This document shows how to solve an American Call and Put with finite differences.

Contents

1 The Model

Let time $t \in [0, T]$. In this interval, the stock price follows

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) \qquad\qquad \text{given } S(0) = S_0.$$

Let the riskless asset follow

$$dB(t) = rB(t)dt.$$

I will price the option using the risk-neutral measure $\mathbb{Q}$ (equivalent martingale measure).

To derive the measure $\mathbb{Q}$, use the fact that under $\mathbb{Q}$ the discounted price process of $S(t)$ is a martingale, i.e. in expectation the price change is in accordance with the riskless asset. Thus,

$$\mathbb{E}_t^{\mathbb{Q}}[dS(t)] = rS(t)dt \qquad \Leftrightarrow \qquad \mathbb{E}_t^{\mathbb{Q}}[\mu S(t)dt + \sigma S(t)dW(t)] = rS(t)dt.$$

Therefore, under the risk-neutral measure $\mathbb{Q}$ it must hold that $\mu = r$. Using Girsanov's Theorem, set $dW(t) = dW^{\mathbb{Q}}(t) - \frac{\mu - r}{\sigma}dt$ where $W^{\mathbb{Q}}(t)$ is a standard Brownian Motion under the $\mathbb{Q}$-measure. Thus, the discounted price process of

$$dS(t) = rS(t)dt + \sigma S(t)dW^{\mathbb{Q}}(t)$$

is a martingale under $\mathbb{Q}$, i.e. $d\mathbb{E}_t^{\mathbb{Q}}[e^{-rt}S(t)] = 0$.

Let $V(t, S(t))$ denote the Value of the Option. By risk-neutral pricing (omitting the arguments in line 2 and 3)

$$d\mathbb{E}_t^{\mathbb{Q}}[e^{-rt}V(t, S(t))] = 0,$$
$$-rV + \partial_t V + rS\partial_S V + 0.5\sigma^2 S^2 \partial_{S^2}^2 V = 0,$$
$$-\partial_t V = F(V),$$

where $F(V) \equiv -rV + r\partial_S V + 0.5\sigma^2 \partial_{S^2}^2 V$.

Since we are dealing with an American Option, we can summarise the following for the value of the option

$$-\partial_t V = -rV + rS\partial_S V + 0.5\sigma^2 S^2 \partial_{S^2}^2 V \equiv F(V) \qquad \forall t \in [0, T] \qquad (1)$$
$$V(T, S(T)) = max\{\text{Payoff}(S(T), K), 0\} \qquad\qquad\qquad\qquad (2)$$
$$V \geq \text{Payoff}(S(t), K) \qquad\qquad \forall t \in [0, T] \qquad (3)$$

Conditions (1) & (2) hold for European options while American options also require condition (3) which states that if the continuation value is less than the exercise value, then the option will be exercised prematurely.

In the usual PDE literature the unknown function is solved forwards in time, but here the function is solved backwards in time using terminal condition (2). If we substitute $t = T - \tau$, then we would obtain $\partial_t V(\tau, \cdot) = F(V)$ where τ runs from 0 to terminal time T and we have the exact same problem as in the usual PDE literature.

2 Numerical Schemes

The idea is to solve PDE (1) backwards in time with finite differences using terminal condition (2) to compute the continuation value. If the continuation value is less than the exercise value, we simply replace the continuation value by the exercise value. It is possible by using the transformation $V(t, log(S))$ to get rid of the non-constant coefficients in front of the derivatives, but this is a problem-specific artefact that I don't want to use for the sake of generality.

2.1 Baseline Calibration

We first require a baseline calibration. I pick the following values

- $\mu = 0.06$, $r = 0.02$, $\sigma = 0.2$, $T = 1$, $S_0 = 10$, $K = S_0$.

The time horizon is 1 year such that after 1 year a riskless return of 2% and an expected risky return of 6% is realised. I further pick the option to be at the money which as will be seen helps to calibration artificial boundary conditions.

