American Put Options Pricing

Grégoire Mourre, Jessica Iacob

January 2024

Introduction

American options form a family of derivatives which give to the owner the right to exercise at any time before the maturity T. The general form of the price at time 0 of an American put option is the following:

$$\sup_{\tau \in \mathcal{T}} \mathbb{E}[e^{-r\tau} g(X_\tau)]$$

where $g : x \mapsto (K - x)_+ := \max(0, K - x)$ and r is the risk free rate (supposed constant).
We discretize in time the problem : $\mathcal{T}$ by $\bar{\mathcal{T}}$ the $\{t_0, \cdots, t_N\}$-valued stopping times, with $0 = t_0 < t_1 < \cdots < t_N = T$ (the maturity) and we obtain by Dynamic Programming Principle :

$$\begin{aligned} V_N(X_{t_N}) &= Y_{t_N} \\ V_k(X_{t_k}) &= \max\left(Y_{t_k}, \mathbb{E}[V_{k+1}(X_{t_{k+1}})|X_{t_k}]\right) \end{aligned}$$

with $(V_i)_{i\in\{0,\cdots,N\}}$ the Snell envelope and in discounted terms $Y_t = e^{-rt} g(X_t),\ \forall t \in [0, T]$.

Duality

In this part, we will derive a dual problem from the primal problem

$$V_0(X_{t_0}) = \sup_{\tau \in \bar{\mathcal{T}}} \mathbb{E}[Y_\tau].$$

More precisely, we will show

$$\sup_{\tau \in \bar{\mathcal{T}}} \mathbb{E}[Y_\tau] = \inf_{M \in \mathcal{M}_0} \mathbb{E}[\max_{k \in \{0, \cdots, N\}} \left(Y_{t_k} - M_k\right)]$$

Duality

First inequality :
For each martingale $(M_i)_{i\in\{0,\cdots,N\}}$ starting from 0 at time 0 and $\tau \in \bar{\mathcal{T}}$, we have

$$\mathbb{E}[Y_\tau] = \mathbb{E}[Y_\tau - M_\tau] \leq \mathbb{E}[\max_{k\in\{0,\cdots,N\}}\big(Y_{t_k} - M_k\big)].$$

So, we get

$$\sup_{\tau\in\bar{\mathcal{T}}} \mathbb{E}[Y_\tau] \leq \inf_{M\in\mathcal{M}_0} \mathbb{E}[\max_{k\in\{0,\cdots,N\}}\big(Y_{t_k} - M_k\big)].$$

Duality

Second inequality :

Let M be such that $M_0 = 0$ and $M_k = \Delta_1 + \cdots + \Delta_k$ for all $k \in \{1, \cdots, N\}$, with

$$\Delta_k = V_k(X_{t_k}) - \mathbb{E}[V_k(X_{t_k})|X_{t_{k-1}}].$$

Such a process is indeed a martingale: for all $k \in \{1, \cdots, N\}$,

$$\begin{aligned}
\mathbb{E}[M_k - M_{k-1}|\sigma(X_0, \cdots, X_{t_{k-1}})] &= \mathbb{E}[\Delta_k|\sigma(X_0, \cdots, X_{t_{k-1}})] \\
&= \mathbb{E}[V_k(X_{t_k})|X_{t_{k-1}}] - \mathbb{E}[V_k(X_{t_k})|X_{t_{k-1}}] \\
&= 0.
\end{aligned}$$

Duality

Second inequality :
Let us show by backward induction that for all $k \in \{0, \cdots, N\}$,

$$V_k(X_{t_k}) = \max\left\{Y_{t_k},\ Y_{t_{k+1}} - \Delta_{k+1},\ \cdots,\ Y_{t_N} - \Delta_N - \cdots - \Delta_{k+1}\right\}.$$

At time t_N,

$$V_N(X_{t_N}) = Y_{t_N}.$$

Let us suppose that the formula is verified for $k+1 \in \{1, \cdots, N\}$ fixed.

