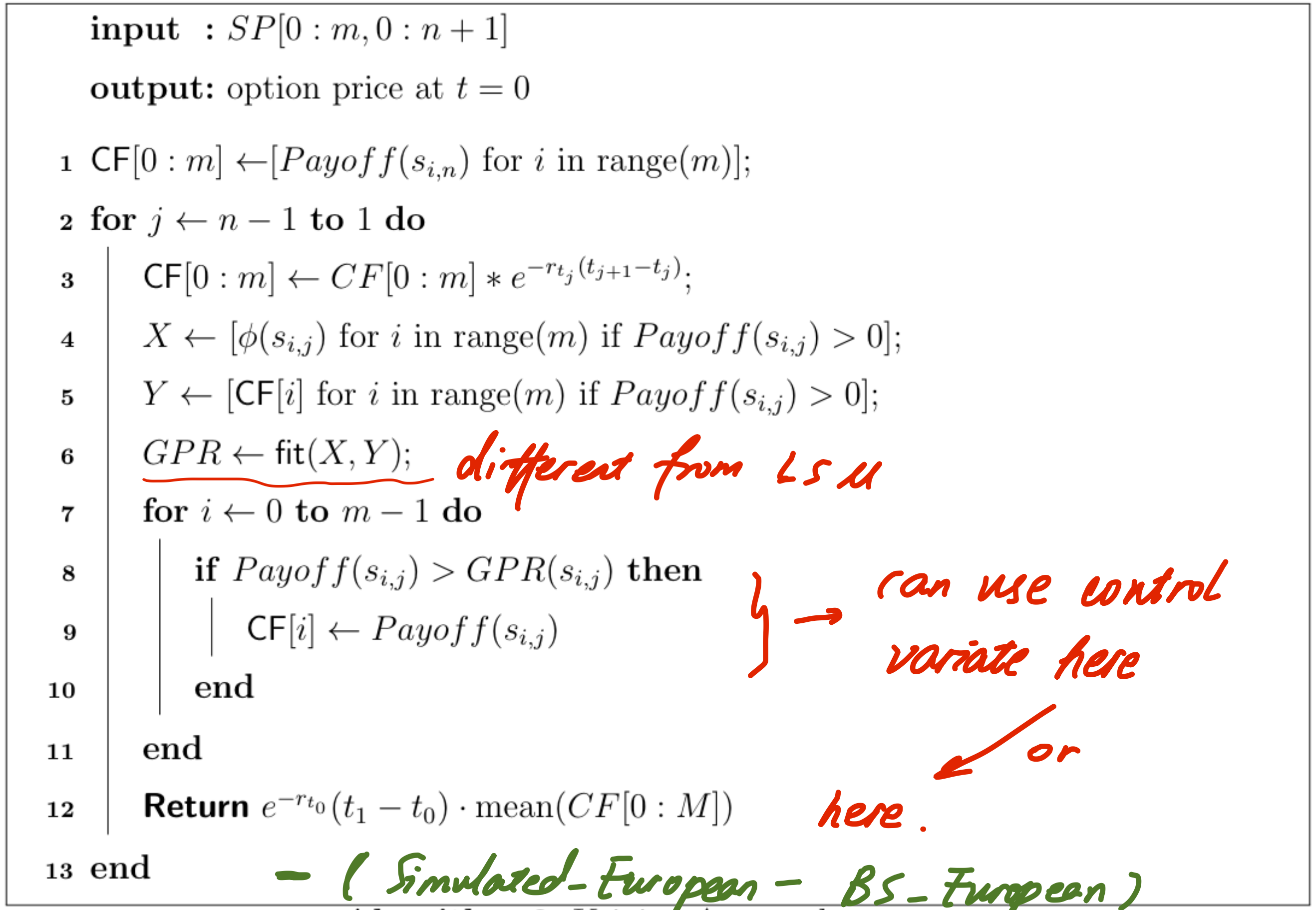
```
input  : $SP[0:m, 0:n+1]$
output: option price at $t = 0$

CF$[0:m]$ ←$[Payoff(s_{i,n})$ for $i$ in range$(m)]$;
for $j \leftarrow n-1$ to 1 do
    CF$[0:m] \leftarrow CF[0:m] * e^{-r_{t_j}(t_{j+1}-t_j)}$;
    $X \leftarrow [\phi(s_{i,j})$ for $i$ in range$(m)$ if $Payoff(s_{i,j}) > 0]$;
    $Y \leftarrow$ [CF$[i]$ for $i$ in range$(m)$ if $Payoff(s_{i,j}) > 0]$;
    $\underline{w \leftarrow (X^T \cdot X)^{-1} \cdot X^T \cdot Y}$;
    for $i \leftarrow 0$ to $m-1$ do
        if $Payoff(s_{i,j}) > w \cdot \phi(s_{i,j})$ then
            CF$[i] \leftarrow Payoff(s_{i,j})$
        end
    end
    Return $e^{-r_{t_0}}(t_1 - t_0) \cdot$ mean$(CF[0:M])$
end
```

