Computational Finance

HANDIN 2

DENIZ CAN TEKCIFTCI
KXL525

Computational Finance
Hand-In 2
Deniz Can Tekciftci

Useful information

The python program has the following dependencies:

- numpy
- scipy.optimize
- matplotlib.pyplot
- seaborn
- pandas

The packages can be installed with pip using the command:

```
pip install numpy scipy matplotlib seaborn pandas
```

The code answers to all the question can be found in the main.py file. Be aware though, running the program took quite a while to complete on my machine. The file can be run with the command:

```
python main.py
```

Functions used for answering "Cash money"-question have been placed in the cashMoneyFunctions.py file.

The class used for solving questions in "Finite difference goes American" can be found in CN.py file.

If you plan on running the code related to question 5 and 7 in "Cash money", I suggest lowering the number of paths by an order of magnitude or two. 100.000 paths took around 45 minutes to complete on my machine. Question 4 also takes quite a while to run with 50 data points, but still an order of magnitude faster than the other two.

Cash money

1)

In order to calculate the zero coupon prices for maturities T = 1, 5, 10, 20, 30, 50, the formula from Appendix A.2:

$$P(0,T) = e^{A(0,T)-B(0,T)r}$$

Has been implemented with the function for A(0, T) and B(0, T). Running the function for the maturities outputs:

```
P(0, 1) = 1.0044695327444173

P(0, 5) = 1.0128289770633

P(0, 10) = 1.0064164936040143

P(0, 20) = 0.9603895952808099

P(0, 30) = 0.8960191792168013

P(0, 50) = 0.7664095425597687
```

2)

It is known, that $\int_0^T r_s ds$ is normally distributed with mean and variance:

$$\mu = T\theta + \frac{r_0 - \theta}{\kappa}(1 - e^{-\kappa T})$$

$$var(\int_0^T r_s ds) = \frac{\sigma^2}{\kappa^3}(\kappa T + 2e^{-\kappa T} - \frac{1}{2}e^{-2\kappa T} - 1.5)$$

This fact can be used to speed up the Monte Carlo simulations, since we will not need to use Euler discretization, instead simply generating numbers distributed as above. And since we know $P(0,T) = e^{-\int_0^T r_s ds}$, we have found the price of the zero coupon. Simulating with 100.000 paths gives the following result:

P(0, 1): mean: 1.0044883539257727, 95%-CI: [1.004463905877748, 1.0045128019737974]

P(0, 5): mean: 1.0127531638188674, 95%-CI: [1.0124896994621637, 1.013016628175571]

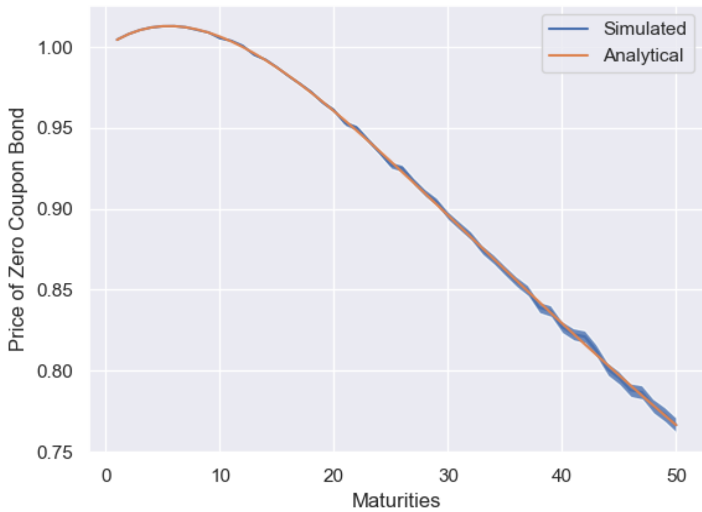
P(0, 10): mean: 1.0054169361132694, 95%-CI: [1.0047190132986445, 1.0061148589278943]

P(0, 20): mean: 0.9608921056081221, 95%-CI: [0.959165518477327, 0.9626186927389172]

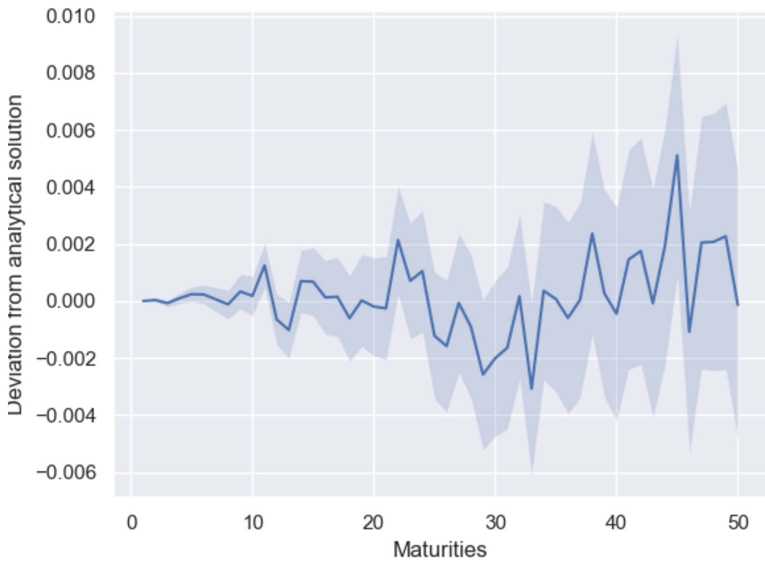
P(0, 30): mean: 0.8958437672155607, 95%-CI: [0.893068867578171, 0.8986186668529503]

P(0, 50): mean: 0.7664464257139905, 95%-CI: [0.7617194915419879, 0.771173359885993]