$$\begin{aligned}
V_k(X_{t_k}) &= \max\left\{Y_{t_k}, \mathbb{E}[V_{k+1}(X_{t_{k+1}})|X_{t_k}]\right\} \\
&= \max\left\{Y_{t_k}, V_{k+1}(X_{t_{k+1}}) - \Delta_{k+1}\right\} \\
&= \max\left\{Y_{t_k}, \max\left\{Y_{t_{k+1}},\ \cdots,\ Y_{t_N} - \Delta_N - \cdots - \Delta_{k+2}\right\} - \Delta_{k+1}\right\} \\
&= \max\left\{Y_{t_k}, Y_{t_{k+1}} - \Delta_{k+1},\ \cdots,\ Y_{t_N} - \Delta_N - \cdots - \Delta_{k+1}\right\}.
\end{aligned}$$

Duality

Taking $k = 0$, we get:

$$\begin{aligned} V_0(X_{t_0}) &= \max\left\{Y_{t_0}, Y_{t_1} - \Delta_1, \ \cdots, \ Y_{t_N} - \Delta_N - \cdots - \Delta_1\right\} \\ &= \max\left\{Y_{t_0}, Y_{t_1} - M_1, \ \cdots, \ Y_{t_N} - M_N\right\} \\ &= \max_{k\in\{0,\cdots,N\}}\left\{Y_{t_k} - M_k\right\}. \end{aligned}$$

Thus,

$$V_0(X_{t_0}) = \max_{k\in\{0,\cdots,N\}}\left\{Y_{t_k} - M_k\right\} = \inf_{M\in\mathcal{M}_0}\mathbb{E}[\max_{k\in\{0,\cdots,N\}}\left(Y_{t_k} - M_k\right)].$$

Duality

To conclude, we had a primal maximisation problem:

$$V_0(X_{t_0}) = \sup_{\tau \in \bar{\mathcal{T}}} \mathbb{E}[Y_\tau].$$

We transformed it into a dual minimisation problem:

$$\sup_{\tau \in \bar{\mathcal{T}}} \mathbb{E}[Y_\tau] = \inf_{M \in \mathcal{M}_0} \mathbb{E}[\max_{k \in \{0,\cdots,N\}} \left(Y_{t_k} - M_k\right)].$$

This infimum is reached with the martingale M: $M_0 = 0$ and $M_k = \Delta_1 + \cdots + \Delta_k$ for all $k \in \{1, \cdots, N\}$, with

$$\Delta_k = V_k(X_{t_k}) - \mathbb{E}[V_k(X_{t_k})|X_{t_{k-1}}].$$

Thus, by approximating the optimal martingale with different methods, we will get a valid upper bound for our initial problem.

Approximation of Continuation Value Functions

We have to approximate the quantities:

$$C_k : x \mapsto \mathbb{E}[V_{k+1}(X_{t_{k+1}})|X_{t_k} = x], \quad \forall k \in \{0, \cdots, N-1\},$$

which are called the continuation value functions. By definition of the conditional expectation in $\mathbb{L}^2$,

$$\begin{aligned}
\mathbb{E}[V_{k+1}(X_{t_{k+1}})|X_{t_k}] &= \underset{Z \in \mathbb{L}^2(\Omega, \sigma(X_{t_k}), \mathbb{P})}{\arg\min} \mathbb{E}\Big[\big(V_{k+1}(X_{t_{k+1}}) - Z\big)^2\Big] \\
&= \underset{\Phi(X_{t_k}) \in \mathbb{L}^2}{\arg\min} \mathbb{E}\Big[\big(V_{k+1}(X_{t_{k+1}}) - \Phi(X_{t_k})\big)^2\Big].
\end{aligned}$$

Two ideas :

1. Non-parametric regression: $\tilde{C}_k : x \mapsto \sum_{i=1}^{I} \alpha_i \phi_i(x)$, for all $k \in \{0, \cdots, N-1\}$ where $I \in \mathbb{N}^*$ and $\phi_1, \cdots, \phi_I$ are base functions of $\mathbb{L}^2$,
2. Neural network: a fully connected neural network with 4 hidden layers which have respective width 3, 5, 5 and 3