Algorithm 1: Least Square Approach

GAUSSIAN PROCESS REGRESSION

ALGORITHMS



```
input  : $SP[0:m, 0:n+1]$
output: option price at $t = 0$

$\mathsf{CF}[0:m] \leftarrow [Payoff(s_{i,n})$ for $i$ in range$(m)]$;
for $j \leftarrow n-1$ to 1 do
    $\mathsf{CF}[0:m] \leftarrow CF[0:m] * e^{-r_{t_j}(t_{j+1}-t_j)}$;
    $X \leftarrow [\phi(s_{i,j})$ for $i$ in range$(m)$ if $Payoff(s_{i,j}) > 0]$;
    $Y \leftarrow [\mathsf{CF}[i]$ for $i$ in range$(m)$ if $Payoff(s_{i,j}) > 0]$;
    $GPR \leftarrow \mathsf{fit}(X, Y)$;
    for $i \leftarrow 0$ to $m-1$ do
        if $Payoff(s_{i,j}) > GPR(s_{i,j})$ then
            $\mathsf{CF}[i] \leftarrow Payoff(s_{i,j})$
        end
    end
    Return $e^{-r_{t_0}}(t_1 - t_0) \cdot \text{mean}(CF[0:M])$
end
```

Algorithm 2: Kriging Approach

different from LSM

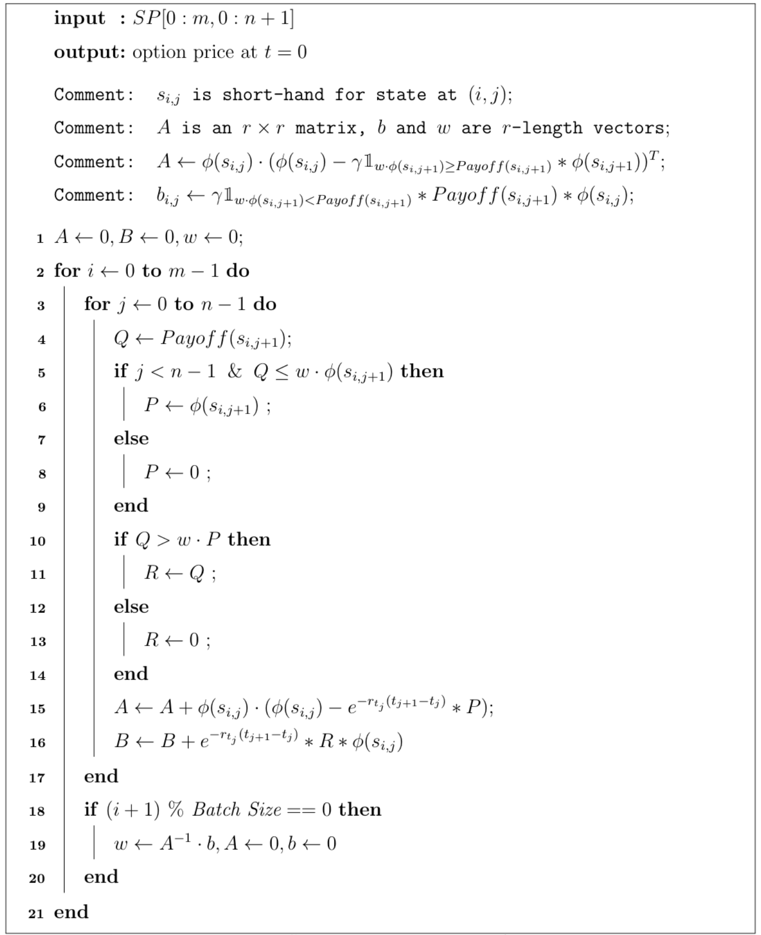
can use control variate here

or

here.