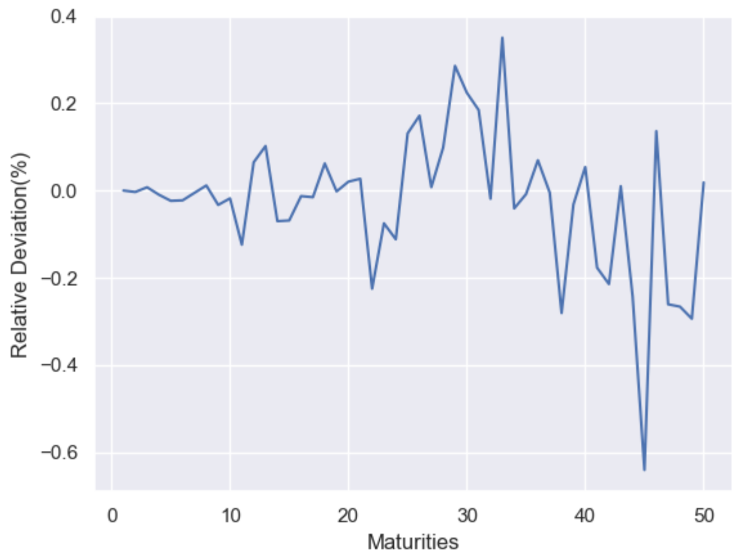
And plotted for every whole year maturity in the range:



Above is also drawn the confidence interval: the blue area. Clearly having higher variance for larger maturities, which is caused by the diffusion term in the model's dynamics. Which is seen more clearly through the residual plot below.



The same pattern is seen when plotting the the residuals relative to the price seen below.



To help with reducing the variance in the Monte Carlo sampling antithetic sampling is applied. Doing so gives the following result:

```
Standard sample variance: 1.56e-05, antithetic sample variance: 1.218861e-10, efficiency ratio: 128395.503
Standard sample variance: 0.0018166, antithetic sample variance: 1.569409e-06, efficiency ratio: 1157.47698
Standard sample variance: 0.0127967, antithetic sample variance: 7.993313024e-05, efficiency ratio: 160.09219
Standard sample variance: 0.0772222, antithetic sample variance: 0.0029725160, efficiency ratio: 25.97873
Standard sample variance: 0.199225, antithetic sample variance: 0.01983652, efficiency ratio: 10.04335
Standard sample variance: 0.5870745, antithetic sample variance: 0.15221487187862337, efficiency ratio: 3.85
```

In the scenario where the sampling is under a time constraint, and the sample is wanted as variance reduced as possible, this means, that antithetic sampling would be more efficient if it is even 3-4 times slower than the regular sampling:

Meaning that, the antithetic sampling would be worth doing with a time constraint, or in any case, really, considering the simplicity of it. On top of that, it only takes around 25-45% longer to do in this case, making it well worth it.

```
Standard sampling time: 0.19782042503, antithetic sampling time: 0.262135744, ratio: 1.3251197092
```

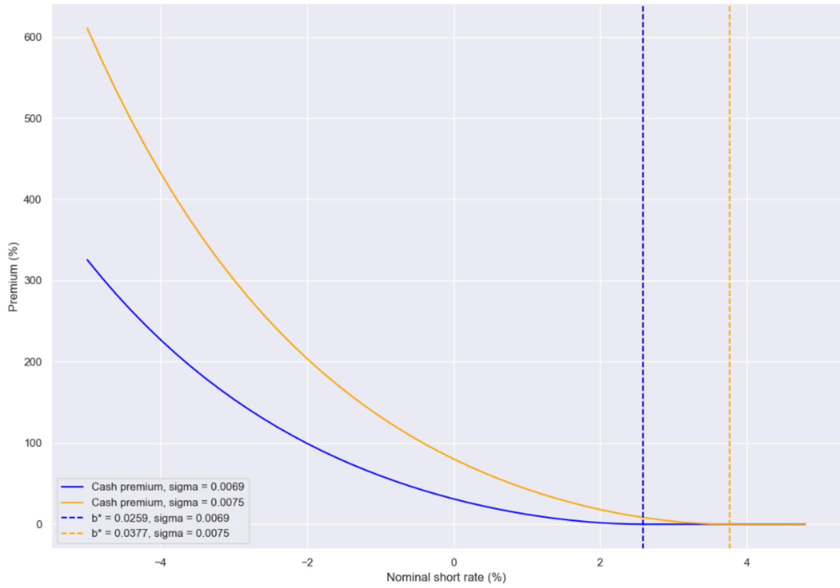
Another bonus of antithetic sampling, is that our normal distribution has mean 0 giving us an unbiased estimation in that regard.

3)

To replicate the Figure 3.1 from the paper, the first thing I need to do, is to find a new b* given the new $\sigma = 0.0075$. The new b*, is found by numerically solving the integral below, and finding for which b this integral is equal to 1.

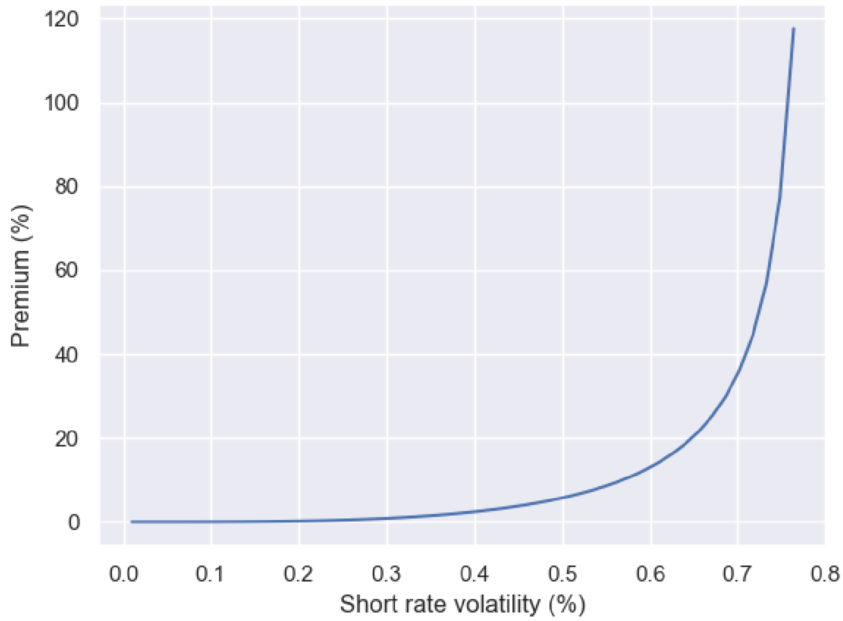
$$C^{\infty}(0) = \int_0^{\infty} P(0,u) \left[f(0,u)\Phi\left(\frac{f(0,u)-b}{v(u)}\right) + v(u)\phi\left(\frac{f(0,u)-b}{v(u)}\right)\right] du.$$

