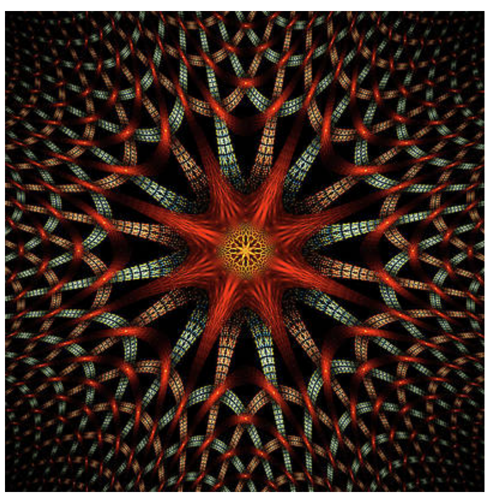
Barrier Options with Time-Dependent Barriers

Raymond Luo

December 12th, 2020

Abstract

In this paper, we find an analytical representation of the value of a barrier option with a time-dependent barrier. We approach this problem with techniques of the same flavor for solving classical barrier options with Black-Scholes PDE. We then analyze our results by considering special cases and showing that they simplify into generic vanilla call options and barrier options. We conclude with numerical analysis of our problem with usage of Monte Carlo techniques.



Contents

1 Introduction

1.1 Set Up and Assumptions

We consider a standard knock-out call (or down-and-out call), with the modification that the lower barrier observed is time-dependent. If at any time before expiration the price falls below some barrier L_t, representing a barrier function $L(t)$ that is dependent on time t, the option becomes worthless. It then follows that we have the following payoff at expiration T:

$$V(S(T),T) = \max\left(S(T) - D), 0\right),\ \text{if } S(t) > L_t \text{ for all } t \in [0,T],\ \text{else } 0$$

For the purpose of our observation, we first set up a series of assumptions. For the purpose of our analytical derivation of our pricing, we have the following:
Assumptions

- $$L_t = E\exp\left(-q(T-t)\right) \tag{1}$$
- $$S(0) > L_0 \tag{2}$$
- At all times t, $0 \leq t \leq T$, we impose the following condition:
  $$L_t = L_0\exp\left(-q(T-t)\right) \leq D\exp\left(-r(T-t)\right) \tag{3}$$
  where r is the interest rate. The condition can be easily satisfied by choosing $E \leq D$ and choosing appropiately the value $q \geq r$

Side Notes:
One interesting choice is $q = r$ and $E = \rho K$, $0 \leq \rho \leq 1$, but lets keep it in a most general form.
Assume that the asset price follows the GBM with constant volatility σ, the interest rate is constant and positive with no dividends.

2 Analytical Derivation

2.1 Usage of PDEs

We denote $V(S(t),t)$ to be the value of this option at time t and stock price $S(t)$. The value of our down-and-out call barrier option $V(S,t)$ for $t<T$ satisfies the Black-Scholes PDE with boundary counditions as follows:

$$\frac{\partial V}{\partial t}+\frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2}+rS\frac{\partial V}{\partial S}-rV=0, \qquad t>0, L(t)<S<\infty \tag{4}$$
$$V(S,T)=\max\{S-D,0\} \tag{5}$$
$$V(L(t),t)=0 \tag{6}$$

We proceed with the following change of variables:
For fixed t,

$$x=\ln\left(S/L(t)\right), u(x,t)=V(S(t),t)$$

We then note that by the chain rule we have:

$$\frac{\partial V}{\partial t}=\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}\cdot\frac{\partial x}{\partial t}=\frac{\partial u}{\partial t}-\frac{\partial u}{\partial x}\cdot\frac{L'(t)}{L(t)} \tag{7}$$
$$\frac{\partial V}{\partial S}=\frac{1}{S}\cdot\frac{\partial u}{\partial x} \tag{8}$$
$$\frac{\partial^2 V}{\partial S^2}=\frac{1}{S^2}\left(\frac{\partial^2 u}{\partial x^2}-\frac{\partial u}{\partial x}\right) \tag{9}$$

Substituting (7)-(9) into our PDE equation (4), we get:

$$\frac{\partial u}{\partial t}+\frac{\sigma^2}{2}\cdot\frac{\partial^2 u}{\partial x^2}+\left(r-\frac{\sigma^2}{2}-\frac{L'(t)}{L(t)}\right)\frac{\partial u}{\partial x}-ru=0, \quad t>0, 0<x<\infty \tag{10}$$
$$u(x,T)=\max\{e^x L(T)-D,0\} \tag{11}$$
$$u(0,t)=0 \tag{12}$$

Here, we consider solutions of the general form $u(x,t)=U(x,t)e^{a(t)x+b(t)}$ with appropiate choices of $a(t),b(t)$. We then have that:

$$\frac{\partial u}{\partial t}=\frac{\partial U}{\partial t}e^{a(t)x+b(t)}+U(x,t)e^{a(t)x+b(t)}(a'(t)x+b'(t)) \tag{13}$$
$$\frac{\partial u}{\partial x}=\frac{\partial U}{\partial x}e^{a(t)x+b(t)}+U(x,t)e^{a(t)x+b(t)}a(t) \tag{14}$$
$$\frac{\partial^2 u}{\partial x^2}=\frac{\partial^2 U}{\partial x^2}e^{a(t)x+b(t)}+2\frac{\partial U}{\partial x}e^{a(t)x+b(t)}a(t)+U(x,t)e^{a(t)x+b(t)}a^2(t) \tag{15}$$

