Assignment

Option Pricing and Risk - Financial Statistics

Alexander J Ohrt

15 mai, 2022

Contents

Problem A - Euler-Maruyama Method

We implement a discretization for the Black-Scholes-Merton (BSM) price dynamics with the parameter set $(s_0, T, \mu, \sigma) = (12, 1, 0.03, 0.17)$, with $n = 250$ steps. We generate $M_1 = 10$, $M_2 = 100$, $M_3 = 1000$, $M_4 = 10000$ and $M_5 = 100000$ paths. The paths are plotted in separate figures for the three first cases, together with the mean value of the price process.

The price dynamics are implemented with two different methods for calculating each path. The R function appended has the parameter `version`, which is used to select which version one wants to use to calculate each price path; `onestep` or `EM`. The first method implemented is the one step solution. Since the Wiener process has the distribution $W_t \sim N(0,t), \forall t \in (0,t]$, and the variables $W_{t_1} - W_{t_0}, W_{t_2} - W_{t_1}, \ldots, W_{t_n} - W_{t_{n-1}}$ are independent, we can think of such a process as a cumulative sum of normally distributed random variables. Thus, on our uniform discrete grid

$$0 = t_0 < \ldots < t_n = T,$$

we can simulate the following relation

$$\begin{aligned}
&X(t_0) = 0, \\
&X(t_1) \sim N(0,t_1) = N(0,t_0+h) \overset{t_0=0}{=} N(0,h), \\
&X(t_2) \sim N(0,t_2) = N(0,t_0+2h) \overset{t_0=0}{=} N(0,2h) \overset{\text{Indep.}}{=} N(0,h) + N(0,h) = 2X(t_1), \\
&\qquad \vdots \\
&X(t_n) \sim N(0,t_n) = N(0,t_0+nh) = nX(t_1),
\end{aligned}$$

where X represents the Wiener process, n is the number of steps in the discretization and $h = \frac{T}{n}$ is the step size. Thus, we simulate $n+1$ $N(0,h)$ variables and use their cumulative sum, at each grid point, as a random draw from the Brownian motion. We insert this, replacing W_t, into the solution of the SDE

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t,$$

which is

$$S_t = S_0 e^{\mu t} e^{\sigma W_t - \frac{\sigma^2}{2} t},$$

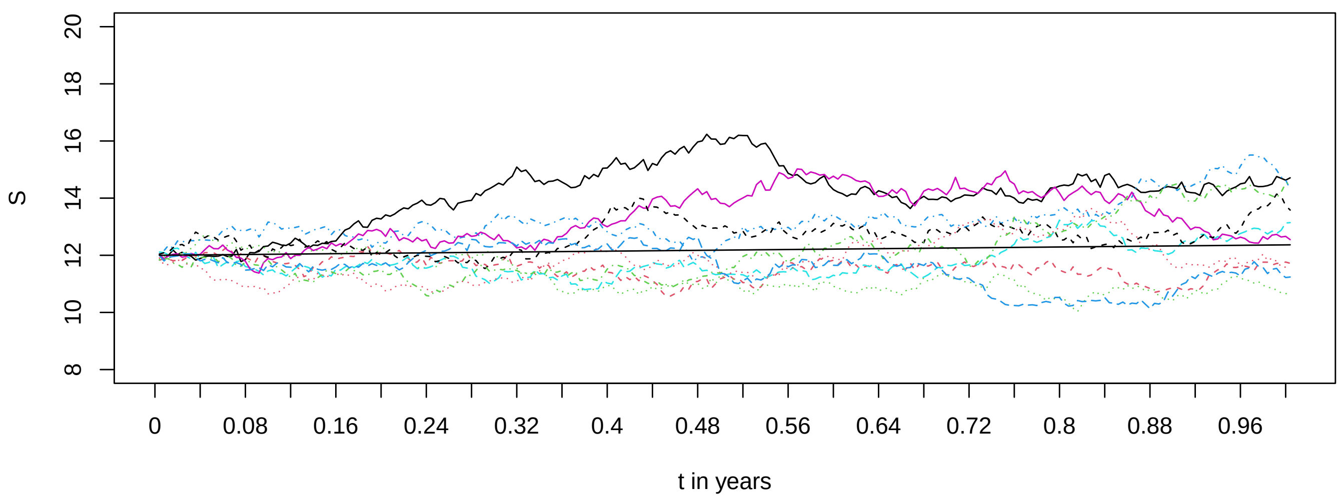
which makes it possible to calculate the one step solution on the grid. The second method implemented is the Euler-Maruyama (EM) scheme