− (Simulated_European − BS_European)

LEAST SQUARES POLICY ITERATION APPROACH



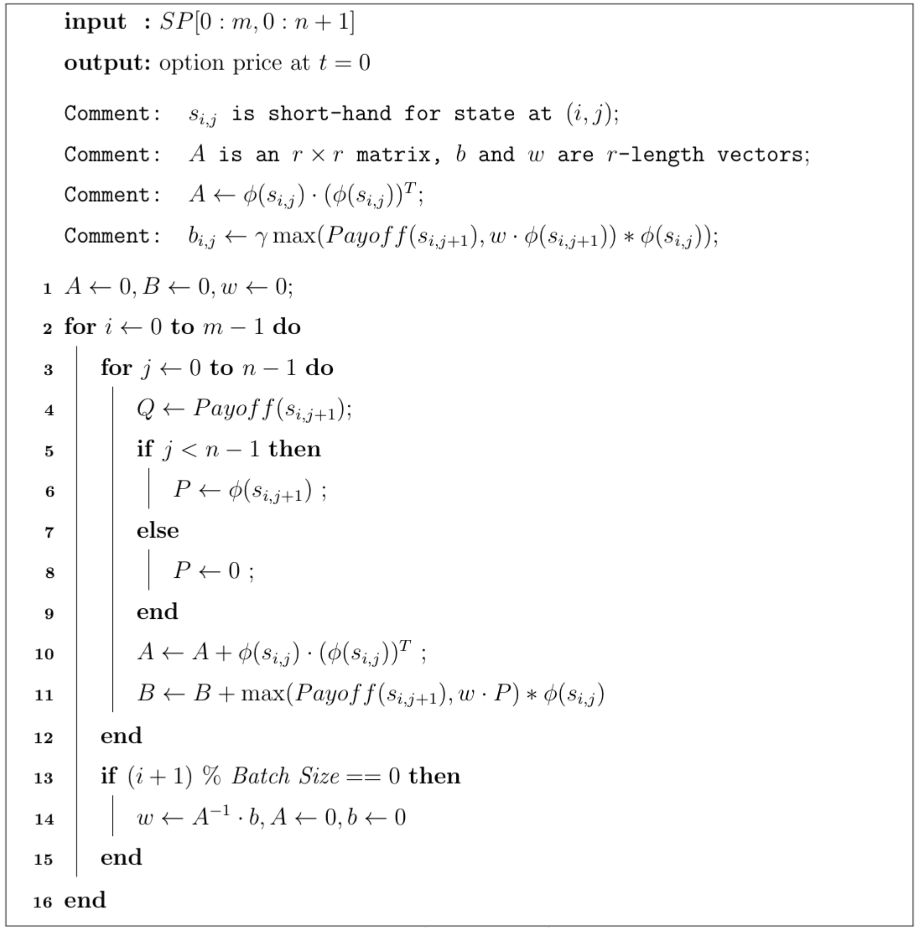
```
input  : $SP[0:m, 0:n+1]$
output: option price at $t = 0$

Comment:  $s_{i,j}$ is short-hand for state at $(i, j)$;
Comment:  $A$ is an $r \times r$ matrix, $b$ and $w$ are $r$-length vectors;
Comment:  $A \leftarrow \phi(s_{i,j}) \cdot (\phi(s_{i,j}) - \gamma \mathbb{1}_{w \cdot \phi(s_{i,j+1}) \geq Payoff(s_{i,j+1})} * \phi(s_{i,j+1}))^T$;
Comment:  $b_{i,j} \leftarrow \gamma \mathbb{1}_{w \cdot \phi(s_{i,j+1}) < Payoff(s_{i,j+1})} * Payoff(s_{i,j+1}) * \phi(s_{i,j})$;

$A \leftarrow 0, B \leftarrow 0, w \leftarrow 0$;
for $i \leftarrow 0$ to $m - 1$ do
    for $j \leftarrow 0$ to $n - 1$ do
        $Q \leftarrow Payoff(s_{i,j+1})$;
        if $j < n - 1$ & $Q \leq w \cdot \phi(s_{i,j+1})$ then
            $P \leftarrow \phi(s_{i,j+1})$ ;
        else
            $P \leftarrow 0$ ;
        end
        if $Q > w \cdot P$ then
            $R \leftarrow Q$ ;
        else
            $R \leftarrow 0$ ;
        end
        $A \leftarrow A + \phi(s_{i,j}) \cdot (\phi(s_{i,j}) - e^{-r_{t_j}(t_{j+1} - t_j)} * P)$;
        $B \leftarrow B + e^{-r_{t_j}(t_{j+1} - t_j)} * R * \phi(s_{i,j})$
    end
    if $(i + 1)$ % $Batch\ Size == 0$ then
        $w \leftarrow A^{-1} \cdot b, A \leftarrow 0, b \leftarrow 0$
    end
end
```