Substituting (13)-(15) into our PDE equation (10), we get:

$$\begin{aligned}&\frac{\partial U}{\partial t}e^{a(t)x+b(t)}+U(x,t)e^{a(t)x+b(t)}(a'(t)x+b'(t))+\\&\qquad+\frac{\sigma^2}{2}\cdot\left(\frac{\partial^2 U}{\partial x^2}e^{a(t)x+b(t)}+2\frac{\partial U}{\partial x}e^{a(t)x+b(t)}a(t)+U(x,t)e^{a(t)x+b(t)}a^2(t)\right)+\\&\quad+\left(r-\frac{\sigma^2}{2}-\frac{L'(t)}{L(t)}\right)\cdot\left(\frac{\partial U}{\partial x}e^{a(t)x+b(t)}+U(x,t)e^{a(t)x+b(t)}a(t)\right)-rU(x,t)e^{a(t)x+b(t)}=0\end{aligned}$$

We can simplify this to:

$$\frac{\partial U}{\partial t}+\frac{\sigma^2}{2}\frac{\partial^2 U}{\partial x^2}+\left(r-\frac{\sigma^2}{2}-\frac{L'(t)}{L(t)}+\sigma^2 a(t)\right)\frac{\partial U}{\partial x}+$$

$$+\left[a'(t)x + b'(t) + \frac{\sigma^2 a^2(t)}{2} + a(t)\left(r - \frac{\sigma^2}{2} - \frac{L'(t)}{L(t)}\right) - r\right]U = 0$$

We can choose $a(t), b(t)$ to simplify this above equation by:

$$\begin{cases} r - \frac{\sigma^2}{2} - \frac{L'(t)}{L(t)} + \sigma^2 a(t) = 0 \\ a'(t)x + b'(t) + \frac{\sigma^2 a^2(t)}{2} + a(t)\left(r - \frac{\sigma^2}{2} - \frac{L'(t)}{L(t)}\right) - r = 0 \end{cases}$$

The first equation simplifies to become:

$$a(t) = \frac{1}{2} + \frac{1}{\sigma^2}\left(\frac{L'(t)}{L(t)} - r\right) \tag{16}$$

We note that for $L(t) = E\exp(-q(T-t))$, $\frac{L'(t)}{L(t)} = q$ where q is a constant. It then follows that $a(t)$ is a constant so that $a'(t) = 0$. We then have that the second equation becomes:

$$\begin{aligned} b'(t) + \frac{\sigma^2 a^2(t)}{2} + a(t)\left(r - \frac{\sigma^2}{2} - \frac{L'(t)}{L(t)} - r\right) = 0 \\ b(t) = -\int_t^T \left(r - \frac{\sigma^2 a^2(s)}{2} - a(s)\left(r - \frac{\sigma^2}{2} - \frac{L'(s)}{L(s)}\right)\right)ds \\ b(t) = -\int_t^T \left(r - \frac{\sigma^2 a^2(s)}{2} + a(s)\cdot\frac{\sigma^2 a(s)}{2}\right)ds \\ b(t) = -\int_t^T \left(r + \frac{\sigma^2 a^2(s)}{2}\right)ds \end{aligned}$$

The above PDE then becomes:

$$\frac{\partial U}{\partial t} + \frac{\sigma^2}{2}\cdot\frac{\partial^2 U}{\partial x^2} = 0 \tag{17}$$

Note that if we let $\tau = \sigma^2(T-t)$, we have that:

$$\frac{\partial U}{\partial \tau} - \frac{1}{2}\frac{\partial^2 U}{\partial x^2} = 0, \quad 0 < \tau < T, x > 0 \tag{18}$$

We have that $a(t) \equiv \frac{1}{2} + \frac{q-r}{\sigma^2}$, $b(T) = -\int_T^T \left(r + \frac{\sigma^2 a^2(s)}{2}\right)ds = 0$
Note that we also have terminal conditions:

$$\begin{cases} u(x,0) = U(x,\tau=T)e^{a(T)x+b(T)} = \max\{e^x L(T) - D, 0\} \\ u(0, t = T - \frac{\tau}{\sigma^2}) = U(0,\tau)e^{a(T-\frac{\tau}{\sigma^2})x + b(T-\frac{\tau}{\sigma^2})} = 0 \end{cases}$$

We then have the terminal conditions:

$$\begin{cases} U(x,0) = e^{-a(T)x}\max\{e^x L(T) - D, 0\} \\ U(0,\tau) = 0 \end{cases} \tag{19}$$

2.2 Green's Function

We note that (17) gives us the heat equation of form $\frac{\partial U}{\partial \tau} = c\cdot\nabla^2 U$ with constant c. We note that in one-dimension, this has fundamental solution, or heat kernel, by Green's function:

$$\Phi(x,\tau) = \frac{1}{\sqrt{4\pi c\tau}}\exp\left(-\frac{x^2}{4c\tau}\right)$$