$$\hat{S}_{t_{j+1}} = \hat{S}_{t_j} + \mu h \hat{S}_{t_j} + \sigma \sqrt{h} Z_{t_j} \hat{S}_{t_j},$$

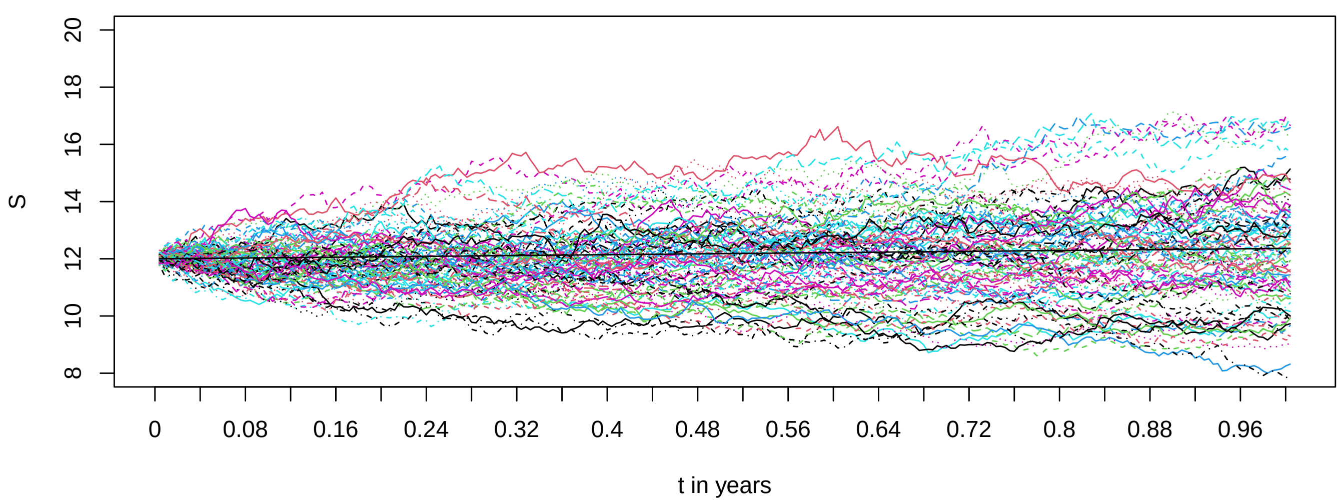
where Z_{t_j} are standard normally distributed variables.

Notice that the two implementations give slightly different price paths and slightly different estimations. Keep in mind that the results from the `onestep` method are the ones shown and discussed during the remainder of the problem, even though the results using the `EM` method are highly comparable.