Algorithm 3: Least Squares Policy Iteration Approach

FITTED Q-ITERATION APPROACH

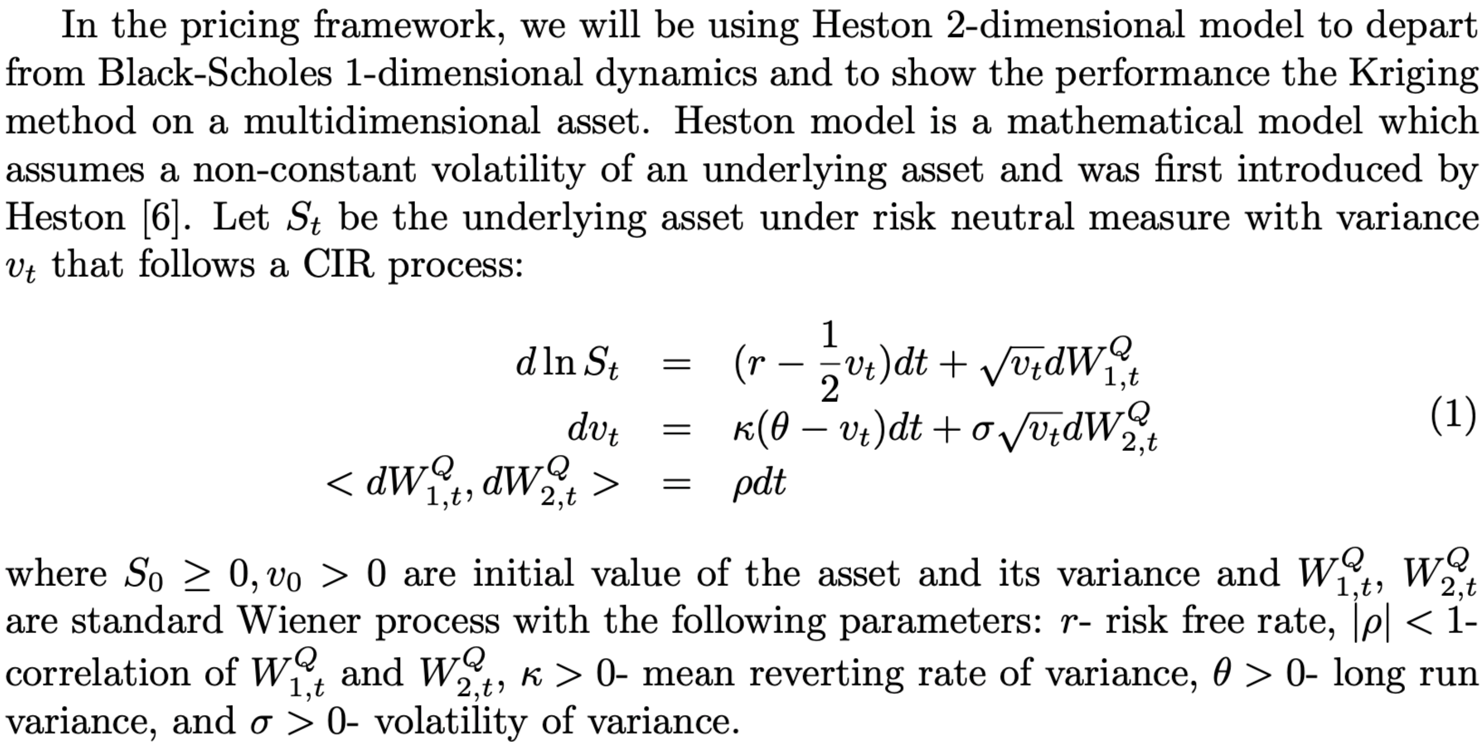


```
input  : $SP[0:m, 0:n+1]$
output: option price at $t = 0$
Comment:  $s_{i,j}$ is short-hand for state at $(i, j)$;
Comment:  $A$ is an $r \times r$ matrix, $b$ and $w$ are $r$-length vectors;
Comment:  $A \leftarrow \phi(s_{i,j}) \cdot (\phi(s_{i,j}))^T$;
Comment:  $b_{i,j} \leftarrow \gamma \max(Payoff(s_{i,j+1}), w \cdot \phi(s_{i,j+1})) * \phi(s_{i,j})$;
$A \leftarrow 0, B \leftarrow 0, w \leftarrow 0$;
for $i \leftarrow 0$ to $m-1$ do
    for $j \leftarrow 0$ to $n-1$ do
        $Q \leftarrow Payoff(s_{i,j+1})$;
        if $j < n-1$ then
            $P \leftarrow \phi(s_{i,j+1})$ ;
        else
            $P \leftarrow 0$ ;
        end
        $A \leftarrow A + \phi(s_{i,j}) \cdot (\phi(s_{i,j}))^T$ ;
        $B \leftarrow B + \max(Payoff(s_{i,j+1}), w \cdot P) * \phi(s_{i,j})$
    end
    if $(i+1)$ % $Batch\ Size == 0$ then
        $w \leftarrow A^{-1} \cdot b, A \leftarrow 0, b \leftarrow 0$
    end
end
```