Martingales from Approximate Value Functions

Reminder: By Duality : $V_0(X_{t_0}) = \inf_{M \in \mathcal{M}_0} \mathbb{E}[\max_{k \in \{0,\cdots,N\}}(Y_{t_k} - M_k)]$ and the infimum is attained with the martingale $(M_k = \Delta_1 + \cdots + \Delta_k)_{k \in \{0,\cdots,N\}}$ where $\Delta_k = V_k(X_{t_k}) - \mathbb{E}[V_k(X_{t_k})|X_{t_{k-1}}]$ for all $k \in \{1, \cdots, N\}$. Denote $C_{k-1} : x \mapsto \mathbb{E}[V_k(X_{t_k})|X_{t_{k-1}} = x]$ the continuation value. So we can re-write $\Delta_k = V_k(X_{t_k}) - C_{k-1}(X_{t_{k-1}})$.

The key of this method, is to approximate Δ_k by $\tilde{\Delta}_k = \tilde{V}_k(X_{t_k}) - \mathbb{E}[\tilde{V}_k(X_{t_k})|X_{t_{k-1}}]$ instead of $\tilde{\Delta}_k = \tilde{V}_k(X_{t_k}) - \tilde{C}_{k-1}$ with $\tilde{C}_{k-1}$ as above.

Thus the approximated Snell envelop is : $\tilde{V}_k(X_{t_k}) = \max(Y_{t_k}, \tilde{C}_k(X_{t_k}))$ (this time $\tilde{C}_k(X_{t_k})$ is as above).

Martingales from Stopping Rules

Here, instead of approximate Δ_k by $\tilde{\Delta}_k = \tilde{V}_k(X_{t_k}) - \mathbb{E}[\tilde{V}_k(X_{t_k})|X_{t_{k-1}}]$, we will use that the Δ_k in the optimal martingale re-writes as:

$$\begin{aligned} \Delta_k &= V_k(X_{t_k}) - \mathbb{E}[V_k(X_{t_k})|X_{t_{k-1}}] \\ &= \mathbb{E}[Y_{\tau_k}|X_{t_k}] - \mathbb{E}[Y_{\tau_k}|X_{t_{k-1}}], \end{aligned}$$

with the optimal stopping policy in discrete time given by the dynamical programming principle:

$$\begin{aligned} \tau_i &= \min\left\{t_k \in \{t_i, \cdots, t_N\} \text{ such that } Y_{t_k} \geq \mathbb{E}[V_{k+1}(X_{t_{k+1}})|X_{t_k}]\right\} \\ &= \min\left\{t_k \in \{t_i, \cdots, t_N\} \text{ such that } Y_{t_k} \geq C_k(X_{t_k})\right\}. \end{aligned}$$

We will use an approximation of the optimal stopping rule: $\tilde{\tau}_i = \min\left\{t_k \in \{t_i, \cdots, t_N\} \text{ such that } Y_{t_k} \geq \tilde{C}_k(X_{t_k})\right\}$, and approximate Δ_k by $\tilde{\Delta}_k = \mathbb{E}[Y_{\tilde{\tau}_k}|X_{t_k}] - \mathbb{E}[Y_{\tilde{\tau}_k}|X_{t_{k-1}}]$.

Martingales from Approximate Value Functions and from Stopping Rules - Implementation