Solving this, gives b* = 0.037678. With the new b*, the integral can be numerically solved for $r_0 = [-5\%, 5\%]$, which plotted gives below figure:



4)

Figure 3.2 is basically a cross-section of Figure 3.1 taken in the σ direction at r(0) = 0. This also means, that b* must be found for every point we want on the new figure, since we are changing σ each time. Doing so, gives the figure:



The figure is plotted for $\sigma = [0.0001, 0.00779]$ since a volatility of 0 results in:

$$v(u)^2 = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa u})$$

becoming 0, which leads to 0-division error when wanting to calculate the marked fractions:

$$C^{\infty}(0) = \int_0^{\infty} P(0,u)\left[f(0,u)\Phi\left(\underline{\frac{f(0,u)-b}{v(u)}}\right) + v(u)\phi\left(\underline{\frac{f(0,u)-b}{v(u)}}\right)\right] du.$$

On top of that, a volatility of 0 does not make much sense in this context either. The upper bound is 0.00779 since $\sigma \approx 0.0078$ makes the price go to infinity.

5)

To calculate $C^{\infty}(0) = E[e^{-\int_0^{\tau} r(s)ds}]$ I sample paths of r(s) using the Euler discretized formula for the nominal short rate under the Vasicek model.

$$dr(t) = \kappa[\theta - r(t)]dt + \sigma dW(t)$$

Then becomes

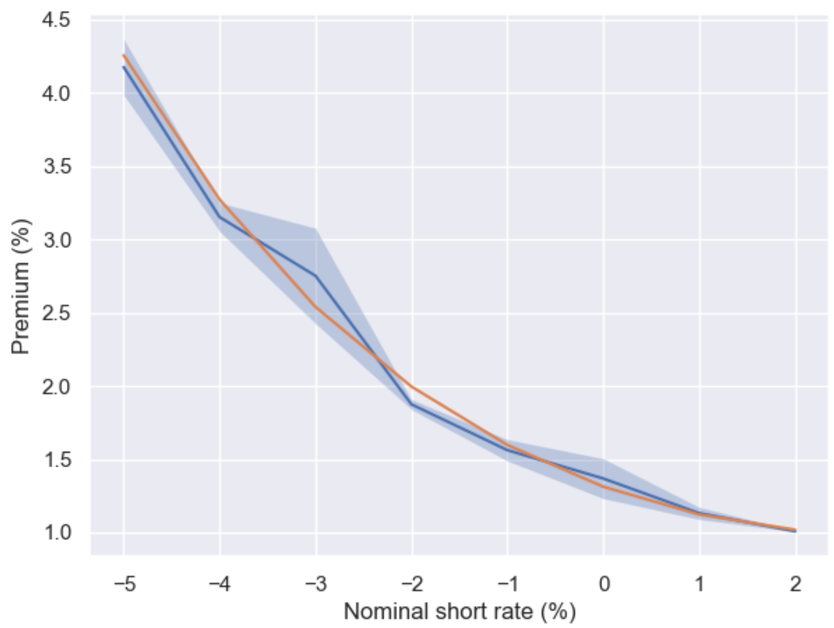
$$\Delta r(t) = \kappa[\theta - r(t)]\Delta + \sigma\sqrt{\Delta}Z$$

Where Δ is the size of the time step and where Z is a standard-normally distributed random variable. This will then be used to simulate paths for r, until we hit the boundary, b*. The integral is then taken from t = 0 to t = τ, at which point r hits the boundary. Below are the integrals calculated using 100.000 paths for each initial r(0) compared to the analytical values calculated through numerically integrating the integral from question 3. Here the time step is 1/20, meaning, we check whether boundary has been hit ones every two

```
r(0) = -0.05, mean price: 4.17550, 95%-CI: [3979438785, 4.371561215], analytical: 4.25587
r(0) = -0.04, mean price: 3.15202, 95%-CI: [3.050827115, 3.253212885], analytical: 3.27223
r(0) = -0.03, mean price: 2.75121, 95%-CI: [2.422333123, 3.080086877], analytical: 2.53917
r(0) = -0.02, mean price: 1.87423, 95%-CI: [1.836282668, 1.912177332], analytical: 1.99556
r(0) = -0.01, mean price: 1.56189, 95%-CI: [1.485995336, 1.637784664], analytical: 1.59701
r(0) = 0.00, mean price: 1.36831, 95%-CI: [1.229169783, 1.507450217], analytical: 1.31241
r(0) = 0.01, mean price: 1.13162, 95%-CI: [1.087348113, 1.175891887], analytical: 1.12216
r(0) = 0.02, mean price: 1.00823, 95%-CI: [1.001905445, 1.014554555], analytical: 1.01858
```

and a half weeks.

Plotting the points gives the following graph, where the orange is the analytical values and the blue the simulated ones, where the transparent blue area is the 95% confidence interval:



As seen, the simulated values tend to follow the analytical ones, but quite often lie a substantial amount away. This may be explained by the time stepping not being continuous or even close for that sake.