Algorithm 4: Fitted Q-Iteration Approach

NUMERICAL EXAMPLES

- Simple American Put
- Heston Model



In the pricing framework, we will be using Heston 2-dimensional model to depart from Black-Scholes 1-dimensional dynamics and to show the performance the Kriging method on a multidimensional asset. Heston model is a mathematical model which assumes a non-constant volatility of an underlying asset and was first introduced by Heston [6]. Let S_t be the underlying asset under risk neutral measure with variance v_t that follows a CIR process:

$$\begin{aligned} d\ln S_t &= (r - \frac{1}{2}v_t)dt + \sqrt{v_t}dW^Q_{1,t} \\ dv_t &= \kappa(\theta - v_t)dt + \sigma\sqrt{v_t}dW^Q_{2,t} \\ < dW^Q_{1,t}, dW^Q_{2,t} > &= \rho dt \end{aligned} \tag{1}$$

where $S_0 \geq 0, v_0 > 0$ are initial value of the asset and its variance and $W^Q_{1,t}$, $W^Q_{2,t}$ are standard Wiener process with the following parameters: r- risk free rate, $|\rho| < 1$- correlation of $W^Q_{1,t}$ and $W^Q_{2,t}$, $\kappa > 0$- mean reverting rate of variance, $\theta > 0$- long run variance, and $\sigma > 0$- volatility of variance.

