American options

Is the same situation possible for an American option?
Consider the following situation:

- American put option for a certain asset
- strike price $K = 50$
- the market price of the asset is $S = 40$
- could the price of the option be equal to 5?

No! Otherwise we could

- buy the option -5
- buy the asset -40
- exercise the option $+50$
- this would imply a risk-free profit (arbitrage) ($+5$), which is impossible.

... and for a call option?

Consider the following situation:

- American call option for a certain asset
- strike price $K = 30$
- the market price of the asset is $S = 40$
- could the price of the option be equal to 5?

No! Otherwise we could

- buy the option -5
- exercise the option -30
- sell the asset $+40$
- this would also imply a risk-free profit (arbitrage) ($+5$), which is impossible.

An American option shall have at least the value of the payoff. Otherwise, this would be an arbitrage opportunity of making a risk-free profit. Therefore,

$$V_P^{Am}(S,t) \geq (K-S)^+,\ \forall (S,t) \tag{47}$$

and

$$V_C^{Am}(S,t) \geq (S-K)^+,\ \forall (S,t). \tag{48}$$

The inequality in the case of an American put option is illustrated in next Figure.

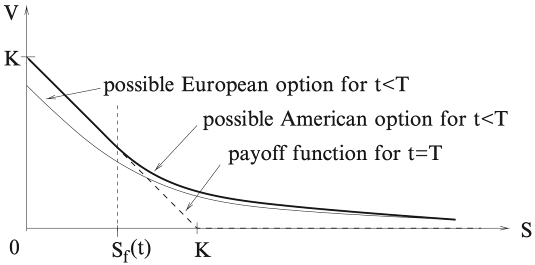
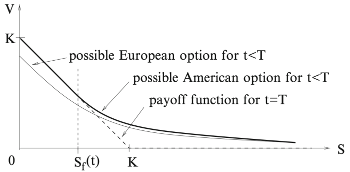


Figure: Plot of possible values $V_P^{Eur}(S,t)$ and $V_P^{Am}(S,t)$ at a certain $t < T$ and plot of the payoff function for t=T.

The calculation of the value of an American option will imply to solve an initial/terminal value problem, as in the case of a European option.

The boundary condition for which we impose $V = 0$ is similar in both cases, but the other boundary condition will be of different nature.

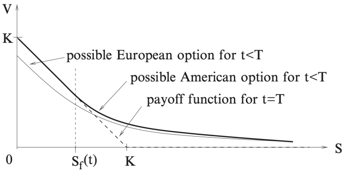
For instance in the case of a put option, without the possibility of early exercise, for some $t < T$, the value of the option may be smaller than the payoff, for sufficiently small $S > 0$.



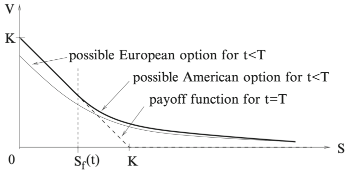
This situation cannot occur in the case of an American option, by (47). Indeed, for sufficiently small S we will have

$$V_P^{Am}(S,t) = (K - S)^+ = K - S.$$

The typical situation is illustrated in the Figure.



The value of the option coincides with the payoff, for small values of S up to a certain critical value that we will be denoted by $S_f(t)$.

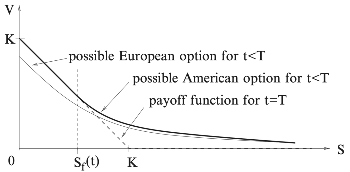


For $S > S_f(t)$, we will have the strict inequality $V_P^{Am}(S,t) > (K-S)^+$. Thus, this contact point $S_f(t)$ is defined by

$$V_P^{Am}(S,t) > (K-S)^+ \quad \text{for} \quad S > S_f(t), \tag{49}$$

$$V_P^{Am}(S,t) = K-S \quad \text{for} \quad S \leq S_f(t). \tag{50}$$

For $S < S_f(t)$, the value V_P^{Am} equals the straight line defined by the payoff function and we do not need to calculate anything and for a fixed value of t, the boundary condition shall be imposed, not at $S=0$, but for $S = S_f(t)$. This situation holds for any $t < T$, where the contact point $S_f(t)$ varies with t.