6)

Below are Monte Carlo simulated τ^* values with 95% confidence interval for the mean estimation of the distribution:

r(0) = -0.05, mean: 2.545797249999964, 95%-CI: [2.515579713510067, 2.5760147864898615]

r(0) = -0.04, mean: 2.349666749999969, 95%-CI: [2.3204388131919633, 2.3788946868079743]

r(0) = -0.03, mean: 2.1613692499999746, 95%-CI: [2.131558879506336, 2.1911796204936134]

r(0) = -0.02, mean: 1.9073994999999806, 95%-CI: [1.8786363113804854, 1.9361626886194758]

r(0) = -0.01, mean: 1.6296509999999846, 95%-CI: [1.602321472048776, 1.6569805279511933]

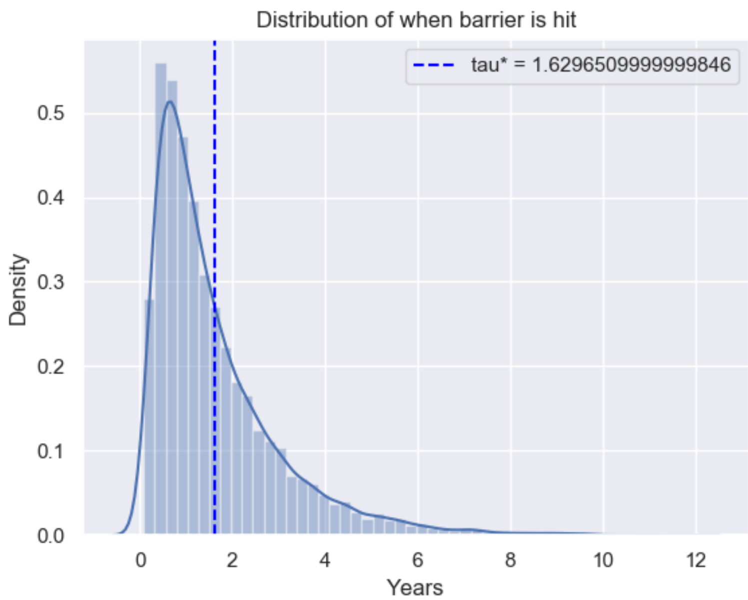
r(0) = 0.00, mean: 1.297797499999989, 95%-CI: [1.27163788822165, 1.3239571117783282]

r(0) = 0.01, mean: 0.9125549999999937, 95%-CI: [0.8887873211686206, 0.9363226788313668]

r(0) = 0.02, mean: 0.416198749999998, 95%-CI: [0.3991550194189199, 0.4332424805810761]

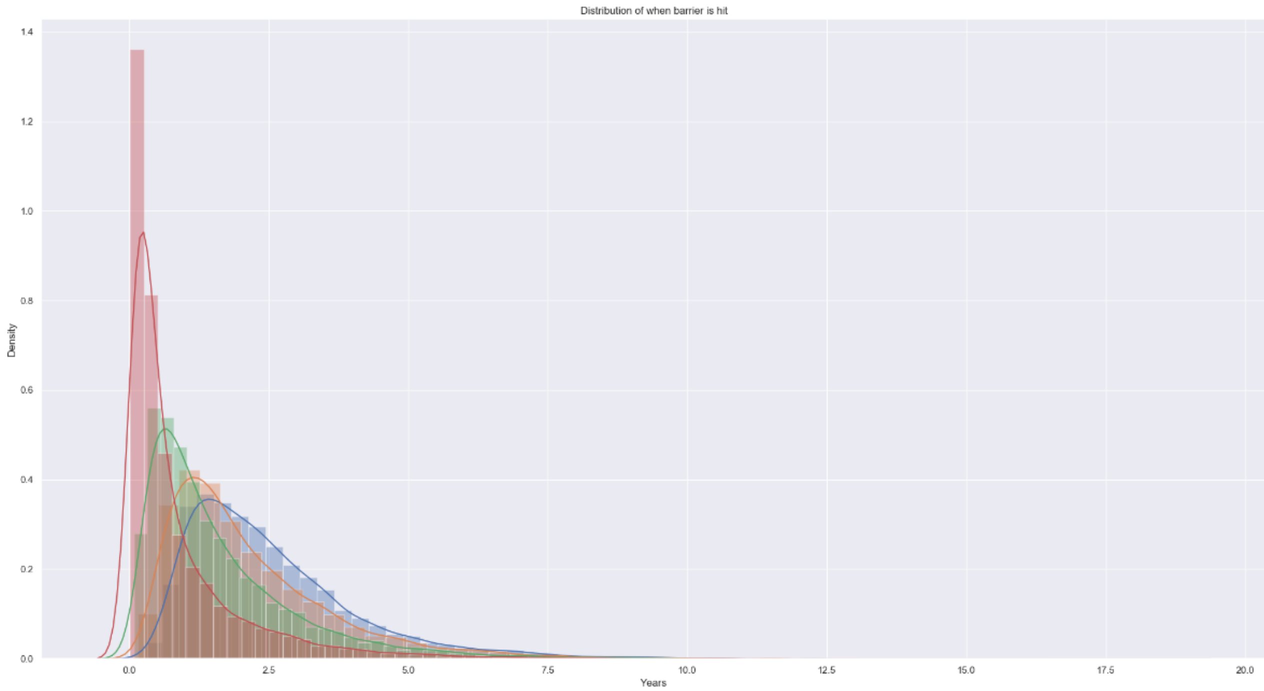
In which we see a clear relationship between starting short rate and the corresponding mean τ^*. The further away the starting short rate is from the boundary, the longer it takes for it to hit the barrier b*.

Below is seen the histogram of τ^* for r(0) = -0.01.



For the remaining plots for single short rates, see Appendix.

Below is seen every second distribution plotted together. This clearly displays the trend seen from the data above, of τ^* being larger, the further away the initial short rate is from the boundary:



7)

To use the random variable, $e^{-\int_0^{200} r(s)ds}$, as a control variable I will be using (48) and (49) from the slides MCmetoder, as seen below:

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(\hat{P}_A^{(i)} - \bar{P}_A)(\hat{P}_G^{(i)} - \bar{P}_G)}{\sum_{i=1}^{n}(\hat{P}_G^{(i)} - \bar{P}_G)^2}, \quad \bar{P}_A = \frac{1}{n}\sum_{i=1}^{n}\hat{P}_A^{(i)}, \quad \bar{P}_G = \frac{1}{n}\sum_{i=1}^{n}\hat{P}_G^{(i)} \tag{48}$$