From this, one may retrieve general solution of the heat equation with initial conditions $U(x,0) = g(x)$ for $(x,\tau) \in \mathbb{R} \times [0,\infty)$ by applying a convolution:

$$U(x,\tau) = \int_{\mathbb{R}} \Phi(x-y,\tau) g(y) dy$$

Here, $c = \frac{1}{2}$ and $g(x)$ is the dirac-delta function such that boundary condition (19) holds, so that we have:

$$U_0(x,\tau) = \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi\tau}} \exp\Big(- \frac{(x-y)^2}{2\tau}\Big) g(y) dy$$
$$\text{with initial conditions}$$
$$U_0(x,0) = \int_{\mathbb{R}} U_0(s)\delta(s-x) ds = e^{-a(T)x} \max\{e^x L(T) - D, 0\}$$

Here, we find the closed form solution by extending the upper-half plane conditions of our PDE (17) onto the an infinitely extending slab by using the **method of images**. Note, then instead of solving (17) on $S > L_t$ which is $x > 0$, we consider:

$$U(x,0) = U_0(x) - U_0(-x) \tag{20}$$

with boundary conditions:

$$U(x,0) = \begin{cases} e^{-a(T)x} \max\{e^x L(T) - D, 0\} & \text{for } x > 0 \\ -e^{-a(T)(-x)} \max\{e^{-x} L_(T) - D, 0\} & \text{for } x < 0 \end{cases}$$

We then have general solution:

$$\begin{aligned}
u(x,t) &= U(x,t) e^{a(t)x + b(t)} = e^{a(t)x+b(t)} \Big(U_0(x,t) - U_0(-x,t) \Big) \\
&= e^{a(t)x+b(t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x-y)^2}{2\sigma^2(T-t)}) U(y,0) dy \\
&\quad - e^{a(t)x+b(t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x+y)^2}{2\sigma^2(T-t)}) U(y,0) dy \\
&= e^{a(t)x+b(t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x-y)^2}{2\sigma^2(T-t)}) e^{-a(T)y} \Big[e^y L(T) - D \Big]^+ dy \\
&\quad - e^{a(t)x+b(t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x+y)^2}{2\sigma^2(T-t)}) e^{-a(T)y} \Big[e^y L(T) - D \Big]^+ dy \\
&= I_1 + I_2
\end{aligned}$$

where

$$I_1 := e^{a(t)x+b(t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x-y)^2}{2\sigma^2(T-t)}) e^{-a(T)y} \Big[e^y L(T) - D \Big]^+ dy \tag{21}$$

$$I_2 := -e^{a(t)x+b(t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x+y)^2}{2\sigma^2(T-t)}) e^{-a(T)y} \Big[e^y L(T) - D \Big]^+ dy \tag{22}$$

Here recall that $a \equiv \frac{1}{2} + \frac{q-r}{\sigma^2}$ and that

$$b(t) = -\int_t^T \Big(r + \frac{\sigma^2 a^2}{2} \Big) ds = -\Big(r + \frac{\sigma^2 a^2}{2} \Big) \cdot \Big(T - t \Big)$$

Here we omit the calculations for the integration. The in depth calculations are found in the appendix. We then have the following analytical results.

2.3 Analytical Results

We summarize our analytical results as follows:

$$V(S,t) = V_{vanilla}(S,t) - S^{\frac{2q-2r}{\sigma^2}}\Big(E\exp\left(-q(T-t)\right)\Big)^{\frac{2r-2q}{\sigma^2}+1} N(d_+^2) + DS^{1+\frac{2q-2r}{\sigma^2}}\Big(Ee^{-q(T-t)}\Big)^{-1+\frac{2r-2q}{\sigma^2}} e^{-r(T-t)}N(d_-^2)$$

where we have:

$$\begin{aligned}
d_-^1 &= \frac{\ln\frac{S}{D} + (r - \frac{\sigma^2}{2})(T-t)}{\sqrt{\sigma^2(T-t)}} \\
d_-^2 &= \frac{\ln\left(\frac{E^2}{S\cdot D}\right) - 2q(T-t) + (r - \frac{\sigma^2}{2})(T-t)}{\sqrt{\sigma^2(T-t)}} \\
d_+^1 &= d_-^1 + \sigma\sqrt{T-t} \\
d_+^2 &= d_-^2 + \sigma\sqrt{T-t}
\end{aligned}$$

and $V_{vanilla}(S,t)$ is the value of a vanilla call option with no dividend rate, interest rate r, strike price D, and maturity T.

2.4 Special Cases

We consider the following special cases as a sanity check to our analytical formulation:

- We first check our formula for $E = 0$, i.e. where we have no barrier option and the option becomes entirely vanilla. We note that $d_-^1, d_-^2, d_+^1, d_+^2$ remain unchanged but that:

$$\begin{aligned}
V(S,t) =& V_{vanilla}(S,t) - S^{\frac{2q-2r}{\sigma^2}}\Big(E\exp\left(-q(T-t)\right)\Big)^{2\frac{r-q}{\sigma^2}+1} N(d_+^2) \\
&+ DS^{1+2\frac{q-r}{\sigma^2}}\Big(Ee^{-q(T-t)}\Big)^{-1+\frac{2r-2q}{\sigma^2}} e^{-r(T-t)}N(d_-^2) \\
&= V_{vanilla}(S,t) - S^{\frac{2q-2r}{\sigma^2}}\Big(0\Big)^{2\frac{r-q}{\sigma^2}+1} N(d_+^2) + DS^{1+\frac{2q-2r}{\sigma^2}}\Big(0\Big)^{-1+\frac{2r-2q}{\sigma^2}} e^{-r(T-t)}N(d_-^2) \\
&= V_{vanilla}(S,t)
\end{aligned}$$

  As expected.