Martingales from Approximate Value Functions

1. Simulate a path $X_{t_0}, \cdots, X_{t_N}$ for the underlying Markov chain
2. At each step X_{t_k}, $k = 1, \cdots, N-1$ of the Markov chain:
    - Compute $\tilde{V}_k(X_{t_k}) = \max(Y_{t_k}, \tilde{C}_k(X_{t_k}))$
    - Generate m successors $X^1_{t_k}, \cdots, X^m_{t_k}$ of $X_{t_{k-1}}$
    - Compute $\frac{1}{n}\sum_{j=1}^m \tilde{V}_k(X^j_{t_k})$ to approximate $\mathbb{E}[\tilde{V}_k(X_{t_k})|X_{t_{k-1}}]$
    - Set $\tilde{\Delta}_k = \tilde{V}_k(X_{t_k}) - \frac{1}{n}\sum_{j=1}^m \tilde{V}_k(X^j_{t_k})$
3. Do (2) again for step X_{t_N} but with $\tilde{V}_N(X_{t_N}) = Y_{t_N}$ and $\tilde{V}_N(X^j_{t_N}) = Y^j_{t_N}$
4. Sum to obtain the martingale $(\tilde{M}_k = \tilde{\Delta}_1 + \cdots + \tilde{\Delta}_k)_{k\in\{0,\cdots,N\}}$
5. Evaluate $\max_{k\in\{0,\cdots,N\}}(Y_{t_k} - \tilde{M}_k)$

Martingales from Stopping Rules

1. Simulate a path $X_{t_0}, \cdots, X_{t_N}$ for the underlying Markov chain
2. At each step X_{t_k}, for $k = 0, 1, \cdots, N-1$:
    - Simulate m subpaths $(X^1_{t_{k+j}})_{j\in\{1,\cdots,m-k\}}, \cdots, (X^m_{t_{k+j}})_{j\in\{1,\cdots,m-k\}}$ starting from X_{t_k}
    - Evaluate $Y^j_{\tilde{\tau}_{k+1}}$ for $j = 1, \cdots, m$, and use $\frac{1}{m}\sum_{j=1}^m Y^j_{\tilde{\tau}_{k+1}}$ as an estimator of $\mathbb{E}[Y_{\tilde{\tau}_{k+1}}|X_{t_k}]$
    - Evaluate the quantity

    $$W_k := \begin{cases} Y_{t_k} & \text{if } Y_{t_k} \geq \tilde{C}_k(X_{t_k}) \\ \frac{1}{m}\sum_{j=1}^m Y^j_{\tilde{\tau}_{k+1}} & \text{otherwise} \end{cases}$$

    and use it as an estimator of $\mathbb{E}[Y_{\tilde{\tau}_k}|X_{t_k}]$
    - Set $\tilde{\Delta}_k := W_k - \frac{1}{m}\sum_{j=1}^m Y^j_{\tilde{\tau}_k}$ (with the $(Y^j_{\tilde{\tau}_k})_{j\in\{1,\cdots,m\}}$ simulated from $X_{t_{k-1}}$ in the previous step)
3. Sum to obtain the martingale $(\tilde{M}_k = \tilde{\Delta}_1 + \cdots + \tilde{\Delta}_k)_{k\in\{0,\cdots,N\}}$
4. Evaluate $\max_{k\in\{0,\cdots,N\}}(Y_{t_k} - \tilde{M}_k)$

Numerical simulations of the Monte-Carlo methods

Let us recap all the parameters.

Algorithms	Approximate Value Function	Stopping Rule
Base functions $\phi_1, \cdots, \phi_l$	Function `phi`	Function `phi`
Number of time $t_0, \cdots, t_N$	`m`	`m`
Number of paths simulated	`n`	`n`
Number of subpaths simulated	`n2`	`n_subpaths`

Table of all the parameters in the algorithms and their name in the code.

We will price an American put option with the parameters of the model: $K = 100, r = 0.06, T = 0.5, \sigma = 0.4$ and $X_{t_0} = 80$. The reel price of the option is ≈ 21.6059 (see [Rogers, 2002], Table 4.1).

Impact of the base functions $\phi_1, \cdots, \phi_l$ (n = 10000, m = 12, n2 = 1000, n_subpaths = 500)

With the canonical basis of $P[X]$ + powers of g:

- $\mathcal{A}_{1,1} = \text{Span}(1, X, X^2, X^3)$
- $\mathcal{A}_{1,2} = \text{Span}(1, X, X^2, X^3, g)$
- $\mathcal{A}_{1,3} = \text{Span}(1, X, X^2, X^3, X^4, g)$
- $\mathcal{A}_{1,4} = \text{Span}(1, X, X^2, X^3, g, g^2)$
- $\mathcal{A}_{1,5} = \text{Span}(1, X, X^2, X^3, g, g^2, g^3)$