Then we can form the CV estimator as

$$\bar{P}^{CV} = \bar{P}_A + \hat{\beta}(P_G - \bar{P}_G) \tag{49}$$

The resulting zero coupon prices from the method, are seen below, with their corresponding 95% confidence intervals:

r(0) = -0.05, price = 3.981862044193669, 95%-CI: [3.95948, 4.00424]

r(0) = -0.04, price = 3.1146897822281745, 95%-CI: [3.10818, 3.1212]

r(0) = -0.03, price = 2.4221145774186135, 95%-CI: [2.41767, 2.42656]

r(0) = -0.02, price = 1.927675145407769, 95%-CI: [1.92183, 1.93352]

r(0) = -0.01, price = 1.5505908323036406, 95%-CI: [1.54923, 1.55195]

r(0) = 0.00, price = 1.292949565318348, 95%-CI: [1.29242, 1.29348]

r(0) = 0.01, price = 1.1270868837461743, 95%-CI: [1.12766, 1.12652]

r(0) = 0.02, price = 1.0155431778257824, 95%-CI: [1.01542, 1. 01567]

The variance compared to the uncontrolled Monte Carlo sampling is, same r(0)'s, these are not corrected for number of paths:

Without control variable: 232.39513, with control variable: 114.19524, efficiency ratio: 2.03507

Without control variable: 60.23988, with control variable: 33.20257, efficiency ratio: 1.81431

Without control variable: 40.92133, with control variable: 22.67546, efficiency ratio: 1.80465

Without control variable: 186.91682, with control variable: 29.81267, efficiency ratio: 6.26971

Without control variable: 115.95971, with control variable: 6.94479, efficiency ratio: 16.6974

Without control variable: 126.25794, with control variable: 2.69264, efficiency ratio: 46.89

Without control variable: 4.51918, with control variable: 2.91125, efficiency ratio: 1.55232

Without control variable: 1.02677, with control variable: 0.63933, efficiency ratio: 1.60601

We see that the sampling with the control variable is consistently around 1.5-2 times more effective with respect to variance. There is also a trend of less variance with r(0) closer to the boundary. But the most interesting, is the fact that, even though they both spike in variance sometimes, as seen for r(0)=-0.02, where the normal variance is at 186.91682, the variance for the control spikes significantly less, which creates the high efficiency ratio.

The control variable makes the computational workload a bit heavier, so we want a high efficiency ratio before we choose to do this, if we are constrained by cpu-time. In this case, a more efficient sampling per unit of cpu-time is uncertain, but when it pays off, it does so big time.

8)

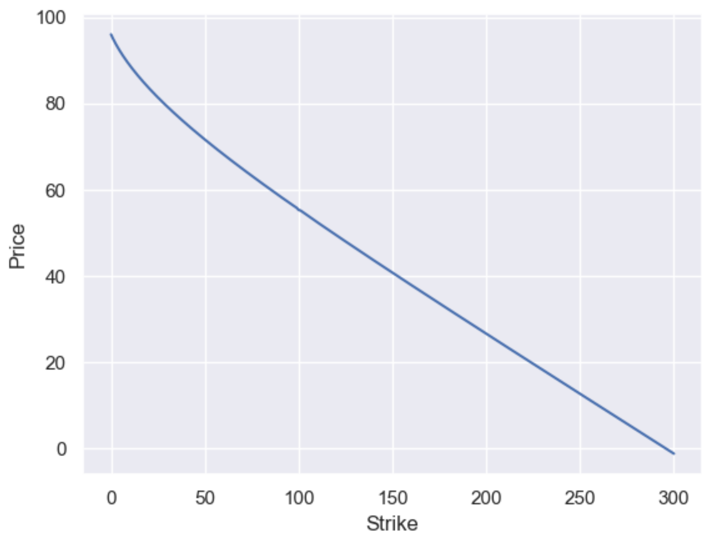
Skipped

Finite Difference Goes American

1)

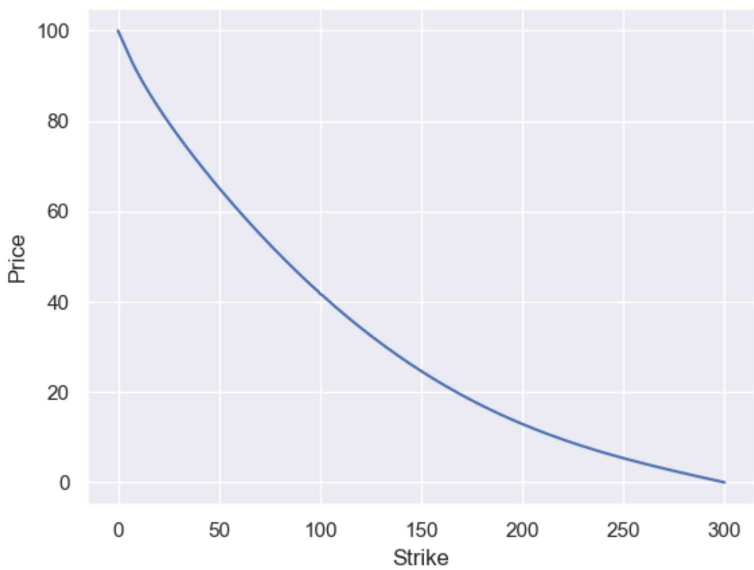
I have applied the Crank-Nicolson method to solve the tridiagonal system of equations, in order to price the European put. The implementation though, keeps the elasticity, α, under the CEV model, at 1, and not the 0.5 as described in the question text.

Plotting the price for strikes between 0 and 300 gives the figure below:



2)

To price the American put, I've applied the PSOR-method. Solving and plotting the price for the same range gives the following figure:



3)

The implied volatility is found by root solving using the Newton-Raphson method. Inserting the price of 45 and strike of 100, and solving for the European pricing, gives the implied volatility of:

```
price = 45, impVol = 1.3462495801108376
```

4)

In Proposition 4, in the paper, it is assumed that the nominal short rate follows a one-dimensional time-homogeneous diffusion process:

$$dr(t) = \alpha(r)dt + \sigma(r)dW(t),$$

The CEV model is a specific variation of the general process above. Replacing $\alpha(r)$ with $(r-\delta)S(t)$ and replacing $\sigma(r)$ with $\sigma S^{\alpha}(t)$ gives us our model:

$$dS(t) = (r-\delta)S(t)dt + \sigma S^{\alpha}(t)dW(t)$$

Naturally the exercise boundary differs between the 'Cash option' and that of an American Put/Call.