Therefore, for all $0 \leq t < T$, the contact points form a curve

$$\Gamma_{S_f} = \left\{(S,t) \in \mathbb{R}^2 : t \in [0,T],\ S = S_f(t)\right\}$$

that separates two regions, one for which the value of the option equals the payoff (stopping region) and another region for which the value of the option is larger than the payoff (continuation region).

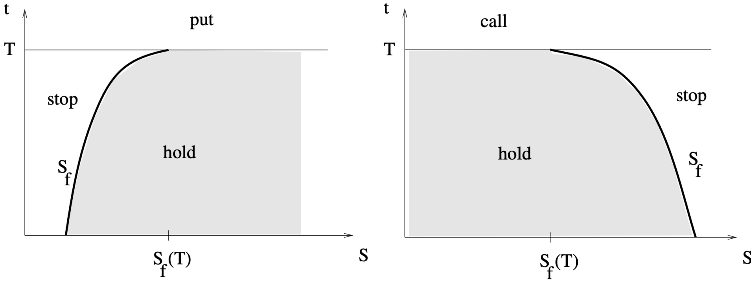


Figure: Continuation (shaded region) and stopping regions and plot of the curve Γ_{S_f} in the case of a put option (left plot) and call option (right plot).

For numerical purposes of calculating the value of the option, the curve S_f should be the missing boundary. However, the location of this curve is unknown and "free" which explains why the problem is called free boundary problem.

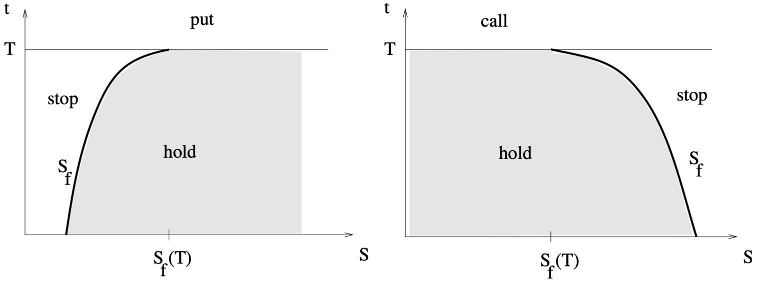


Figure: Continuation (shaded region) and stopping regions and plot of the curve Γ_{S_f} in the case of a put option (left plot) and call option (right plot).

In finance context, the location of Γ_{S_f} is very important, since it defines the early exercise curve for which exercising the option is optimal. The right plot corresponds to the case of a call option.

If we have a European put option, one of the “boundary conditions” if defined by $V_P(S,t) \to 0$, as $S \to +\infty$.

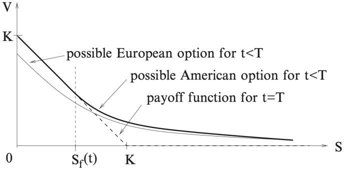
The other boundary condition is formulated for $S = 0$.

For American put options, the latter shall be imposed at the curve Γ_{S_f}.

However, since the location of this curve is unknown we need to impose an extra condition to define its location.

We note that for fixed t and $S < S_f(t)$, we have $V_P^{Am}(S,t) = K - S$ and thus, $\frac{\partial V_P^{Am}}{\partial S}(S,t) = -1$, the slope of the straight line $K - S$.

Actually, it can be proven that the derivative $\frac{\partial V_P^{Am}}{\partial S}(S_f(t), t)$ exists and that $\frac{\partial V_P^{Am}}{\partial S}(S_f(t), t) = -1$, that is, the graph of V_P^{Am} touches the graph of the payoff function tangentially.



Thus, at each of the points $S_f(t)$ in Γ_{S_f} two conditions must hold:

$$\begin{cases} V_P^{Am}(S_f(t), t) & = K - S_f(t) \\ \frac{\partial V_P^{Am}}{\partial S}(S_f(t), t) & = -1. \end{cases} \tag{51}$$

The case of an American call option is similar.