SIMPLE AMERICAN PUT

Step 4

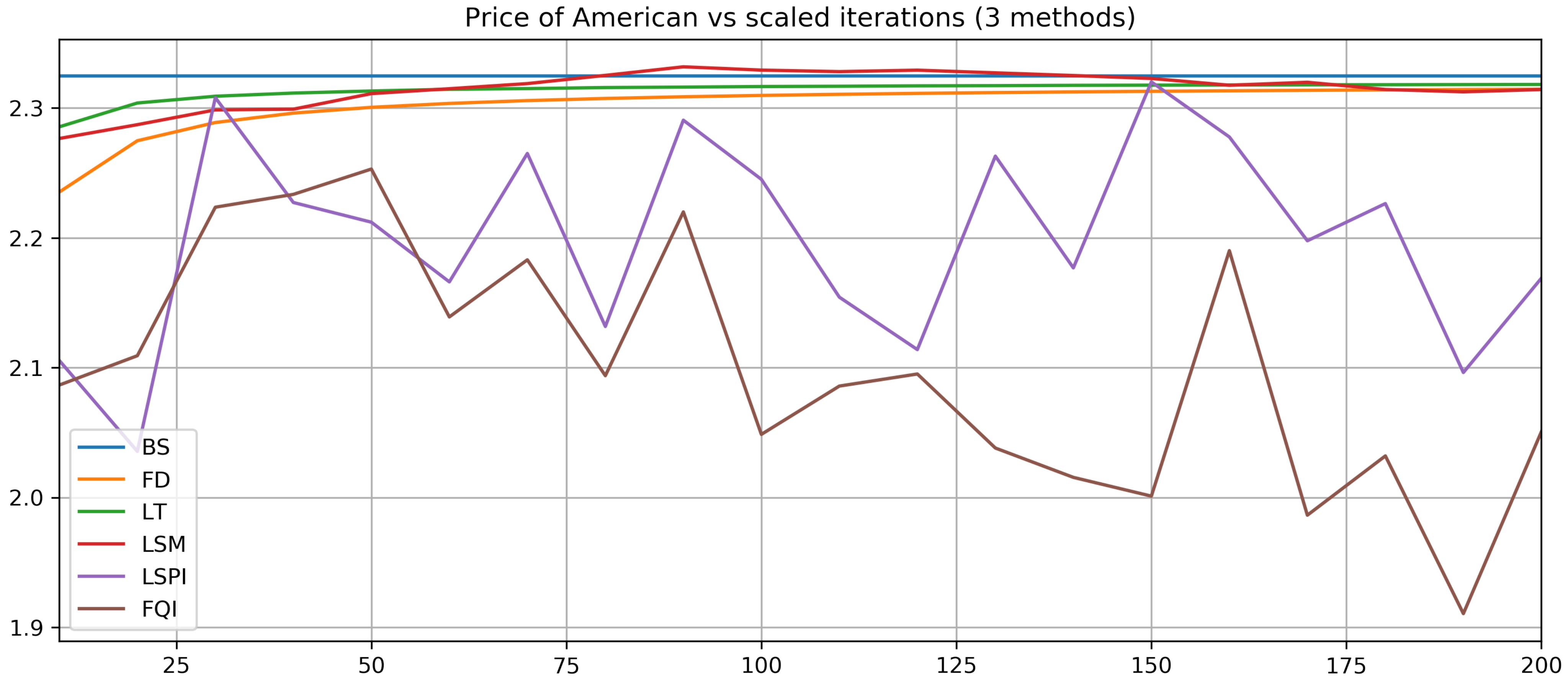
non machine learning

machine learning.