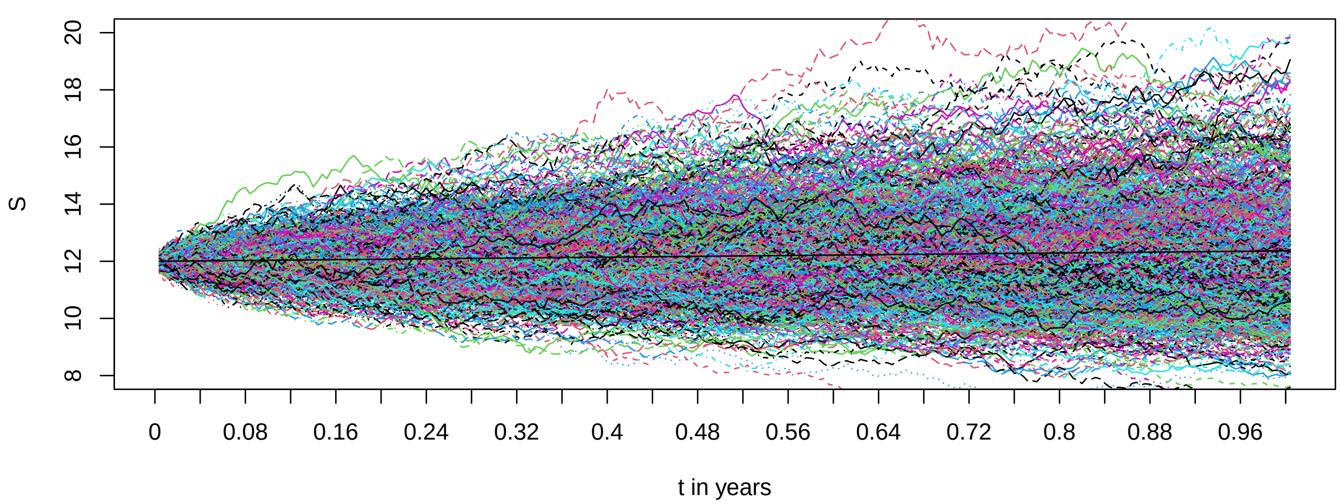
10 paths



100 paths



1000 paths



The Monte Carlo estimator for $\hat{S}_T$ is calculated separately for each of the values of M_i, $i \in \{1, 2, 3, 4, 5\}$. The 95% confidence interval (CI) is provided for each estimator. A comparison between these estimators and the analytical solution of $\mathbb{E}(S_T)$ is done and differences are explained.

The Monte Carlo estimator for $\hat{S}_T$ is simply calculated by averaging the values of all the different BSM price paths plotted above at time $T = 1$. Thus, in practice, we only need the last value of the price paths to calculate this estimator (and its standard error). The $(1-\alpha)\cdot 100\% = (1-0.05)\cdot 100\% = 95\%$ CIs are calculated by finding the standard error of the values of all the different BSM price paths at time T and using the approximation given by

$$CI_\alpha = \left(\hat{S}_T - z_{\alpha/2}\frac{se_{\hat{S}_T}}{\sqrt{M}}, \hat{S}_T + z_{\alpha/2}\frac{se_{\hat{S}_T}}{\sqrt{M}}\right),$$

where $se_{\hat{S}_T}$ is the aforementioned standard error, $z_{\alpha/2}$ is the $1-\alpha$ quantile of the standard normal distribution and M is the number of price paths simulated. This is an asymptotically valid $(1-\alpha)\cdot 100\%$ CI, which means it converges to the exact analytic value when M is increased.

Table 1: Monte Carlo Estimation for S_T, varying M

	M1 = 10	M2 = 100	M3 = 1000	M4 = 10000	M5 = 100000
Est.	12.79035	12.27904	12.41963	12.32897	12.36536
Lower CI	11.86917	11.90612	12.28574	12.28727	12.35226
Upper CI	13.71154	12.65195	12.55352	12.37067	12.37845

Now, what is the analytic solution of $\mathbb{E}(S_T)$? We know that the process of the risky asset in $t \in [0, T]$ is distributed as

$$S_t \sim \mathcal{L}N(\mu^*, \sigma^{*2}),$$

where $\mathbb{E}(S_t) = \mu^*$. In addition, we know that, when $W_t \sim N(0, t)$,

$$X_t \sim N(\ln S_0 - \left(\frac{\sigma^2}{2} - \mu\right)t, \sigma^2 t) = N(\mu_{X_t}, \sigma^2_{X_t}),$$

where $S_t = e^{X_t}$. Finally, we know that the expected value of the log-normally distributed variable S_t is $\exp\left(\mu_{X_t} + \frac{\sigma^2_{X_t}}{2}\right)$. This means that the analytic solution for $\mathbb{E}(S_T)$ is

$$\mu^*|_{t=T} = \exp\left(\mu_{X_t} + \frac{\sigma^2_{X_t}}{2}\right)\Bigg|_{t=T} = \exp\left(\ln S_0 - \left(\frac{\sigma^2}{2} - \mu\right)t + \frac{\sigma^2 t}{2}\right)\Bigg|_{t=T} = S_0 e^{\mu t}|_{t=T} = S_0 e^{\mu T},$$