2.2 Grids

To use finite differences, we firstly require grids of the two state variables. The time grid is given by the evenly spaced grid

$$t_{grid} = \{0, \Delta_t, 2\Delta_t, \ldots, 1\}, \qquad \Delta_t = 0.02, \qquad |t_{grid}| = N_t.$$

It is important that the time step is small as otherwise the approximation of PDE (1) will be bad.

Unlike the time grid, the stock price grid does not have a finite domain as in theory the stock price at any time point can be in $(0, \infty)$. However, large values are decreasingly likely so that I simulate the stock price under the $\mathbb{P}$-measure to get an idea of what the stock price grid ought to be. Figure 1 shows the simulation of 500 paths. Naturally, the dispersion of the price gets larger over time, so that in theory time dependent grids should be taken. In other words, at each time step the ergodic set of the stock price, i.e. the set where most probability mass of the stock price distribution lies, is different and gets larger over time. This implies that closer to $t = 0$ we can use a domain of smaller length as opposed to time periods closer to $t = T$.

Even though time-dependent grids do not pose an issue for finite differences as long as functions can be appropriately inter- and extrapolated, for simplicity I take at $t = T$ the 99% percentile as the maximum grid value and the 1% percentile as the minimum value of the simulated stock price distribution and keep the grid constant over time.

Thus, I set

$$S_{grid} = \{S_{min}, S_{min} + \Delta_S, S_{min} + 2\Delta_S, \ldots, S_{max}\} \qquad |S_{grid}| = N_s = 200.$$

Given Figure 1, clearly more probability mass lies towards the center than at the edges so a non-uniformly space grid with more gridpoints in the center would be appropriate. However, non-uniformly spaced grids are a nuisance with finite differences which is why I choose a uniformly spaced grid.

Figure 1

Simulated Price Paths
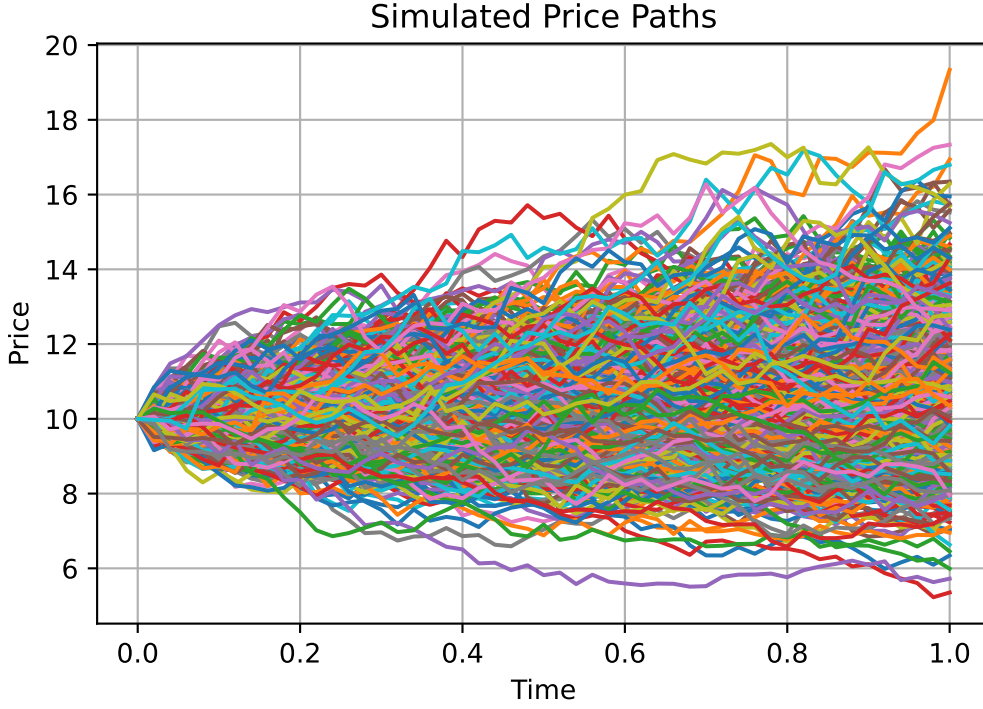
2.3 Implicit Scheme

Let's start with the implicit scheme. For the time derivative I use

$$\partial_t V = \frac{V(t, \cdot) - V(t - \Delta, \cdot)}{\Delta},$$

although it would be equivalent to use $\partial_t V = \frac{V(t-\Delta, \cdot) - V(t, \cdot)}{-\Delta}$.