S0	sigma	T	BS	LT	FD	LSM	GPR	FQI	LSPI
36	**0.2**	**1**	4.4840	*4.48 (0.01%)*	*4.46 (-0.44%)*	*4.48 (0.02%)*	*4.38 (-2.38%)*	*4.38 (-2.37%)*	*4.37 (-2.53%)*
36	**0.2**	**2**	4.8501	*4.85 (-0.08%)*	*4.83 (-0.44%)*	*4.83 (-0.31%)*	*4.81 (-0.91%)*	*4.85 (0.01%)*	*4.79 (-1.15%)*
36	**0.4**	**1**	7.1098	*7.1 (-0.14%)*	*7.08 (-0.47%)*	*7.09 (-0.21%)*	*6.8 (-4.38%)*	*6.87 (-3.39%)*	*6.78 (-4.64%)*
36	**0.4**	**2**	8.5195	*8.51 (-0.14%)*	*8.49 (-0.40%)*	*8.51 (-0.13%)*	*8.2 (-3.70%)*	*7.94 (-6.77%)*	*8.22 (-3.56%)*
38	**0.2**	**1**	3.2568	*3.25 (-0.12%)*	*3.24 (-0.64%)*	*3.25 (-0.24%)*	*3.14 (-3.47%)*	*3.17 (-2.78%)*	*3.01 (-7.46%)*
38	**0.2**	**2**	3.7540	*3.75 (-0.15%)*	*3.73 (-0.54%)*	*3.74 (-0.26%)*	*3.65 (-2.90%)*	*3.57 (-4.90%)*	*3.67 (-2.17%)*
38	**0.4**	**1**	6.1533	*6.18 (0.44%)*	*6.12 (-0.51%)*	*6.13 (-0.30%)*	*5.87 (-4.66%)*	*5.89 (-4.28%)*	*5.86 (-4.73%)*
38	**0.4**	**2**	7.6713	*7.69 (0.23%)*	*7.65 (-0.34%)*	*7.69 (0.20%)*	*6.91 (-9.88%)*	*6.89 (-10.13%)*	*7.41 (-3.35%)*
40	**0.2**	**1**	2.3245	*2.31 (-0.49%)*	*2.3 (-1.03%)*	*2.31 (-0.58%)*	*2.21 (-4.77%)*	*2.25 (-3.08%)*	*2.21 (-4.84%)*
40	**0.2**	**2**	2.8965	*2.89 (-0.39%)*	*2.87 (-0.77%)*	*2.89 (-0.23%)*	*2.58 (-10.88%)*	*2.55 (-11.94%)*	*2.77 (-4.21%)*
40	**0.4**	**1**	5.3260	*5.3 (-0.44%)*	*5.29 (-0.74%)*	*5.32 (-0.21%)*	*5.1 (-4.32%)*	*5.08 (-4.68%)*	*5.1 (-4.30%)*
40	**0.4**	**2**	6.9351	*6.91 (-0.31%)*	*6.89 (-0.60%)*	*6.94 (0.04%)*	*7.05 (1.60%)*	*6.86 (-1.06%)*	*7.32 (5.51%)*
42	**0.2**	**1**	1.6204	*1.62 (0.22%)*	*1.61 (-0.94%)*	*1.62 (-0.10%)*	*1.59 (-1.94%)*	*1.59 (-1.94%)*	*1.6 (-1.13%)*
42	**0.2**	**2**	2.2187	*2.22 (-0.07%)*	*2.2 (-0.71%)*	*2.23 (0.31%)*	*1.86 (-16.38%)*	*1.85 (-16.70%)*	*2.06 (-7.24%)*
42	**0.4**	**1**	4.5884	*4.61 (0.55%)*	*4.56 (-0.67%)*	*4.59 (0.10%)*	*4.43 (-3.38%)*	*4.41 (-3.91%)*	*4.43 (-3.36%)*
42	**0.4**	**2**	6.2464	*6.26 (0.29%)*	*6.22 (-0.42%)*	*6.27 (0.38%)*	*5.65 (-9.60%)*	*5.3 (-15.17%)*	*5.96 (-4.63%)*
44	**0.2**	**1**	1.1118	*1.12 (0.84%)*	*1.1 (-1.02%)*	*1.12 (0.30%)*	*1.12 (0.65%)*	*1.13 (1.30%)*	*1.09 (-1.82%)*
44	**0.2**	**2**	1.6921	*1.7 (0.31%)*	*1.68 (-0.62%)*	*1.69 (0.17%)*	*1.43 (-15.38%)*	*1.33 (-21.44%)*	*1.47 (-13.36%)*
44	**0.4**	**1**	3.9529	*3.96 (0.22%)*	*3.92 (-0.72%)*	*3.96 (0.17%)*	*3.87 (-2.12%)*	*3.86 (-2.29%)*	*3.87 (-1.98%)*
44	**0.4**	**2**	5.6511	*5.65 (0.01%)*	*5.62 (-0.62%)*	*5.67 (0.26%)*	*4.93 (-12.76%)*	*4.64 (-17.83%)*	*5.29 (-6.41%)*

Step 4

SIMPLE AMERICAN PUT OPTION



Summary

- The non machine learning method is more stable and efficient in time.
- The machine learning method is fair enough but needs a lot of parameter tuning. Takes longer time and may result in fluctuated estimation.
- Machine learning performs better when it is in high-dimensional case, but still, much slower than the mainstream way.

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS

THANKS FOR LISTENING