  However, note that the above consideration is only stable for $r \leq q$, which makes sense in a practical sense as the dividend rate should not be greater than the interest rate.

- We then consider the situation where the barrier is constant with $q = 0$. We note that $d_-^1, d_-^2, d_+^1, d_+^2$ remain unchanged but that:

$$\begin{aligned}
V(S,t) =& V_{vanilla}(S,t) - S^{\frac{-2r}{\sigma^2}} E^{2\frac{r}{\sigma^2}+1} N(d_+^2) + DS^{1+\frac{-r}{\sigma^2}} E^{-1+\frac{2r}{\sigma^2}} e^{-r(T-t)} N(d_-^2) \\
&= V_{vanilla}(S,t) - S^{\frac{-2r}{\sigma^2}+1} E^{\frac{2r}{\sigma^2}-1}\left(\frac{E^2}{S}\right) N(d_+^2) + De^{-r(T-t)} S^{1+\frac{-2r}{\sigma^2}} E^{-1+\frac{2r}{\sigma^2}} N(d_-^2) \\
&= V_{vanilla}(S,t) - \left(\frac{S}{E}\right)^{\frac{-2r}{\sigma^2}+1}\left(\frac{E^2}{S} N(d_+^2) - De^{-r(T-t)} N(d_-^2)\right) \\
&= V_{vanilla}(S,t) - \left(\frac{S}{E}\right)^{\frac{-2r}{\sigma^2}+1} V_{vanilla}(\frac{E^2}{S}, t) = V_{barrier}(S,t)
\end{aligned}$$

  Where $V_{barrier}$ is the value of a standard barrier with barrier E, interest rate r, strike price D, and maturity T as expected.

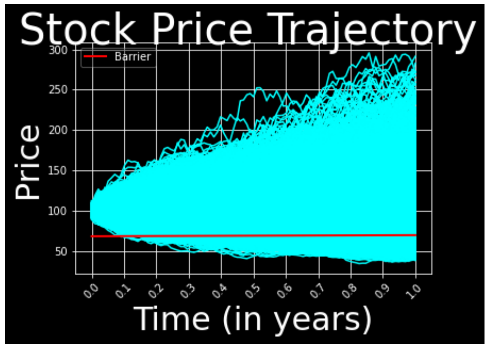
- We then check our formula for $r = q$. This is more exploration than for a sanity check. We note that $d_-^1, d_-^2, d_+^1, d_+^2$ remain unchanged but that:

$$\begin{aligned} V(S,t) =& V_{vanilla}(S,t) - S^0 \left(E \exp(-q(T-t)) \right)^{0+1} N(d_+^2) + DS^{1+0} \left(E^{-q(T-t)} \right)^{-1+0} e^{-r(T-t)} N(d_-^2) \\ &= V_{vanilla}(S,t) - E \exp(-r(T-t)) N(d_+^2) + \frac{DS}{E} N(d_-^2) \end{aligned}$$

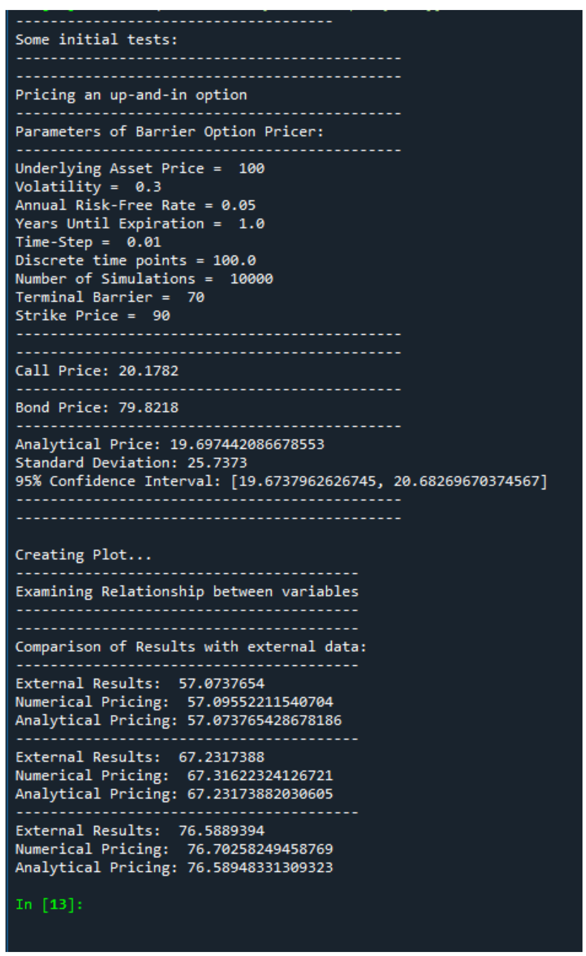
Where d_-^2, d_+^2 are modified to be:
$d_-^2 = \frac{\ln\left(\frac{E^2}{S \cdot D}\right) + (r + \frac{\sigma^2}{2})(T-t)}{\sqrt{\sigma^2 (T-t)}}$ and $d_+^2 = d_-^2 + \sigma\sqrt{T-t}$