-	Approximate Value Functions			Stopping Rule		
-	Upper bound	STD	Time	Upper bound	STD	Time
$\mathcal{A}_{1,1}$	21.82199	0.00841	3.4 s	21.73372	0.00631	29.6 s
$\mathcal{A}_{1,2}$	21.72501	0.00738	4.0 s	21.74073	0.00609	28.9 s
$\mathcal{A}_{1,3}$	21.68976	0.00619	5.0 s	21.72331	0.00607	32.8 s
$\mathcal{A}_{1,4}$	21.71974	0.00748	4.3 s	21.72331	0.00598	30.5 s
$\mathcal{A}_{1,5}$	21.67081	0.00549	5.5 s	21.71392	0.00579	35.2 s

Impact of $\mathtt{m}$ $(\mathcal{A}_{1,5}, \mathtt{n} = 10000, \mathtt{n2} = 1000, \mathtt{n\_subpaths} = 500)$

-	Upper bound	STD	Time	Upper bound	STD	Time
$\mathtt{m} = 3$	21.41145	0.00452	1.2 s	21.44011	0.00605	1.8 s
$\mathtt{m} = 5$	21.53178	0.00479	2.1 s	21.58186	0.00594	4.2 s
$\mathtt{m} = 10$	21.62896	0.00492	4.5 s	21.68813	0.00587	20.6 s
$\mathtt{m} = 11$	21.66594	0.00556	5.1 s	21.69998	0.00586	27.1 s
$\mathtt{m} = 12$	21.67081	0.00549	5.6 s	21.71392	0.00579	43.1 s
$\mathtt{m} = 15$	21.68842	0.00571	6.9 s	21.75519	0.00596	106.8 s
$\mathtt{m} = 20$	21.71342	0.00596	10.1 s	21.78757	0.00591	216.6 s

Impact of n2 and n_subpaths ($\mathcal{A}_{1,5}, \mathtt{n} = 10000, \mathtt{m} = 12$)

-	Approximate Value Functions			Stopping Rule		
n2/ n_s	UB	STD	Time	UB	STD	Time
200 / 100	21.88965	0.00914	1.2 s	22.18451	0.01307	6.2 s
400 / 200	21.75476	0.00709	2.1 s	21.89817	0.00924	10.4 s
600 / 300	21.70843	0.00626	3.2	21.80254	0.00751	18.7 s
800 / 400	21.68687	0.00580	4.2	21.74008	0.00653	25.2 s
1000 / 500	21.67081	0.00549	5.3 s	21.71392	0.00579	37.4 s
1500 / 600	21.65134	0.00512	8.3 s	21.66960	0.00489	59.2 s

Impact of the parameter n ($\mathcal{A}_{1,5}$, `n2 = 1000, n_subpaths = 500, m = 12`)

-	Approximate Value Functions			Stopping Rule		
-	UB	STD	Time	UB	STD	Time
n = 1000	21.99514	0.02982	0.5 s	21.78501	0.02209	2.7 s
n = 3000	21.70631	0.01143	1.7 s	21.75219	0.01089	7.3 s
n = 6000	21.68539	0.00763	3.3 s	21.73110	0.00769	15.8 s
n = 10000	21.67081	0.00549	5.6 s	21.71392	0.00579	36.7 s
n = 15000	21.64745	0.00390	9.3 s	21.71359	0.00481	82.4 s
n = 20000	21.64141	0.00348	14.2 s	21.70614	0.00411	180.1 s

Numerical simulations of the Monte-Carlo methods

With a neural network with 4 hidden layers and with widths 3, 5, 5 and 3 (the other parameters are $\texttt{n} = 10000$, $\texttt{m} = 12$, $\texttt{n2} = 1000$, $\texttt{n\_subpaths} = 500$), we find:

- Martingale from Approximate Value Functions: an upper bound of 21.77267 and a standard deviation of 0.00637
- Martingale from Stopping Rule: an upper bound of 21.75629 and a standard deviation of 0.00922