which in this case has the numerical value

```
#> [1] 12.36545
```

From the results above we can clearly see that the MC estimations move closer and closer to the analytic solution when the number of paths M is increased. For M_5 the estimation is precise to the first three decimals (depending on which `version` one uses; this is true for `onestep`), which I would regard as a very good estimation. We also see that the CI's clearly change with the increase of M; the lower value of the CI's increase with M and the upper value of the CI's decrease with M, meaning that the 95% CI's become narrower when the number of paths increase. This is what we expect from the MC theory, since the estimators are unbiased and, as stated by the strong Law of Large Numbers, the sample average converges a.s. to the true expected value.

Now we fix the number of paths $M^* = 1000$ and vary the values of n, i.e. the number of steps, while repeating the discussion done above.

Table 2: Monte Carlo Estimation for S_T, varying n

	n1 = 12	n2 = 24	n3 = 250	n4 = 1000
Est.	12.38563	12.27697	12.43257	12.40517
Lower CI	12.24430	12.15002	12.30242	12.27156
Upper CI	12.52696	12.40391	12.56272	12.53879

We can make several observations in this case. First of all, the estimations are hovering around the true mean value of the price process. Secondly, the increase in precision is not as regular as in table 1. For example, the point estimate yielded from $n_1 = 12$ is the closest to the analytic solution (noting that the CI's are large), which is unexpected, since this is the least amount of steps used. Moreover, when increasing n, the lower limit of the CI decreases from n_1 to n_2, which is not as expected, considering the behaviour we saw in table 1. The same happens from n_3 to n_4. We also see an increase in the upper limit of the CI when moving from n_1 (or n_2) to n_3. What are the possible explanations? Of course, these results are (pseudo-) random, so changing the seed will change the results. Thus, a different seed might lead to CI's that look more regular or closer to what we expect. However, what this signals is that the number of paths M is too low. Moreover, this also leads to the conclusion that the fineness of the grid is important to a certain extent, but if M is too small, we are not able to improve the estimations simply by calculating each price path on a finer grid. Thus it is the number of paths M, and not the number of discretization points n, that yields dramatic differences when its value is increased. In other words, the estimates hover around the true value when n is increased while M is fixed, but the changes seem to be more dramatic when n is fixed and M is increased. It is however important to notice that, in our experiment, n is only changed across three orders of magnitude, while M is changed across five orders of magnitude, which might lead to a somewhat biased discussion. All in all, the conclusion is that a large value of M is more important compared to a very large value of n (to a certain extent), which is in accordance with the MC theory, as mentioned earlier.

Problem B - Option Pricing I

We calculate the price of a Call option with parameter set $(s_0, T, r, \sigma, K) = (24, 1.5, 0.02, 0.22, 23.5)$, using the Black-Scholes-Merton (BSM) formula.

```
#> [1] 3.149899
```

The price is approximately equal to 3.15, when rounded to 2 decimals.

We implement a Monte Carlo estimator for the price of this option by simulating paths with the Euler-Maruyama method for steps $n = 10, 100, 1000$ and paths $M = 10, 100, 1000, 10000, 100000$.

Notice that for the path-independent options, like standard European Call and Put options, it is not necessary to save the price path as is done in my implementation. This is done to keep the function as general as possible, in order to re-use it for the rest of the assignment. To increase computational efficiency and decrease the use of memory one could simply iteratively overwrite one variable, instead of saving the entire price path history.

Assuming that $\mu = r$, we use the MC estimator to calculate the price of the option.

Table 3: Relative Errors

	n = 10	n = 100	n = 1000
M = 10	-1.0924571	0.4669412	0.5786881
M = 100	0.1971622	-0.0796902	-0.2981574
M = 1000	-0.0274607	0.0206036	0.0394080
M = 10000	-0.0193889	0.0442892	0.0199679
M = 100000	-0.0033752	-0.0016929	0.0024548

Table 4: Absolute Errors

	n = 10	n = 100	n = 1000
M = 10	3.4411299	1.4708178	1.8228094
M = 100	0.6210411	0.2510161	0.9391658
M = 1000	0.0864983	0.0648991	0.1241311
M = 10000	0.0610730	0.1395064	0.0628968
M = 100000	0.0106317	0.0053325	0.0077324

How can these results be interpreted in view of n and M?