3 Numerical Analysis

3.1 Monte Carlo Simulations



We conduct a numerical analysis to verify our analytical results. Here, we use Monte Carlo methods to test out our expected barrier option prices. Note that the Monte Carlo method may not be that efficient here due to its $O(n^{-1/2})$ convergence, as opposed to $O(n^{-2/d})$ convergence of some integration and finite difference methods. Future considerations for faster convergence could include quasi-Monte Carlo methods or analogous approaches. We then conduct a series of tests of our numerical analysis as well as analytical formula against pre-existing data collected externally with the parameters listed below:



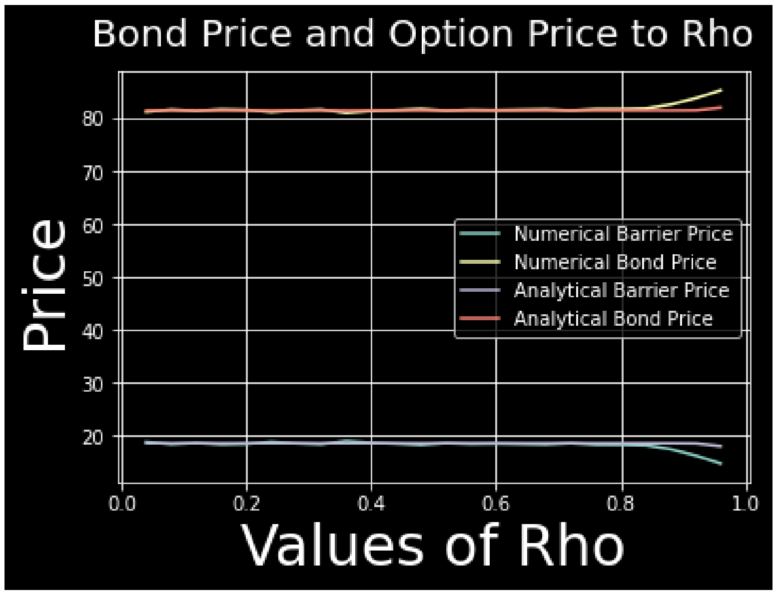
```
------------------------------------
Some initial tests:
--------------------------------------------
--------------------------------------------
Pricing an up-and-in option
--------------------------------------------
Parameters of Barrier Option Pricer:
--------------------------------------------
Underlying Asset Price =  100
Volatility =  0.3
Annual Risk-Free Rate = 0.05
Years Until Expiration =  1.0
Time-Step =  0.01
Discrete time points = 100.0
Number of Simulations =  10000
Terminal Barrier =  70
Strike Price =  90
--------------------------------------------
--------------------------------------------
Call Price: 20.1782
--------------------------------------------
Bond Price: 79.8218
--------------------------------------------
Analytical Price: 19.697442086678553
Standard Deviation: 25.7373
95% Confidence Interval: [19.6737962626745, 20.68269670374567]
--------------------------------------------
--------------------------------------------

Creating Plot...
---------------------------------------
Examining Relationship between variables
---------------------------------------
---------------------------------------
Comparison of Results with external data:
---------------------------------------
External Results:  57.0737654
Numerical Pricing:  57.09552211540704
Analytical Pricing: 57.073765428678186
---------------------------------------
External Results:  67.2317388
Numerical Pricing:  67.31622324126721
Analytical Pricing: 67.23173882030605
---------------------------------------
External Results:  76.5889394
Numerical Pricing:  76.70258249458769
Analytical Pricing: 76.58948331309323

In [13]:
```

Firstly, we note that our analytical formula is proven to be rather accurate when compared to an external source. Given the credibility of analytical pricing, We can see that our Monte Carlo approximations are rather weak with our simulation iteration count at $n = 10000$. This could be greatly strengthened with more iterations but is rather exhaustive for the purpose of this paper. Substitute numerical methods may perform a lot better and could be considered in the future.

We then have a comparison of the bond prices and barrier option prices to the varying values of $E = \rho \cdot D$ with ρ is from 0.1 to 1.0. That is, we see how our numerical analysis performs as the barrier gets closer to the strike price. Here, I did not consider the case where $\rho = 0$, as the analytical formula does not simplify immediately on a numerical level. The results are as follows:



We can observe that our numerical approximations work much better as the barrier is further from the strike price. As $E \to D$, we note that there is a significant divergence of our approximation from our analytical formula. This could have some implications regarding the usage of Monte Carlo methods on pricing barrier options with time-dependent barriers. Of course, this could be a future investigation on numerical analysis.

4 Appendix

4.1 Integration Procedure of the BS PDE

$$
\begin{aligned}
I_1 &= e^{ax-r(T-t)-\frac{\sigma^2 a^2}{2}(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x-y)^2}{2\sigma^2(T-t)}) e^{-ay} \Big[e^y L(T) - D\Big]^+ dy \\
&= e^{-r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp\Big(- \frac{(x-y)^2}{2\sigma^2(T-t)} + a(x-y) - \frac{\sigma^2 a^2}{2}(T-t)\Big) \Big[e^y L(T) - D\Big]^+ dy \\
&= e^{-r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x-y-a\sigma^2(T-t)\big)^2\Big\} \Big[e^y L(T) - D\Big]^+ dy \\
&= e^{-r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x-y-(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\} \Big[e^y L(T) - D\Big] dy \\
&= F_1 + F_2
\end{aligned}
$$

where:

$$
\begin{aligned}
F_1 &:= e^{-r(T-t)} \frac{L(T)}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty \exp\Big\{ - \frac{(x-y)^2}{2\sigma^2(T-t)} + a(x-y) - \frac{\sigma^2 a^2}{2}(T-t)\Big\} e^y dy \\
&= e^{-r(T-t)} \frac{L(T)}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty \exp\Big\{ - \frac{(x-y)^2}{2\sigma^2(T-t)} + (-\frac{1}{2}+\frac{q-r}{\sigma^2})(x-y) \\
&\qquad + (x-y) - \frac{\sigma^2(-\frac{1}{2}+\frac{q-r}{\sigma^2})^2}{2}(T-t) - (q-r)(T-t)\Big\} e^y dy \\
&= e^{-r(T-t)} \frac{L(T)}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty e^{x-y+y-(q-r)(T-t)} \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x-y-(-\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\} dy \\
&\text{Note that } e^x = \frac{S}{L(t)} \text{ so that:} \\
&= S e^{-q(T-t)} \frac{L(T)}{L(t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x-y-(-\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\} dy \\
&= S e^{-q(T-t)} e^{q(T-t)} \int_{-\infty}^{z=\frac{x-\ln\frac{D}{L(T)}-(-\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \\
&= S N(d^1_+)
\end{aligned}
$$

$$
\begin{aligned}
F_2 &:= -e^{-r(T-t)} \frac{D}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x-y-(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\} dy \\
&= -D e^{-r(T-t)} \int_{-\infty}^{z=\frac{x-\ln\frac{D}{L(T)}-(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \\
&= -D e^{-r(T-t)} N(d^1_-)
\end{aligned}
$$

where

$$
\begin{aligned}
d^1_- &= \frac{x-\ln\frac{D}{L(T)}-(\frac{q-r}{\sigma^2}-\frac{1}{2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}} \\
&= \frac{\ln\frac{S}{L(t)}-\ln\frac{D}{L(T)}-(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\ln \frac{S}{D} - \ln \frac{L(t)}{L(T)} - (\frac{1}{2} + \frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}} \\
&= \frac{\ln \frac{S}{D} - \ln\left(\exp\left(-q(T-t)\right)\right) - (\frac{1}{2} - \frac{r}{\sigma^2})\sigma^2(T-t) - q\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}} \\
&= \frac{\ln \frac{S}{D} + (\frac{r}{\sigma^2} - \frac{\sigma^2}{2})(T-t)}{\sqrt{\sigma^2(T-t)}} \\
d_+^1 &= d_-^1 + \sigma\sqrt{T-t}
\end{aligned}
$$

On the other hand, we also have:

$$
\begin{aligned}
I_2 &= -e^{ax - r(T-t) - \frac{\sigma^2 a^2}{2}(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp(-\frac{(x+y)^2}{2\sigma^2(T-t)}) e^{-ay} \Big[e^y L(T) - D\Big]^+ dy \\
&= -e^{2ax - r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp\Bigg(- \frac{(x+y)^2}{2\sigma^2(T-t)} - a(x+y) - \frac{\sigma^2 a^2}{2}(T-t)\Bigg) \Big[e^y L(T) - D\Big]^+ dy \\
&= -e^{2ax - r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_0^\infty \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x + y + a\sigma^2(T-t)\big)^2\Big\} \Big[e^y L(T) - D\Big]^+ dy \\
&= -e^{2ax - r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x + y + (\frac{1}{2} + \frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\} \Big[e^y L(T) - D\Big] dy \\
&= H_1 + H_2
\end{aligned}
$$

$$
\begin{aligned}
H_1 &:= -e^{2(\frac{1}{2} + \frac{q-r}{\sigma^2})x - r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty e^y L(T) \exp\Big\{ - \frac{(x+y)^2}{2\sigma^2(T-t)} - a(x+y) - \frac{\sigma^2 a^2}{2}(T-t)\Big\} \\
&= -e^{2(\frac{1}{2} + \frac{q-r}{\sigma^2})x - r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty e^y L(T) \exp\Big\{ - \frac{(x+y)^2}{2\sigma^2(T-t)} \\
&\qquad - (\frac{1}{2} + \frac{q-r}{\sigma^2})(x+y) - \frac{\sigma^2(\frac{1}{2} + \frac{q-r}{\sigma^2})^2}{2}(T-t)\Big\} \\
&= -e^{2(\frac{1}{2} + \frac{q-r}{\sigma^2})x - r(T-t)} \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty e^y L(T) \exp\Big\{ - \frac{(x+y)^2}{2\sigma^2(T-t)} \\
&\qquad - (-\frac{1}{2} + \frac{q-r}{\sigma^2})(x+y) - (x+y) - \frac{\sigma^2(-\frac{1}{2} + \frac{q-r}{\sigma^2})^2}{2}(T-t) - (q-r)(T-t)\Big\} \\
&= -e^{2\frac{q-r}{\sigma^2}x - q(T-t)} \frac{L(T)}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty e^{-x-y+y+x} \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x + y + (-\frac{1}{2} + \frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\} dy
\end{aligned}
$$