Finite differences method

We consider X_t the log of the solution of Black-Scholes SDE (under the $\mathbb{Q}$-EMM), i.e. $dX_t = \left(r - \frac{\sigma^2}{2}\right)dt + \sigma dW_t$ and $\Phi(x) = (K - e^x)_+$ instead of g above. And we obtain the variational inequality

$$\max\left(\partial_t u(t,x) + \mathcal{A}u(t,x), \Phi(x) - u(t,x)\right) = 0,$$

where $\mathcal{A}f = \frac{\sigma^2}{2}\partial^2_{xx}f + (r - \frac{\sigma^2}{2})\partial_x f - rf$. By discretization in a θ-scheme, we have

$$\max\left((I - h\theta A_\delta)u^n - (I + h(1-\theta)A_\delta)u^{n+1}, \Phi_\delta - u^n\right) = 0,$$

where $A_\delta \in \mathbb{R}^{I\times I}$ is a tridiagonal matrix with coefficients $a_{i,i-1} = \frac{\sigma^2}{2\delta^2} - \frac{1}{2\delta}\left(r - \frac{\sigma^2}{2}\right)$, $a_{i,i} = -\left(\frac{\sigma^2}{\delta^2} + r\right)$, $a_{i,i+1} = \frac{\sigma^2}{2\delta^2} + \frac{1}{2\delta}\left(r - \frac{\sigma^2}{2}\right)$.

Conditions to have properties of monotonicity, stability and consistency : $\sigma^2(1-\theta)h \leq \delta^2$ and $\delta|r - \frac{\sigma^2}{2}| \leq \sigma^2$. Moreover, if $\theta = 1$, the scheme is unconditionnaly converging. Else, the convergence holds if $\lim_{h\to 0, k\to 0} \frac{h}{\delta^2} = 0$.

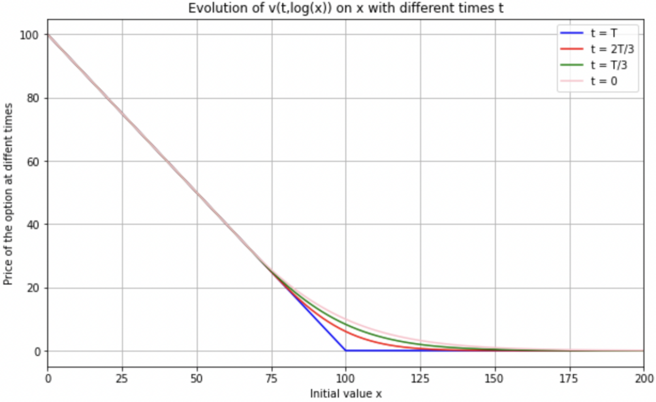
Finite difference method - Implementation

For the discretization, we localize the inequality to the space $\mathcal{O}_z = (-z, z)$ and add additional restrictions like Neumann condition $\partial_x v(t, \pm z) = 0$.

1. Define discretization parametres : $h = \frac{T}{m}$ (time steps) and $\delta = \frac{2z}{l+1}$ (space steps), $(u_i^n)_{0 \leq n \leq m,\ 0 \leq i \leq l+1}$ is an approximation of $(u(t,x))_{t \in [0,T], x \in (-z,z)}$ and we set $u^m = \Phi_\delta = (\Phi(x_i))_{1 \leq i \leq l}$.
2. Define matrix A_δ with the coefficient given above, the terminal time $u^m = \Phi_\delta$ and Neumann conditions $u_0^n = u_1^n$ and $u_l^n = u_{l+1}^n$ for each time $n \in \{0, \cdots, m\}$.
3. For each time, knowing u^{n+1}, solve $(I - h\theta A_\delta)u^{n+\frac{1}{2}} - (I + h(1-\theta)A_\delta)u^{n+1} = 0$.
4. The solution we expect is given by $u^n := \max\left(u^{n+\frac{1}{2}}, \Phi_\delta\right)$ at each time (with $u^{n+\frac{1}{2}}$ found at previous step).