For the American Option, the PDE is given as:

$$\frac{dV}{dt} + a(x,t)\frac{dV}{dx} + \frac{1}{2}\sigma^2(x,t)\frac{d^2V}{dx^2} = 0$$

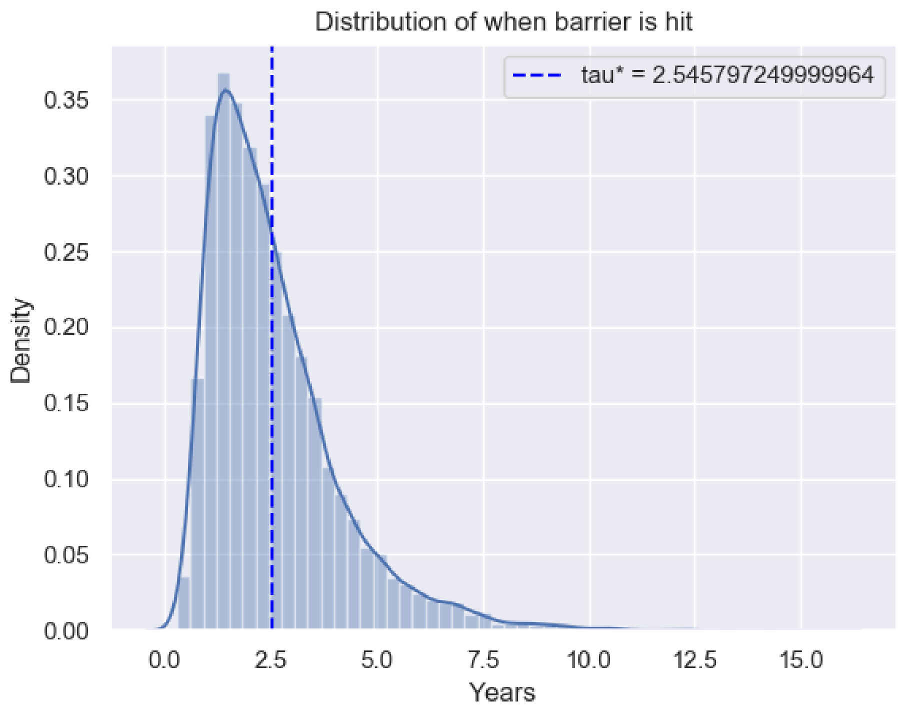
And the boundary condition: $V(T,s) = (\mathrm{S_T} - \mathrm{K})^+$ for the call.

One difference we notice between the American option and the ODE representation of the cash-option is the theta term. The cash option has an infinite time horizon and thus is considered a "perpetual option". Trivially, a perpetual or infinite-time-horizon option has no theta term as the option will not expire.

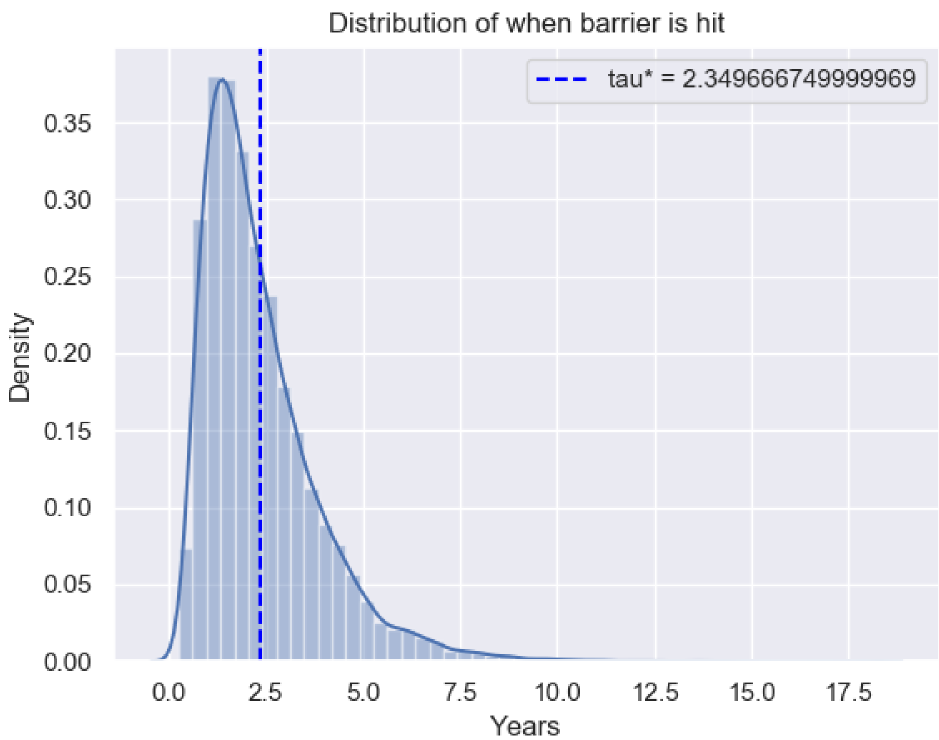
To solve either ODE/PDE, we can apply the same numerical solution methods – and in fact – the paper does indeed use finite-difference method as we do for our American option. Different drift and diffusion dynamics and different boundary conditions – but same approach.