The boundary condition at $S = 0$ is similar to the case of an European call option and now the free boundary conditions to be imposed at Γ_{S_f} are

$$\begin{cases} V_C^{Am}(S_f(t), t) & = S_f(t) - K \\ \frac{\partial V_C^{Am}}{\partial S}(S_f(t), t) & = 1. \end{cases} \tag{52}$$

Before addressing a numerical algorithm for the calculation of the value of an American option, we will introduce a simpler problem - the obstacle problem - and we will see that the numerical solution of the two problems is similar.

The obstacle problem

We assume that we have an obstacle defined by the graph of a function g that satisfies some conditions:

- $g \in C^2(]-1,1[) \cap C^0([-1,1])$
- $g(x) > 0, \ \forall x \in]\alpha, \beta[$, for some $-1 < \alpha < \beta < 1$
- $g''(x) < 0, \ \forall x \in]\alpha, \beta[$
- $g(-1) < 0$ and $g(1) < 0$

and a string defined by a function u with minimal length passes over the obstacle and is held fixed at two points, say 1 and -1.

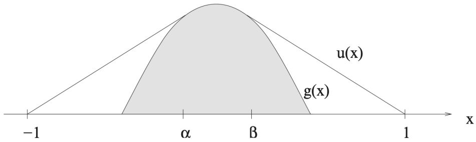


Figure: Plots of u and g in the obstacle problem.

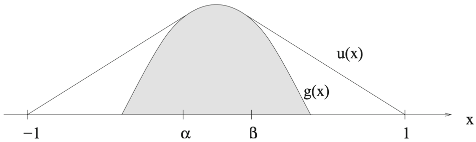


Figure: Plots of u and g in the obstacle problem.

For $x \in]\alpha, \beta[$, the string is in contact with the obstacle, but the points α and β are unknown.

We just know that the string is either in contact with the obstacle or that it must be straight because the string has minimal length.

We will formulate the obstacle problem as:

Obstacle problem - Given a function g (defining the obstacle) satisfying the conditions above, find a function u such that

$$\begin{array}{lll} \text{for } -1 < x < \alpha, & u'' = 0 & (u > g) \\ \text{for } \alpha < x < \beta, & u = g & (\text{which implies that } u'' = g'' < 0) \\ \text{for } \beta < x < 1, & u'' = 0 & (u > g) \end{array} \tag{53}$$

We note that the previous conditions can be written as

$$\begin{array}{lll} \text{if } u > g, & \text{then} & u'' = 0 \\ \text{if } u = g, & \text{then} & u'' < 0 \end{array} \tag{54}$$

which implies that in both cases we have $u''(u - g) = 0$.

At the points $x = \alpha$ and $x = \beta$, this equality shall be understood in the sense of the limits,

$$\begin{cases} \lim_{x \to \alpha^-} u''(x)(u(x) - g(x)) = 0 \\ \lim_{x \to \alpha^+} u''(x)(u(x) - g(x)) = 0 \end{cases}$$

and

$$\begin{cases} \lim_{x \to \beta^-} u''(x)(u(x) - g(x)) = 0 \\ \lim_{x \to \beta^+} u''(x)(u(x) - g(x)) = 0 \end{cases}$$

The obstacle problem can be reformulated as
Obstacle problem - complementary version - Given a function g (the obstacle) satisfying the conditions above, find a function $u \in C^1(]-1,1[) \cap C^0([-1,1])$ and that is twice differentiable almost everywhere in $]-1,1[$ such that

- $u''(u-g) = 0$
- $-u'' \geq 0$
- $u - g \geq 0$
- $u(-1) = u(1) = 0.$

The advantage of this complementary version of the obstacle problem when compared with the first version is that it does not mention explicitly the points α and β that are unknown.

Assuming that we are able to solve the complementary version, the points α and β can be determined easily.

Remark

Actually, it can be proven that the solution of the obstacle problem, u, satisfies the conditions

$$u'(\alpha) = g'(\alpha) \quad \text{and} \quad u'(\beta) = g'(\beta).$$

Next, we present a numerical method for solving the complementary version of the obstacle problem.

We define a grid of points $\mathcal{G} = \{x_i,\ i = 0, ..., N_x\}$, where $x_i = -1 + ih_x,\ i = 0, 1, ..., N_x$, where $h_x = \frac{2}{N_x}$ and will denote by w_i the approximation of $u(x_i)$ and we will use the notation $g_i = g(x_i)$. We define also $\mathcal{W} = \{w_i,\ i = 0, ..., N_x\}$.