First of all, we can clearly see that, in the majority of cases, the absolute values of both types of errors decrease when the number of paths M increases. This is coherent with what we expect, as noted earlier, based on the MC estimator. The best value obtained for the relative error is ≈ -0.0017, meaning that we are able to get within 0.2% of the closed form solution, which is very accurate.

Secondly, we notice that it is the change in M that generally leads to more dramatic changes (often of an order of magnitude) in the (absolute values of the) errors, as already noted in Problem A. Thus, from the errors seen in these tables, we would conclude the same as in Problem A regarding the level of importance of the values M and n. Increasing M (moving south in the tables), while n is fixed generally decreases the error, whereas this does not happen as frequently when increasing n (moving east in the tables), while M is fixed.

Additionally, we notice that the lowest errors tend to be found in the lower right corner of the tables, which is as expected, since this is the area of the table with the finest discretized grids (large n) and more paths (large M).

Moreover, notice that all absolute values are smaller in table 3 (relative errors) compared to their respective value in table 4 (absolute errors), since the relative errors are calculated in almost the same way as the absolute errors, where the only difference is not taking absolute values and dividing by the BSM price (≈ 3.15). Notice that if the absolute value is added in the calculation of the relative errors, the values are all the same, with the seven negative signs disappearing. I cannot find a clear pattern in the negative signs that appear in the relative errors, which means that it is not clear that some specific combinations of n and M overestimate the BSM price.

The results that are yielded for values of M smaller than 1000 are useless, since they are very bad estimations far from the BSM price. When setting $M = 1000$ we are getting into the 5% domain of relative errors, which is further improved when M is further increased. Thus, we see that we need a (relatively) substantial amount of paths in order for the MC estimator to yield useful estimations - it is not enough with 10 or 100 paths.

Problem C - Option Pricing II

A Monte Carlo estimator is implemented for an Asian Call Option. This Asian Call Option averages prices every Friday. We set $n = 252$ and assume that $n = 1$ is Monday. This means that we average the prices $t_5, t_{10}, t_{15}, \ldots$. The payoff profile for this option is thus $(\bar{S} - K)^+$, where $\bar{S}$ is the arithmetic average over all the prices $t_i,\ i \in \{5, 10, 15, \ldots\}$. We set $M = 10000$ and use the parameter set $(s_0, T, r, \sigma, K) = (20, 1, 0.02, 0.24, 20)$.

```
#> [1] 1.186436
```

As we can see from the result above, the price estimated via the MC estimator is 1.1864.

A Monte Carlo estimator is implemented for a Lookback option with payoff profile $(S_{max} - K)^+$, where S_{max} refers to the maximum price during the time to maturity of the option. We use the parameter set $(s_0, T, r, \sigma, K) = (22, 2, 0.02, 0.29, 21)$.

We calculate estimations and 95% confidence intervals of this option price for $M_1 = 1000$, $M_2 = 10000$ and $M_3 = 100000$ paths.

Table 5: Monte Carlo Estimation for Lookback Option, Varying M

	M1 = 1000	M2 = 10000	M3 = 100000
Est.	9.402367	9.044822	9.177744
Lower CI	8.911186	8.892006	9.128017
Upper CI	9.893549	9.197637	9.227471

Similar observations as in earlier problems can be made concerning how these CI's change with increasing M; they become narrower with increasing number of paths, which is the behavior we expect. Notice however that in some of the cases, the lower CI limit decreases with increasing M and in some of the cases the upper CI limit increases with increasing M. If I was to choose the fair price of this Lookback option to one decimal place, perhaps to use as a first estimate in a practical application, I would set it to 9.2, since this is within the CI of the price estimated with the largest M. The 95% CI's can be interpreted as; out of all the 95% CI's calculated from this "experiment" or procedure, 95% of them should contain the true fair price of the option.