Note that $e^x = \dfrac{S}{L(t)}$ so that:

$$
\begin{aligned}
&= -e^{2\frac{q-r}{\sigma^2}x - q(T-t)} L(T) \frac{1}{\sqrt{2\pi\sigma^2(T-t)}} \int_{\ln(\frac{D}{L(T)})}^\infty \exp\Big\{ - \frac{1}{2(T-t)\sigma^2}\big(x + y + (-\frac{1}{2} + \frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\} e^y dy \\
&= -e^{2\frac{q-r}{\sigma^2}x - q(T-t)} E \int_{-\infty}^{z = \frac{-x - \ln\frac{D}{L(T)} - (-\frac{1}{2} + \frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \\
&= -\Bigg(\frac{S}{L(t)}\Bigg)^{2\frac{q-r}{\sigma^2}} E e^{-q(T-t)} N(d_+^2) \\
&= -\Bigg(S\Bigg)^{2\frac{q-r}{\sigma^2}} \Bigg(E \exp\left(-q(T-t)\right)\Bigg)^{2\frac{r-q}{\sigma^2}+1} N(d_+^2)
\end{aligned}
$$

$$
\begin{aligned}
H_2 :=& De^{2ax-r(T-t)}\frac{1}{\sqrt{2\pi\sigma^2(T-t)}}\int_{\ln(\frac{D}{L(T)})}^{\infty}\exp\Big\{-\frac{1}{2(T-t)\sigma^2}\big(x+y+(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)\big)^2\Big\}\\
=& De^{2ax-r(T-t)}\int_{-\infty}^{z=\frac{-x-\ln\frac{D}{L(T)}-(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}}}\frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}dz\\
=& D\bigg(\frac{S}{Ee^{-q(T-t)}}\bigg)^{1+2\frac{q-r}{\sigma^2}}e^{-r(T-t)}N(d_-^2)\\
=& D\bigg(S\bigg)^{1+\frac{2q-2r}{\sigma^2}}\bigg(Ee^{-q(T-t)}\bigg)^{-1+\frac{2r-2q}{\sigma^2}}e^{-r(T-t)}N(d_-^2)
\end{aligned}
$$

where

$$
\begin{aligned}
d_-^2 =& \frac{-x-\ln\frac{D}{L(T)}-(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}}\\
=& \frac{-\ln(\frac{S}{L(t)})-\ln\frac{D}{L(T)}-(\frac{1}{2}+\frac{q-r}{\sigma^2})\sigma^2(T-t)}{\sqrt{\sigma^2(T-t)}}\\
=& \frac{\ln(\frac{L(t)}{S})+\ln\frac{L(T)}{D}-(\frac{1}{2}-\frac{r}{\sigma^2})\sigma^2(T-t)-q(T-t)}{\sqrt{\sigma^2(T-t)}}\\
=& \frac{\ln(\frac{E^2}{S\cdot D}\cdot\exp(-q(T-t)))-(\frac{1}{2}-\frac{r}{\sigma^2})\sigma^2(T-t)-q(T-t)}{\sqrt{\sigma^2(T-t)}}\\
=& \frac{\ln(\frac{E^2}{S\cdot D})-2q(T-t)+(r-\frac{\sigma^2}{2})(T-t)}{\sqrt{\sigma^2(T-t)}}\\
d_+^2 =& d_-^2+\sigma\sqrt{T-t}
\end{aligned}
$$

4.2 Code for Monte Carlo Method

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Dec 14 23:00:50 2020

@author: raymo
"""

#import libraries
import numpy as np
from matplotlib import pyplot as plt
from matplotlib import style
from random import gauss
from math import exp, sqrt, log
from scipy.stats import norm

#-----------------------------------------------
#Class for BS Parameters
#-----------------------------------------------

class BS_Param():
    """parameters of BS PDE"""

    #-------------------------
    def __init__(self,S, v, r, T, D, delta_t, simulations, q, E):
        """initialization of parameters"""

        self.S = S                              #underlying asset price
        self.v = v                              #volatility
        self.r = r                              #risk free rate
        self.T = float(T)                       #years until maturity
        self.D = D                              #strike price
        self.delta_t = delta_t                  #timestep
        self.N = self.T/self.delta_t            #Number of discrete time points
        self.simulations = simulations          #num of simulations
        self.B = []                             #Barrier price
        self.q = q                              #Barrier discount
        self.E = E                              #Barrier terminal value


    #--------------------------
    def init_B(self):
        t = range(0,int(self.N)-1)
        t = self.delta_t*np.array(t)
        self.B = self.E*np.exp(-1*self.q*(self.T-t))
        #print((self.B))

    #--------------------------
    def print_parameters(self):
        """print parameters"""

        print ("--------------------------------------------")
        print ("--------------------------------------------")
        print ("Pricing an up-and-in option")
        print ("--------------------------------------------")
        print ("Parameters of Barrier Option Pricer:")
        print ("--------------------------------------------")
        print ("Underlying Asset Price = ", self.S)
```

```python
        print ("Volatility = ", self.v)
        print ("Annual Risk-Free Rate =", self.r)
        print ("Years Until Expiration = ", self.T)
        print ("Time-Step = ", self.delta_t)
        print ("Discrete time points =", self.N)
        print ("Number of Simulations = ", self.simulations)
        print ('Terminal Barrier = ', self.E)
        print ('Strike Price = ', self.D)
        print ("--------------------------------------------")
        print ("--------------------------------------------")


#-----------------------------------------------
#Class for Monte Carlo
#-----------------------------------------------

class down_and_out_mc(BS_Param):
    """This is the class to price the barrier option
       defined as an up and in option. Paramaters are
       fed in as a subclass"""