Numerical simulations of the Finite differences method (1)

Watching the evolution of the solution given by this method (for the 3 scheme) at several times, we can observe how the finite difference method converge to the solution when the parameters are well set i.e $\delta|r - \frac{\sigma^2}{2}| \leq \sigma^2$ and $\sigma^2(1-\theta)h \leq \delta^2$:



Numerical simulations of the Finite differences method (2)

Taking $m = 1$, we observe the method does not converge to the solution. Indeed, time discretization is not small enough as $\frac{h}{\delta^2} \approx 1252$ which is quite too large. For $m = 10$, we have $\frac{h}{\delta^2} \approx 125.2$. It seems to be too large but the solution by approximation looks close to the solution. And for $m = 1000$, the approximation is smoother and $\frac{h}{\delta^2} \approx 1.25$ which explains it.

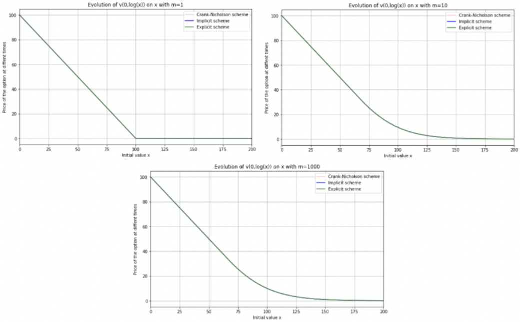


Figure 6: Evolution of $v(0, \log(x))$ for different space steps

Numerical simulations of the Finite differences method (3)

For I = 15, the finite difference method is not a good approximation. Indeed the space step $\delta \approx 1.25$ is too large, so space discretization is not precize enough. When I = 100, the approximation is close to the solution but not enough smooth and $\delta \approx 0.2$. Taking I = 1000, we obtain $\delta \approx 0.02$ and the approximation is very close to the solution.

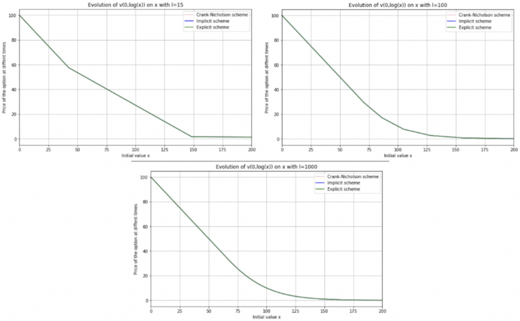


Figure 5: Evolution of $v(0, \log(x))$ for different time steps

Conclusion

Martingale from Approximate Value Functions

- (+) linear complexity with respect to most of the parameters
- (−) highly sensitive to most of the parameters (the base function family, `m`, `n` or `n2`)

Martingale from Stopping Rule

- (−) exponential complexity with respect to some of the parameters (`m` and `n`)
- (+) not too sensitive to some of the parameters (the base function family, `n`)

Finite difference method

- (+) Extremely fast to compute
- (+) Gives the value of the option at time 0 for all the value of x
- (−) Gives a single value and not an confidence interval

Conclusion

What we could have done:

- Use Thomas' algorithm to inverse the tridiagonal matrix (FDS)
- Different boundary conditions, such as the Dirichlet conditions (FDS)
- Try variance reduction methods (Monte Carlo)
- Price a multi-asset put option (Monte Carlo)
- Choose a better structure for the neural network (Approximate Value Functions)
- Try the tree-based pricing techniques

References

Claise, J. (2021).
Monte Carlo and Finite Difference Methods with Applications in Finance.
Lecture notes for M2 M.A.TH./MASEF.

Glasserman, P. (2004).
Monte Carlo methods in financial engineering.
Springer, New York.

Lamberton, D. and Lapeyre, B. (2008).
Introduction to stochastic calculus applied in finance.
Chapman & Hall/CRC Financial Mathematics Series. Chapman & Hall/CRC, Boca Raton, FL, second edition.

Rogers, L. (2002).
Monte carlo valuing of american options.
Mathematical Finance, 12:271–286.