Considering the second order finite difference formula to approximate the second derivative of u we get the discrete version of the obstacle problem

Discrete version of the obstacle problem - complementary version - Given a grid of points $\mathcal{G}$, determine $\mathcal{W}$ such that

- $\left(-w_{i+1} + 2w_i - w_{i-1}\right)\left(w_i - g_i\right) = 0,\ i = 1, 2, .., N_x - 1$ (it is convenient to take this "sign" in order to have the matrix A defined below being positive definite)
- $-w_{i+1} + 2w_i - w_{i-1} \geq 0,\ i = 1, 2, .., N_x - 1$
- $w_i \geq g_i,\ i = 1, 2, .., N_x - 1$
- $w_0 = w_{N_x} = 0.$

For the solution of this problem we define $w_0 = w_{N_x} = 0$ and $w_i,\ i = 1, ..., N_x - 1$ are obtained by solving

$$
\begin{cases}
(W-G)^T AW = 0 \\
AW \geq 0 \\
W \geq G,
\end{cases}
\tag{55}
$$

where

$$
A = \begin{bmatrix}
2 & -1 & 0 & \dots & \dots & \dots & 0 \\
-1 & 2 & -1 & 0 & \dots & \dots & 0 \\
0 & -1 & 2 & -1 & 0 & \dots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \dots & 0 & -1 & 2 & -1 & 0 \\
0 & \dots & \dots & 0 & -1 & 2 & -1 \\
0 & \dots & \dots & \dots & 0 & -1 & 2
\end{bmatrix}, \;
W = \begin{bmatrix}
w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{N_x-3} \\ w_{N_x-2} \\ w_{N_x-1}
\end{bmatrix}, \;
G = \begin{bmatrix}
g_1 \\ g_2 \\ g_3 \\ \vdots \\ g_{N_x-3} \\ g_{N_x-2} \\ g_{N_x-1}
\end{bmatrix}.
$$

Defining

$$X = W - G, \quad Y = AW$$

the problem (55) can be written as

$$\begin{cases} AX = Y - AG \\ X^T Y = 0 \\ Y \geq 0 \\ X \geq 0. \end{cases} \tag{56}$$

This problem can be solved by the Projected Successive Over-relaxation (PSOR) Method which is a variant of the Successive Over-relaxation (SOR) Method .

The Successive Over-relaxation (SOR) and Projected Successive Over-relaxation (PSOR) Methods

Before introducing the PSOR Method we recall the SOR method as an iterative method for solving a (general) linear system

$$Mx = b, \tag{57}$$

where M is a general $N \times N$ invertible matrix and b is a $N \times 1$ vector.

The SOR method provides an iterative fixed point method for building a sequence of vectors $\left(x^{(n)}\right)_{n=1,2,\ldots}$ that under certain conditions will converge to x, the solution of the linear system (57).

Jacobi method

Consider the linear system:

$$m_{1,1}x_1 + m_{1,2}x_2 + m_{1,3}x_3 + \cdots + m_{1,N}x_n = b_1$$

$$m_{2,1}x_1 + m_{2,2}x_2 + m_{2,3}x_3 + \cdots + m_{2,N}x_n = b_2$$

$$\ldots$$

$$m_{N,1}x_1 + m_{N,2}x_2 + \cdots + m_{N,N-1}x_{N-1} + m_{N,N}x_N = b_N$$

Assume that $m_{i,i} \neq 0, i = 1, \ldots, N.$

We can write the previous linear system as

$$x_1 = \frac{b_1 - m_{1,2}x_2 - m_{1,3}x_3 - \cdots - m_{1,N}x_N}{m_{1,1}}$$

$$x_2 = \frac{b_2 - m_{2,1}x_1 - m_{2,3}x_3 - \cdots - m_{2,N}x_N}{m_{2,2}}$$

$$\ldots$$

$$x_N = \frac{b_N - m_{N,1}x_1 - m_{N,2}x_2 - \cdots - m_{N,N-1}x_{N-1}}{m_{N,N}}$$