    #-------------------------
    def __init__(self, S, v, r, T, D, delta_t, simulations, q, E):
        """initialize the class including the subclass"""
        BS_Param.__init__(self, S, v, r, T, D, delta_t, simulations, q, E)
        BS_Param.init_B(self)
        self.payoffs = []
        self.price_trajectories = []
        self.cp = 0


    #-------------------------
    def call_payoff(self,s):
        """use to price a call"""
        self.cp = max(s - self.D, 0.0)
        return self.cp



    #-------------------------
    def calculate_payoff_vector(self):
        """Main iterative loop to run the
        Monte Carlo simulation. Returns a vector
        of different potential payoffs"""

        for i in range(0, self.simulations):
            self.stock_path = []
            self.S_j = self.S
            for j in range(0, int(self.N-1)):
                self.xi = gauss(0,1.0)

                self.S_j *= (exp((self.r-.5*self.v*self.v) * self.delta_t + self.v *sqrt(self.delta_t) * self.xi))

                self.stock_path.append(self.S_j)
            self.price_trajectories.append(self.stock_path)
            worthless = False
            for j in range(int(self.N)-1):
```

```python
                if self.stock_path[j] <= self.B[j]:
                    worthless = True
            if worthless:
                self.payoffs.append(0)
            else:
                self.payoffs.append(self.call_payoff(self.stock_path[-1]))


        return self.payoffs

    #--------------------------
    def compute_price(self):
        """Uses payoff vector and discount
        factor to compute the price of the
        option. Numpy used for efficiency"""
        self.np_payoffs = np.array(self.payoffs, dtype=float)
        self.np_Vi = exp(-self.r * self.T)*self.np_payoffs
        self.price = np.average(self.np_Vi)

    #--------------------------
    def print_price(self):
        """prints the option price"""
        print (str("Call Price: %.4f") % self.price)
        print ("--------------------------------------------")
        print (str("Bond Price: %.4f") % (self.S - self.price))
        print ("--------------------------------------------")

    #--------------------------
    def get_price(self):
        return self.price

    #--------------------------
    def calc_statistics(self):
        """uses payoffs and price to calc
        variance, standard deviation, and
        a 95% confidence interval for price"""
        self.sd = np.std(self.np_Vi)
        #95% C.I. uses 1.96 z-value
        self.CI = [self.price - (1.96*self.sd/sqrt(float(self.simulations))),
                   self.price + (1.96*self.sd/sqrt(float(self.simulations)))]

    #--------------------------
    def print_statistics(self):
        """prints option statistics that were
        calculated to the terminal"""
        print ("Standard Deviation: %.4f" % self.sd)
        print ("95% Confidence Interval:", self.CI)
        print ("--------------------------------------------")
        print ("--------------------------------------------\n")
        #print ('E',self.D)

    #--------------------------
    def plot_trajectories(self):
        """uses matplotlib to plot the trajectory
        of each individual stock stored in earlier
        trajectory array"""
        print ("Creating Plot...")
        #use numpy to plot
```

```python
        self.np_price_trajectories = np.array(self.price_trajectories, dtype=float)
        self.times = np.linspace(0, int(self.T), int(self.N-1))

        #style/plot/barrier line
        style.use('dark_background')

        self.fig = plt.figure()
        self.ax1 = plt.subplot2grid((1,1),(0,0))
        for sublist in self.np_price_trajectories:
            if max(sublist) > self.B.all():
                self.ax1.plot(self.times,sublist,color = 'cyan')
            else:
                self.ax1.plot(self.times,sublist,color = '#e2fb86')
        plt.plot(self.times,self.B,linewidth=2, color = 'red', label = 'Barrier')

        #rotate and add grid
        for label in self.ax1.xaxis.get_ticklabels():
            label.set_rotation(45)
        self.ax1.grid(True)

        #plotting stuff
        plt.xticks(np.arange(0, self.T+self.delta_t, .1))
        plt.suptitle('Stock Price Trajectory', fontsize=40)
        plt.legend()
        self.leg = plt.legend(loc= 2)
        self.leg.get_frame().set_alpha(0.4)
        plt.xlabel('Time (in years)', fontsize = 30)
        plt.ylabel('Price', fontsize= 30)
        plt.show()


def tDep_Barrier(S, sigma, r, T, D, q, E, t=0):
    d2 = (log(E**2/(S*D))-2*q*T + (r-sigma**2/2)*T)/(sigma**2*sqrt(T))
    d1 = d2 + sigma*sqrt(T)
    n1 = norm.cdf(d1)
    n2 = norm.cdf(d2)
    a = bsm_price(S, D, r, sigma, T)
    Lt = E*exp(-q*T)
    b = (S/Lt)**(2*(q-r)/(sigma**2))*Lt*n1
    c = D*(S/Lt)**(1 + 2*(q-r)/(sigma**2))*exp(-r*T)*n2
    return a - b + c


def bsm_price(s, k, r, sigma, T):
    d1 = (log(s/k)+(r+sigma**2/2)*T)/(sigma*sqrt(T))
    d2 = d1 - sigma*sqrt(T)
    n1 = norm.cdf(d1)
    n2 = norm.cdf(d2)
    return n1*s - n2*k*exp(-r*T)

#----------------------------------------------
#Main Program
#----------------------------------------------

def main():
    #print(bsm_price(100, 80, 0.02, 0.3, 1))
    #print(tDep_Barrier(S=100.0, sigma=0.30, r=0.05, T=1.0, D=70, q=0, E=70, t=0))
    S = 100
```