It is common to denote time with a superscript as superscripts typically denote the iteration we are solving for. Moreover, let's index V by its remaining argument so that

$$V_i^n \equiv V(T - n\Delta_t, S_i) \qquad \forall S_i \in S_{grid}.$$

Discretising PDE (1) with central differences with the implicit scheme yields[1]

$$\frac{V^{n+1} - V^n}{\Delta_t} = F^{n+1}(V)$$

$$\frac{V_i^{n+1} - V_i^n}{\Delta_t} = -rV_i^{n+1} + rS_i\left(\frac{V_{i+1}^{n+1} - V_{i-1}^{n+1}}{2\Delta_S}\right) + \frac{1}{2}\sigma^2 S_i^2\left(\frac{V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}}{(\Delta_S)^2}\right) = 0$$

$$\forall i \in \{2, \ldots, N_S - 1\}, \ \forall n \in \{0, \ldots, N_t - 1\}.$$

At $i = 1$, i.e. $S_1 = S_{min}$, and $i = I = N_S$ the issue is that central difference cannot be computed anymore since gridpoint $i = 0$ and gridpoint $i = N_S + 1$ no longer exist. However, we don't need to compute V_1^{n+1} or V_I^{n+1} since we can approximate the boundary value $V(t, S_{min})$ and $V(t, S_{max}) \ \forall t \in [0, T]$ by considering

$$V(t, S_{max}) \approx max\{\text{Payoff}(S_{max}, K), 0\}, \qquad V(t, S_{min}) \approx max\{\text{Payoff}(S_{min}, K), 0\} \qquad \forall t \in t_{grid}.$$

---

[1]It is also directly possible to use a time-varying riskless rate and volatility as long as they are not stochastic (the stochastic case is a different model and the PDE would also not be the same). We can incorporate time-varying coefficients by using r^{n+1} and σ^{n+1}.

For example, if we're dealing with a call option, then at S_{max} the option is so deep in the money that the option will be exercised because the stock price will very likely only go down from that point onwards. At S_{min} the option will be so deep out of the money that it is likely worthless as it will very likely never surpass the strike price again..

This is visualised in Figure 2. Columns denote the time gridpoint and rows denote the stock price gridpoint. The red gridpoints are the terminal condition while the blue dots are the boundary conditions. Thus, we only need to solve the function V at the interior black gridpoints.

Figure 2: Lattice with Boundary & Terminal Condition.
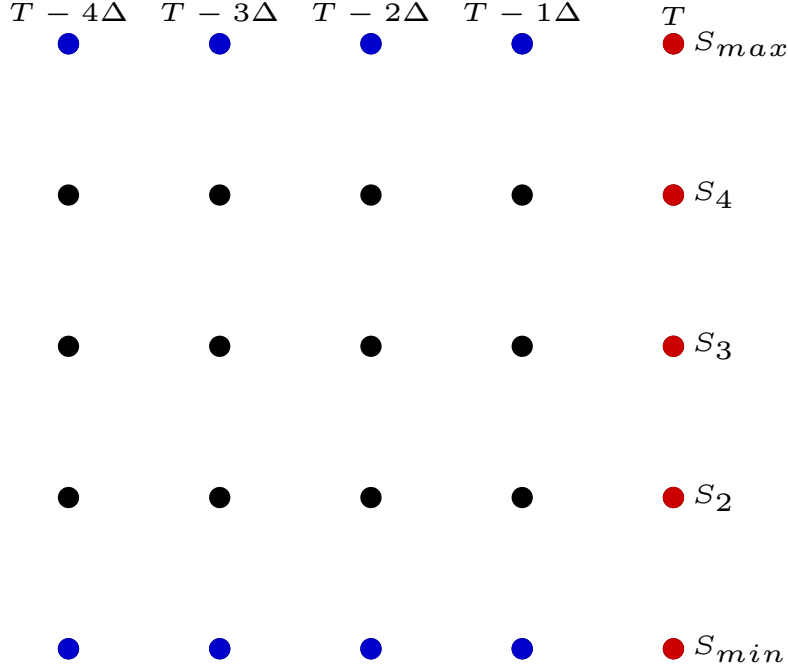
Before progressing, we can re-arrange the system of $N_S - 2$ equations to obtain a tridiagonal system representation. Firstly, multiply by the time difference and collect terms on the right hand side

$$V_i^{n+1} - V_i^n = \left(\frac{\sigma^2 S_i^2 \Delta_t}{2(\Delta_S)^2} - \frac{rS_i\Delta_t}{2\Delta_S}\right) V_{i-1}^{n+1} + \left(-r\Delta_t - \frac{\sigma^2 S_i^2 \Delta_t}{(\Delta_S)^2}\right) V_i^{n+1} + \left(\frac{\sigma^2 S_i^2 \Delta_t}{2(\Delta_S)^2} + \frac{rS_i\Delta_t}{2\Delta_S}\right) V_{i+1}^{n+1}.$$

Then, isolating V_i^n on the right hand side and factoring yields

$$\left(-\frac{\sigma^2 S_i^2 \Delta_t}{2(\Delta_S)^2} + \frac{rS_i\Delta_t}{2\Delta_S}\right) V_{i-1}^{n+1} + \left(1 + r\Delta_t + \frac{\sigma^2 S_i^2 \Delta_t}{(\Delta_S)^2}\right) V_i^{n+1} + \left(-\frac{\sigma^2 S_i^2 \Delta_t}{2(\Delta_S)^2} - \frac{rS_i\Delta_t}{2\Delta_S}\right) V_{i+1}^{n+1} = V_i^n.$$

From here we can define

$$\alpha_i V_{i-1}^{n+1} + \beta_i V_i^{n+1} + \gamma_i V_{i+1}^{n+1} = V_i^n \qquad \forall n \in \{0, N_t - 1\}, \ \forall i \in \{2, \ldots, N_S - 1\}, \qquad (4)$$

which is a tridiagonal system at every n where V_1^{n+1}, V_I^{n+1} and V_i^n are known.

The algorithm works as follows

1. Compute $V^0(S_i)$ from terminal condition (2).

2. Solve the system of linear equations (4). For tridiagonal systems the efficient Thomas algorithm exists which is essentially a form of Gaussian elimination. However, for simplicity I will solve the $N_S - 2$ system of equations by numerical root-finding. The solution to the PDE governs the continuation value of the option.

3. Impose the restriction (3) on the solution such that

$$V_i^{n+1} \leftarrow max\{V_i^{n+1}, \text{Payoff}(S_i, K)\} \qquad \forall i \in \{2, \ldots, N_S - 1\}.$$

4. Go back to step 2 and iterate through time until all remaining time periods have been solved for.

2.4   Implicit Scheme no Boundary Conditions

It is also possible to solve PDE (1) without boundary conditions. To tackle the previously encountered issue, three cases are required:

- Interior Grid. On the interior of the grid we can use the same central differences as before and obtain

$$\alpha_i V_{i-1}^{n+1} + \beta_i V_i^{n+1} + \gamma_i V_{i+1}^{n+1} = V_i^n \qquad \forall n \in \{0, N_t - 1\}, \ \forall i \in \{2, \ldots, N_S - 1\}.$$

- Lower bound. At the lower bound $i = 1$ of S_{grid} we use forward differences only and obtain the following equation

$$\frac{V_1^{n+1} - V_1^n}{\Delta_t} + \frac{1}{2}\sigma^2 S_1^2 \left(\frac{V_3^{n+1} - 2V_2^{n+1} + V_1^{n+1}}{(\Delta_S)^2} \right) + rS_1 \left(\frac{V_2^{n+1} - V_1^{n+1}}{\Delta_S} \right) - rV_1^{n+1} = 0.$$

- Upper bound. Analogously, at the upper bound $i = I = N_S$ of S_{grid} we use backward differences only and obtain the following equation

$$\frac{V_I^{n+1} - V_I^n}{\Delta_t} + \frac{1}{2}\sigma^2 S_I^2 \left(\frac{V_I^{n+1} - 2V_{I-1}^{n+1} + V_{I-2}^{n+1}}{(\Delta_S)^2} \right) + rS_I \left(\frac{V_I^{n+1} - V_{I-1}^{n+1}}{\Delta_S} \right) - rV_I^{n+1} = 0.$$

The algorithm works exactly as in the previous Section.

2.5   Explicit Method no Boundary Conditions

The explicit method uses $-\partial_t V = F^n(V)$ and solves for V^{n+1} which only appears on the left-hand side and is immediately given.

Without using any boundary constraints, I once more use central differences in the interior of the grid, forward differences at the lower bound and backward differences at the upper bound. This yields the following

- Interior. Central differences $\forall i \in \{2, \ldots, N_S - 1\}$ yield

$$\frac{V_i^{n+1} - V_i^n}{\Delta_t} + \frac{1}{2}\sigma^2 S_i^2 \left(\frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{(\Delta_S)^2} \right) + rS_i \left(\frac{V_{i+1}^n - V_{i-1}^n}{2\Delta_S} \right) - rV_i^n = 0.$$

Solving for the unknown V_i^{n+1} then yields

$$V_i^{n+1} = V_i^n - \Delta_t \left[\frac{1}{2}\sigma^2 S_i^2 \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{(\Delta_S)^2} + rS_i \frac{V_{i+1}^n - V_{i-1}^n}{2\Delta_S} - rV_i^n \right].$$

- Lower bound. Forward differences at $i = 1$ yield

$$\frac{V_1^{n+1} - V_1^n}{\Delta_t} + \frac{1}{2}\sigma^2 S_1^2 \left(\frac{V_3^n - 2V_2^n + V_1^n}{(\Delta_S)^2} \right) + rS_1 \left(\frac{V_2^n - V_1^n}{\Delta_S} \right) - rV_1^n = 0.$$

Solving for the unknown V_1^{n+1} then yields

$$V_1^{n+1} = V_1^n - \Delta_t \left[\frac{1}{2}\sigma^2 S_1^2 \frac{V_3^n - 2V_2^n + V_1^n}{(\Delta_S)^2} + rS_1 \frac{V_2^n - V_1^n}{\Delta_S} - rV_1^n \right].$$

- Upper bound. Backward differences at $i = I = N_S$ yield

$$\frac{V_I^{n+1} - V_I^n}{\Delta_t} + \frac{1}{2}\sigma^2 S_I^2 \left(\frac{V_I^n - 2V_{I-1}^n + V_{I-2}^n}{(\Delta_S)^2} \right) + rS_I \left(\frac{V_I^n - V_{I-1}^n}{\Delta_S} \right) - rV_I^n = 0.$$

Solving for the unknown V_I^{n+1} then yields

$$V_I^{n+1} = V_I^n - \Delta_t \left[\frac{1}{2}\sigma^2 S_I^2 \frac{V_I^n - 2V_{I-1}^n + V_{I-2}^n}{(\Delta_S)^2} + rS_I \frac{V_I^n - V_{I-1}^n}{\Delta_S} - rV_I^n \right].$$

However, I could not get this method to work even for the European Call Option. This did not change when including the boundary condition. Using very small Δ_t didn't rectify the issue either.

3   Results

Here I show the results. Firstly, the American Call Option has the same price as the European Call Option. Since for the European Call Option we can use the Black-Scholes formula, we can see how well the numerical scheme worked.

Firstly, Figure 3 shows the implicit scheme. At the upper bound of the stock price a deviation is noticeable. Although it is hard to see in the plot, there is also a slight deviation at the lower bound. This is to be expected since the boundary conditions used only hold approximately. However, the numerical approximation is still very good.

The implicit scheme without boundary conditions performed even better as Figure 4 shows.

For the American Put Option, Figure 5 displays the implicit scheme and Figure 6 displays the implicit scheme without the boundary condition. Notice that for small stock prices the value of the option is basically $K - S$ which makes sense as the option is deep in the money.

With or without boundary conditions, the numerical solutions look almost the same. Even though no analytical solution exists, the numerical approximation satisfies the sanity check since the price of an American Option should never be smaller than that of its European counterpart as the American Option grants an additional right which must have a non-negative price under no-arbitrage.
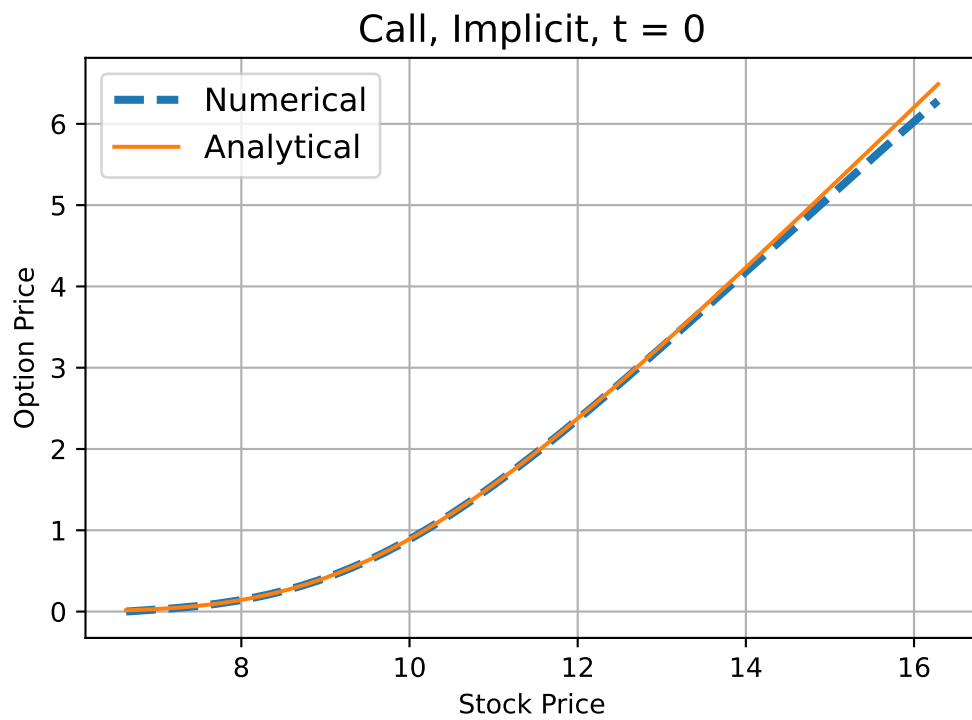
Figure 3

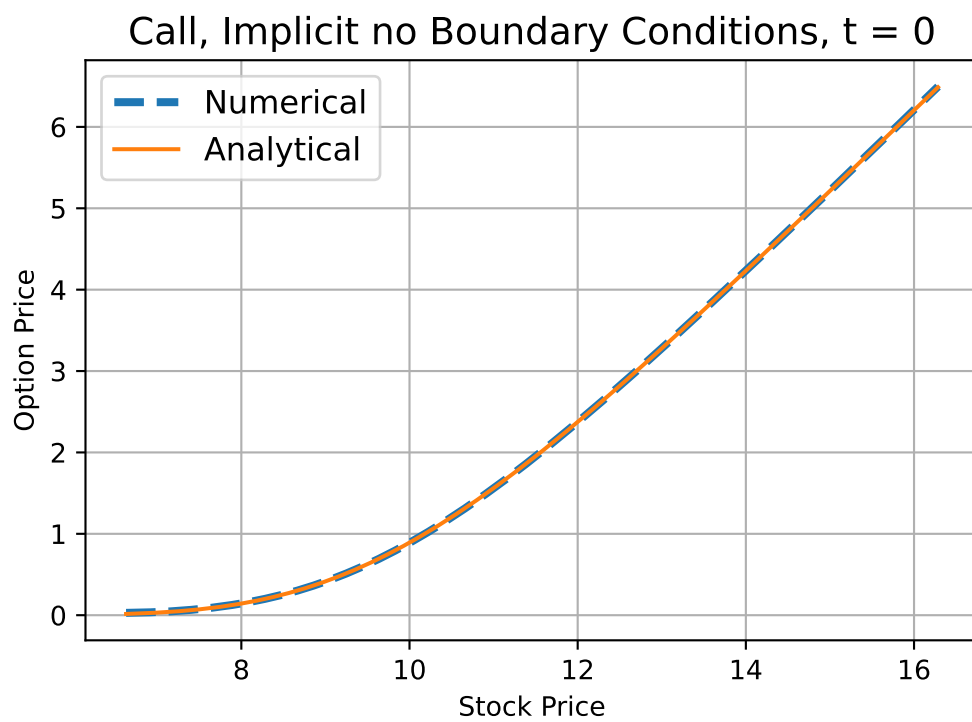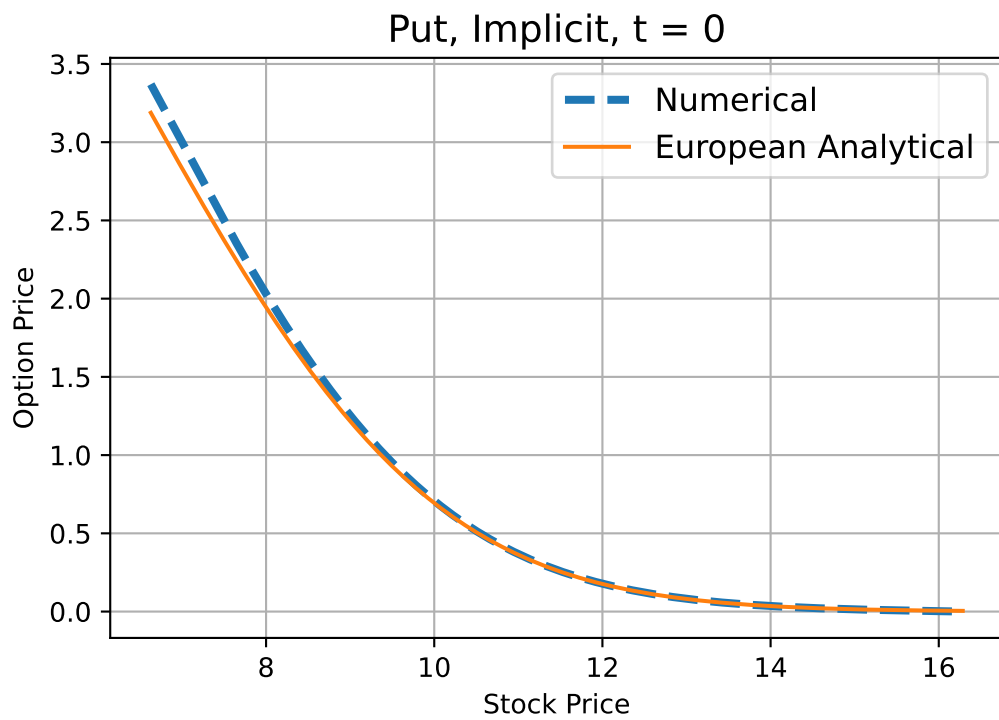Call, Implicit, t = 0

Figure 4

Call, Implicit no Boundary Conditions, t = 0

Figure 5

Figure 6