Jacobi method

The previous system can be written as $x = G(x)$:

$$\begin{aligned} x_1 &= \frac{b_1 - m_{1,2}x_2 - m_{1,3}x_3 - \cdots - m_{1,N}x_N}{m_{1,1}} \\ x_2 &= \frac{b_2 - m_{2,1}x_1 - m_{2,3}x_3 - \cdots - m_{2,N}x_N}{m_{2,2}} \\ &\cdots \\ x_N &= \frac{b_N - m_{N,1}x_1 - m_{N,2}x_2 - \cdots - m_{N,N-1}x_{N-1}}{m_{N,N}} \end{aligned} .$$

and we will apply a fixed point method of type: $x^{(n+1)} = G(x^{(n)})$

$$\begin{aligned} x_1^{(n+1)} &= \frac{b_1 - m_{1,2}x_2^{(n)} - m_{1,3}x_3^{(n)} - \cdots - m_{1,N}x_N^{(n)}}{m_{1,1}} \\ x_2^{(n+1)} &= \frac{b_2 - m_{2,1}x_1^{(n)} - m_{2,3}x_3^{(n)} - \cdots - m_{2,N}x_N^{(n)}}{m_{2,2}} \\ &\cdots \\ x_N^{(n+1)} &= \frac{b_N - m_{N,1}x_1^{(n)} - m_{N,2}x_2^{(n)} - \cdots - m_{N,N-1}x_{N-1}^{(n)}}{m_{N,N}} \end{aligned} , \qquad n = 0, 1, 2, \ldots$$

Jacobi method

$$
\left.
\begin{aligned}
x_1^{(n+1)} &= \frac{b_1 - m_{1,2}x_2^{(n)} - m_{1,3}x_3^{(n)} - \cdots - m_{1,N}x_N^{(n)}}{m_{1,1}} \\
x_2^{(n+1)} &= \frac{b_2 - m_{2,1}x_1^{(n)} - m_{2,3}x_3^{(n)} - \cdots - m_{2,N}x_N^{(n)}}{m_{2,2}} \\
&\dots \\
x_N^{(n+1)} &= \frac{b_N - m_{N,1}x_1^{(n)} - m_{N,2}x_2^{(n)} - \cdots - m_{N,N-1}x_{N-1}^{(n)}}{m_{N,N}}
\end{aligned}
\right. , \qquad n = 0, 1, 2, \dots
$$

Compact form:

$$
x_i^{(n+1)} = \frac{b_i}{m_{i,i}} - \frac{\sum_{j=1,\dots,n, j \neq i}^{N} m_{i,j} x_j^{(n)}}{m_{i,i}},
$$

$i = 1, \dots, N; \;\; n = 0, 1, 2, \dots$

Jacobi method

Consider the linear system $M\,x = b$:

$$M = \begin{bmatrix} 2 & 1 & 0 \\ -1 & 2 & 1 \\ 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}.$$

- Calculate one iteration of Jacobi method, taking the initial vector $x^{(0)} = (0.5, 0.8, 1)$.

  Solution:

$$x_1^{(1)} = \frac{b_1 - m_{1,2}x_2^{(0)} - m_{1,3}x_3^{(0)}}{m_{1,1}} = \frac{1}{2}\,(2 - 0.8 - 0) = 0.6;$$

$$x_2^{(1)} = \frac{b_2 - m_{2,1}x_1^{(0)} - m_{2,3}x_3^{(0)}}{m_{2,2}} = \frac{1}{2}(2 + 0.5 - 1) = 0.75;$$

$$x_3^{(1)} = \frac{b_3 - m_{3,1}x_1^{(0)} - m_{3,2}x_2^{(0)}}{m_{3,3}} = \frac{1}{2}(1 - 0 + 0.8) = 0.9;$$

Gauss-Seidel method

Jacobi method

$$
\left.\begin{aligned}
x_1^{(n+1)} &= \frac{b_1 - m_{1,2}x_2^{(n)} - m_{1,3}x_3^{(n)} - \cdots - m_{1,N}x_N^{(n)}}{m_{1,1}} \\
x_2^{(n+1)} &= \frac{b_2 - m_{2,1}x_1^{(n)} - m_{2,3}x_3^{(n)} - \cdots - m_{2,N}x_N^{(n)}}{m_{2,2}} \\
&\dots \\
x_N^{(n+1)} &= \frac{b_N - m_{N,1}x_1^{(n)} - m_{N,2}x_2^{(n)} - \cdots - m_{N,N-1}x_{N-1}^{(n)}}{m_{N,N}}
\end{aligned}\right. \quad , \qquad n = 0, 1, 2, \dots
$$

In the calculation of $x_2^{(n+1)}$ we could use the value $x_1^{(n+1)}$, instead of $x_1^{(n)}$ because it was already calculated. The same procedure can be applied in the calculation of the other components, using the components $x_i^{(n+1)}$, instead of $x_i^{(n)}$.

Gauss-Seidel method

This leads to a different numerical method know as Gauss-Seidel method:

$$
\left.
\begin{aligned}
x_1^{(n+1)} &= \frac{b_1 - m_{1,2}x_2^{(n)} - m_{1,3}x_3^{(n)} - \cdots - m_{1,N}x_N^{(n)}}{m_{1,1}} \\
x_2^{(n+1)} &= \frac{b_2 - m_{2,1}x_1^{(n+1)} - m_{2,3}x_3^{(n)} - \cdots - m_{2,N}x_N^{(n)}}{m_{2,2}} \\
&\cdots \\
x_N^{(n+1)} &= \frac{b_N - m_{N,1}x_1^{(n+1)} - m_{N,2}x_2^{(n+1)} - \cdots - m_{N,N-1}x_{N-1}^{(k+1)}}{m_{N,N}}
\end{aligned}
\right. , \qquad n = 0, 1, 2, \ldots
$$

Gauss-Seidel method

Compact form:

$$x_i^{(n+1)} = \frac{b_i}{m_{i,i}} - \frac{\sum_{j=1}^{i-1} m_{i,j} x_j^{(n+1)}}{m_{i,i}} - \frac{\sum_{j=i+1}^{N} m_{i,j} x_j^{(n)}}{m_{i,i}},$$

$i = 1, \ldots, N,\ n = 0, 1, 2, \ldots$

Example - Gauss-Seidel method

Consider the linear system $M\,x = b$:

$$M = \begin{bmatrix} 2 & 1 & 0 \\ -1 & 2 & 1 \\ 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}.$$

- Calculate one iteration of Gauss-Seidel method taking $x^{(0)} = (0.5, 0.8, 1)$.

Solution:

$$x_1^{(1)} = \frac{b_1 - m_{1,2}x_2^{(0)} - m_{1,3}x_3^{(0)}}{m_{1,1}} = \frac{1}{2}\,(2 - 0.8 - 0) = 0.6;$$

$$x_2^{(1)} = \frac{b_2 - m_{2,1}x_1^{(1)} - m_{2,3}x_3^{(0)}}{m_{2,2}} = \frac{1}{2}(2 + 0.6 - 1) = 0.8;$$

$$x_3^{(1)} = \frac{b_3 - m_{3,1}x_1^{(1)} - m_{3,2}x_2^{(1)}}{m_{3,3}} = \frac{1}{2}(1 - 0 + 0.8) = 0.9;$$

Compact form of Gauss-Seidel method:

$$x_i^{(n+1)} = \frac{b_i}{m_{i,i}} - \frac{\sum_{j=1}^{i-1} m_{i,j} x_j^{(n+1)}}{m_{i,i}} - \frac{\sum_{j=i+1}^{N} m_{i,j} x_j^{(n)}}{m_{i,i}}, \qquad (58)$$

$i = 1, \ldots, N, \; n = 0, 1, 2, \ldots$

The sequence of vectors $\left(x^{(n)}\right)_{n=1,2,\ldots}$ of SOR method is defined by

$$\begin{cases} \text{given } x^{(0)}, \text{ iterate } n = 0, 1, \ldots \\ x_i^{(n+1)} = x_i^{(n)} + \frac{\omega}{m_{i,i}} \left(b_i - \sum_{j=1}^{i-1} m_{i,j} x_j^{(n+1)} - \sum_{j=i}^{N} m_{i,j} x_j^{(n)} \right), i = 1, \ldots, N \end{cases}$$

for some $\omega \in]0, 2[$. Note that for $\omega = 1$ we obtain Gauss-Seidel method.

The solution of problem (56) is similar, but the inequality $X \geq 0$ is imposed component-wise. This is the main idea of the following iterative method known as PSOR method,

$$\begin{cases} \text{given } x_i^{(0)} \geq 0, \text{ iterate } n = 0, 1, \ldots \\ x_i^{(n+1)} = \max\left\{0, x_i^{(n)} + \frac{\omega}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(n+1)} - \sum_{j=i}^{N} a_{i,j} x_j^{(n)} \right) \right\}, \\ i = 1, \ldots, N \end{cases}$$

Example

Next, we present an example of an obstacle problem for which we are able to determine the solution and then, apply the PSOR method to solve this obstacle problem.

Let $g(x) = \frac{15}{16} + \frac{3}{8}x - \frac{25}{16}x^2$. It is straightforward to prove that the solution for the obstacle problem is

$$u(x) = \begin{cases} x+1, & -1 \leq x < \alpha = -\frac{1}{5} \\ g(x), & \alpha \leq x \leq \beta = \frac{3}{5} \\ -\frac{3}{2}\left(x - \frac{3}{5}\right) + \frac{3}{5}, & \beta < x \leq 1. \end{cases}$$

Indeed, for $x < \alpha$, we have $u''(x) = 0$, and the same happens for $x > \beta$. For $\alpha < x < \beta$, we have $u(x) = g(x)$.

Moreover, taking into account Remark 1.1, we verify that

$$g'(x) = \frac{3}{8} - \frac{25}{8}x.$$

Therefore,

$$g'(\alpha) = g'\left(-\frac{1}{5}\right) = \frac{3}{8} + \frac{5}{8} = 1 = u'(\alpha)$$

and

$$g'(\beta) = g'\left(\frac{3}{5}\right) = \frac{3}{8} - \frac{25}{8} \times \frac{3}{5} = \frac{12}{8} = -\frac{3}{2} = u'(\beta).$$

Now we apply the PSOR method to solve the obstacle problem of the previous Example, where we defined $b = -AG$. Next Figure shows the plots of g and u, the solution of the obstacle problem of the previous example, obtained with N=200.

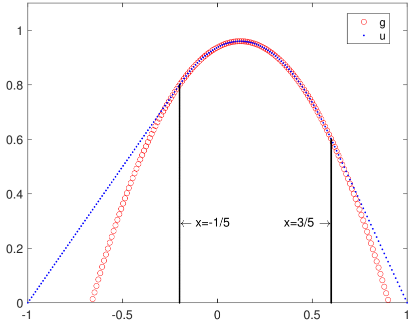


Figure: Plots of g and u, the solution of the obstacle problem, obtained with the PSOR with N=200.

Black-Scholes inequality for an American option

In this section we will introduce a complementary problem related with the calculation of an American option value. Define the Black-Scholes operator as

$$\mathcal{L}_{BS}(V) := \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV$$

for which we can write the Black-Scholes equation as

$$\frac{\partial V}{\partial t} + \mathcal{L}_{BS}(V) = 0. \tag{59}$$

It can be proven that in the stopping region, the value of an American option satisfies

$$\frac{\partial V}{\partial t} + \mathcal{L}_{BS}(V) < 0. \tag{60}$$

Therefore, in a similar way as we defined for the obstacle problem, for an American option we have

$$\begin{array}{lll} \text{if } V > \text{payoff}, & \text{then} & \frac{\partial V}{\partial t} + \mathcal{L}_{BS}(V) = 0 \\ \text{if } V = \text{payoff}, & \text{then} & \frac{\partial V}{\partial t} + \mathcal{L}_{BS}(V) < 0 \end{array} \tag{61}$$

which suggests to formulate the linear complementary problem, where $\phi(S)$ denotes the payoff function and we consider the change of variables $\tau = T - t$,

- $\left(\frac{\partial V}{\partial \tau}(S,\tau) + \mathcal{L}_{BS}(V(S,\tau))\right)(V(S,\tau) - \phi(S)) = 0$
- $\frac{\partial V}{\partial t}(S,\tau) - \mathcal{L}_{BS}(V(S,\tau)) \geq 0$
- $V(S,\tau) \geq \phi(S)$
- $V(S,0) = \phi(S)$
- $V(S_{min},\tau) = \phi(S_{min})$
- $V(S_{max},\tau) = \phi(S_{max})$.

For instance, defining

$$U_j = [U_{1,j}, U_{2,j}, ..., U_{N_S-1,j}]^T ,$$

using Crank-Nicolson method, for any $j = 0, 1, ..., N_t - 1$ given U_j we can calculate U_{j+1}, by solving a linear system which can be written as $AU_{j+1} = b$. Therefore, defining

$$G = [\phi(S_1), \phi(S_2), ..., \phi(S_{N_S-1})]^T ,$$

for each $j = 0, 1, ..., N_t - 1$, setting $W = U_{j+1}$ we can write the complementary problem as (compare with (55) for the obstacle problem)

$$\begin{cases} (W - G)^T (AW - b) = 0 \\ AW - b \geq 0 \\ W \geq G. \end{cases} \tag{62}$$

Defining

$$X = W - G, \quad Y = AW - b,$$

we have

$$AX = AW - AG = Y + \underbrace{b - AG}_{=\tilde{b}}$$

and defining $\tilde{b} = b - AG$ the problem (62) can be written as (compare with (56))

$$\begin{cases} AX = Y - AG \\ X^T Y = 0 \\ Y \geq 0 \\ X \geq 0. \end{cases} \tag{63}$$

that can be solved by PSOR Method.

$$
\begin{cases}
\text{given } X_i^{(0)} \geq 0, \text{ iterate } n = 0, 1, ... \\
X_i^{(n+1)} = \max\left\{0, X_i^{(n)} + \frac{\omega}{a_{i,i}}\left(\tilde{b}_i - \sum_{j=1}^{i-1} a_{i,j} X_j^{(n+1)} - \sum_{j=i}^{N} a_{i,j} X_j^{(n)}\right)\right\}, \\
i = 1, 2, ..., N_S - 1.
\end{cases}
$$

Next Figure shows the plot of $V(S,t)$ for $t = 0, 0.5, 1$ in the case of an American put option, obtained with the parameters $r = 0.06$, $\sigma = 0.3$, $T = 1$ and $K = 10$, taking $S^* = 15$.

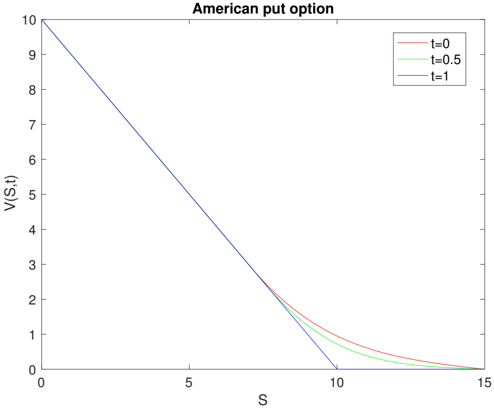


Figure: Plot of $V(S,t)$ for $t = 0, 0.5, 1$ in the case of an American put option, obtained with the parameters $r = 0.06$, $\sigma = 0.3$, $T = 1$ and $K = 10$, taking $S^* = 15$.

Next Figure shows the plot of $V(S,t)$ in the continuation region and the early exercise curve.

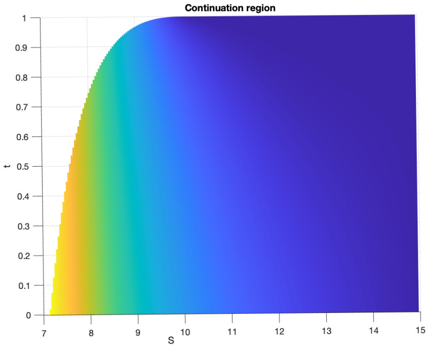


Figure: Continuation region of an American put option, obtained with the parameters $r = 0.06$, $\sigma = 0.3$, $T = 1$ and $K = 10$, taking $S^* = 15$.