Appendix

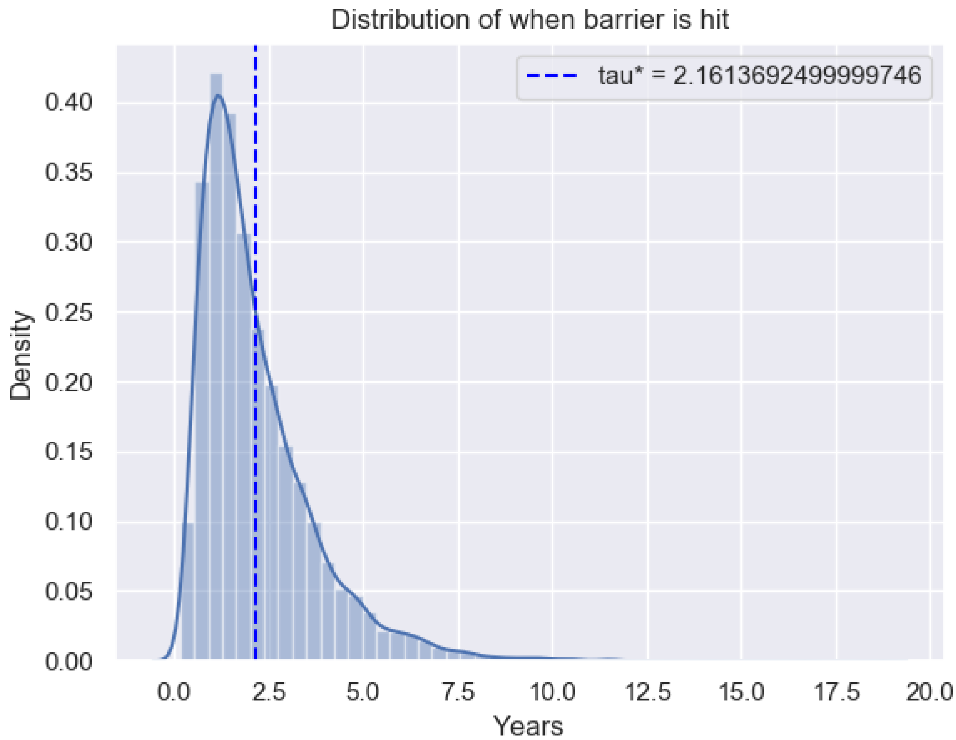
Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = -0.05:



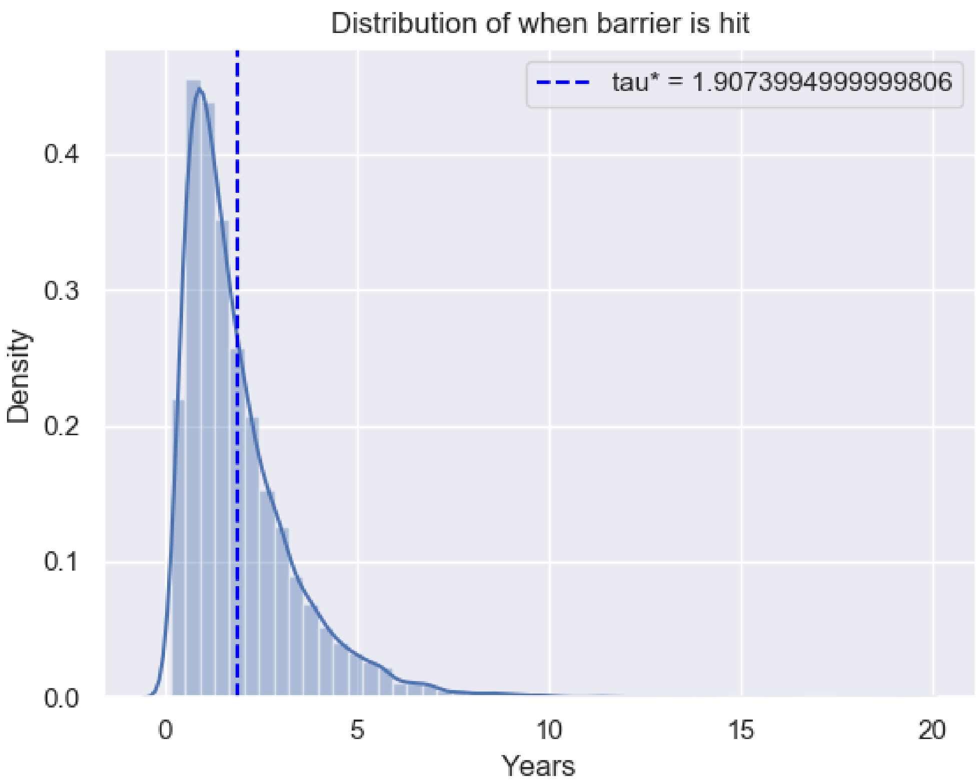
Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = -0.04:



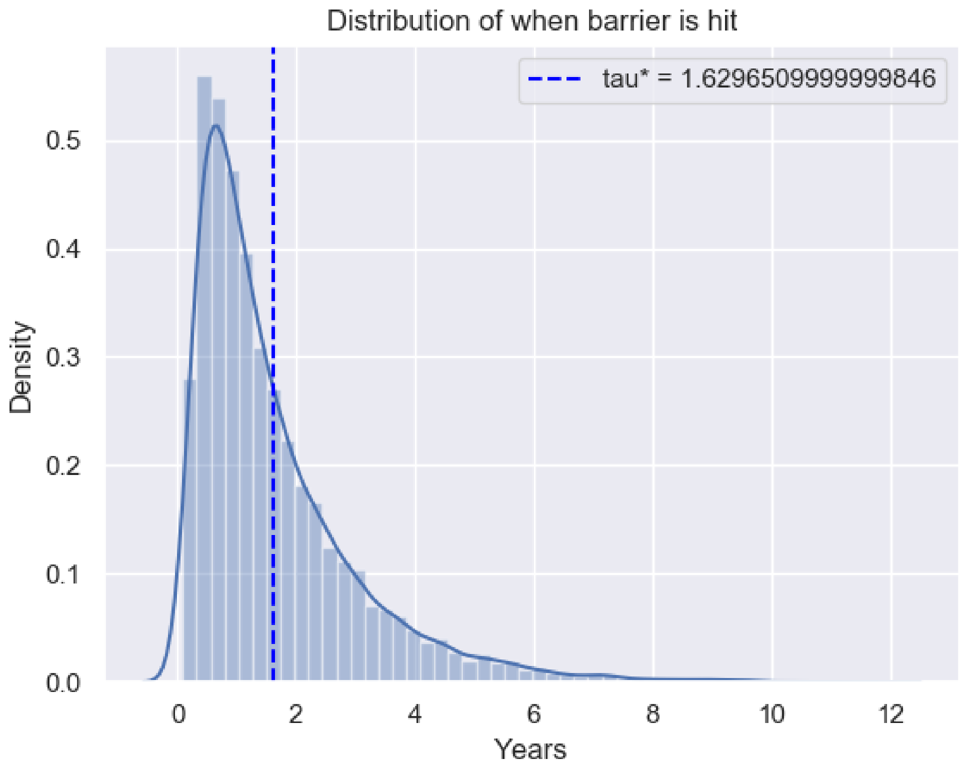
Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = -0.03:



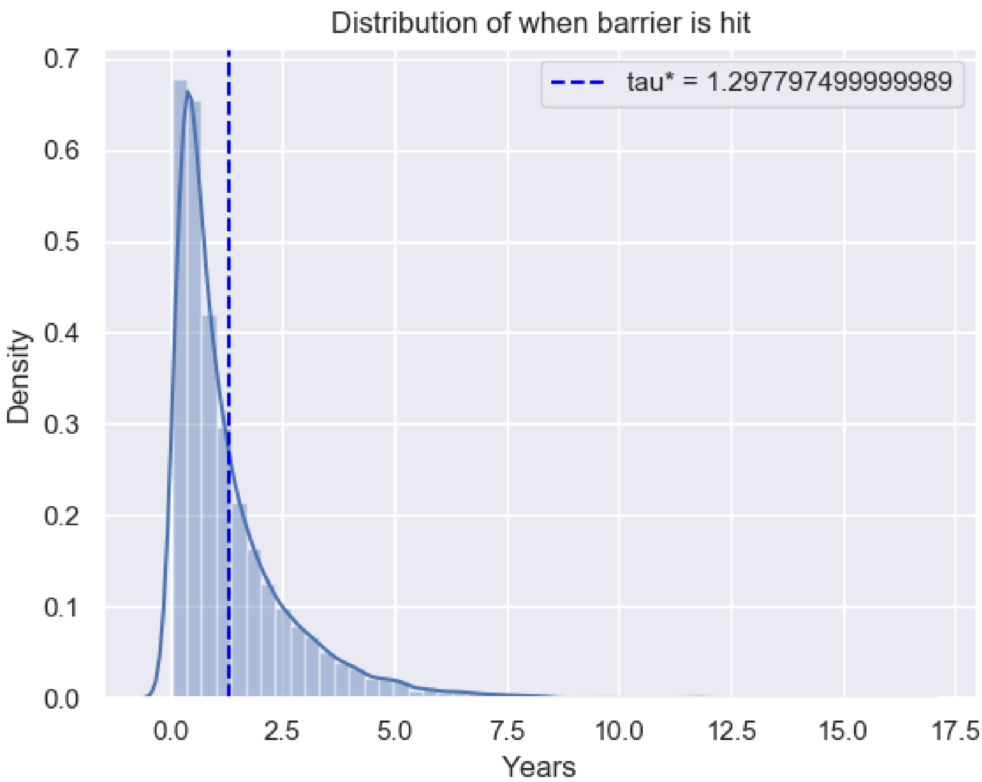
Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = -0.02:



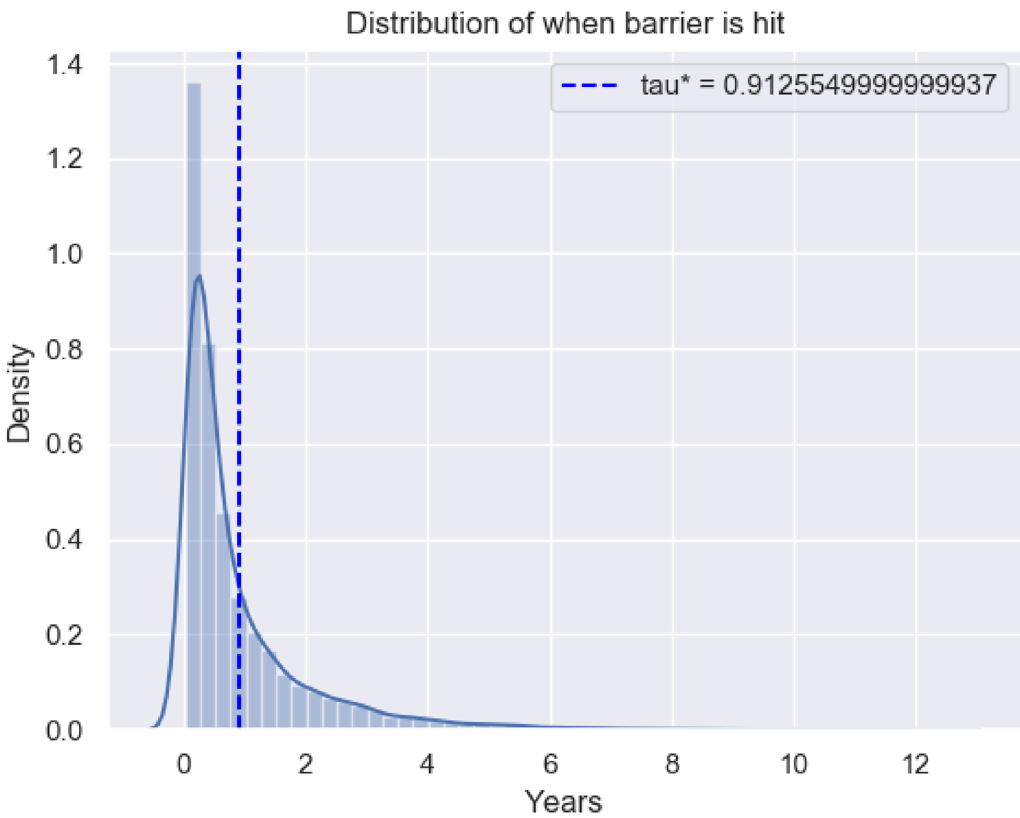
Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = -0.01:



Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = 0.0:



Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = 0.01:



Histogram for Monte Carlo simulations of time for short rate to break barrier b*. Here r(0) = 0.02: