Assignment

Option Pricing and Risk - Financial Statistics

Alexander J Ohrt

03 mai, 2022

Contents

Problem A - Euler-Maruyama Method

First 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.

The price dynamics are implemented using the function shown below, which is used to calculate each path. Notice that, since the Wiener process has the distribution $W_t \sim N(0, t), \forall t \in (0, t]$, we can think of such a process as a cumulative summation of normally distributed random variables. Thus, on our uniform discrete grid

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

we can simulate the following relation

$$X(t_0) = 0,\ X(t_1) \sim N(0, t_1) = N(0, t_0 + h), \ldots,\ X(t_n) \sim N(t_0 + nh) = X(t_1) + \ldots + X(t_{n-1}),$$

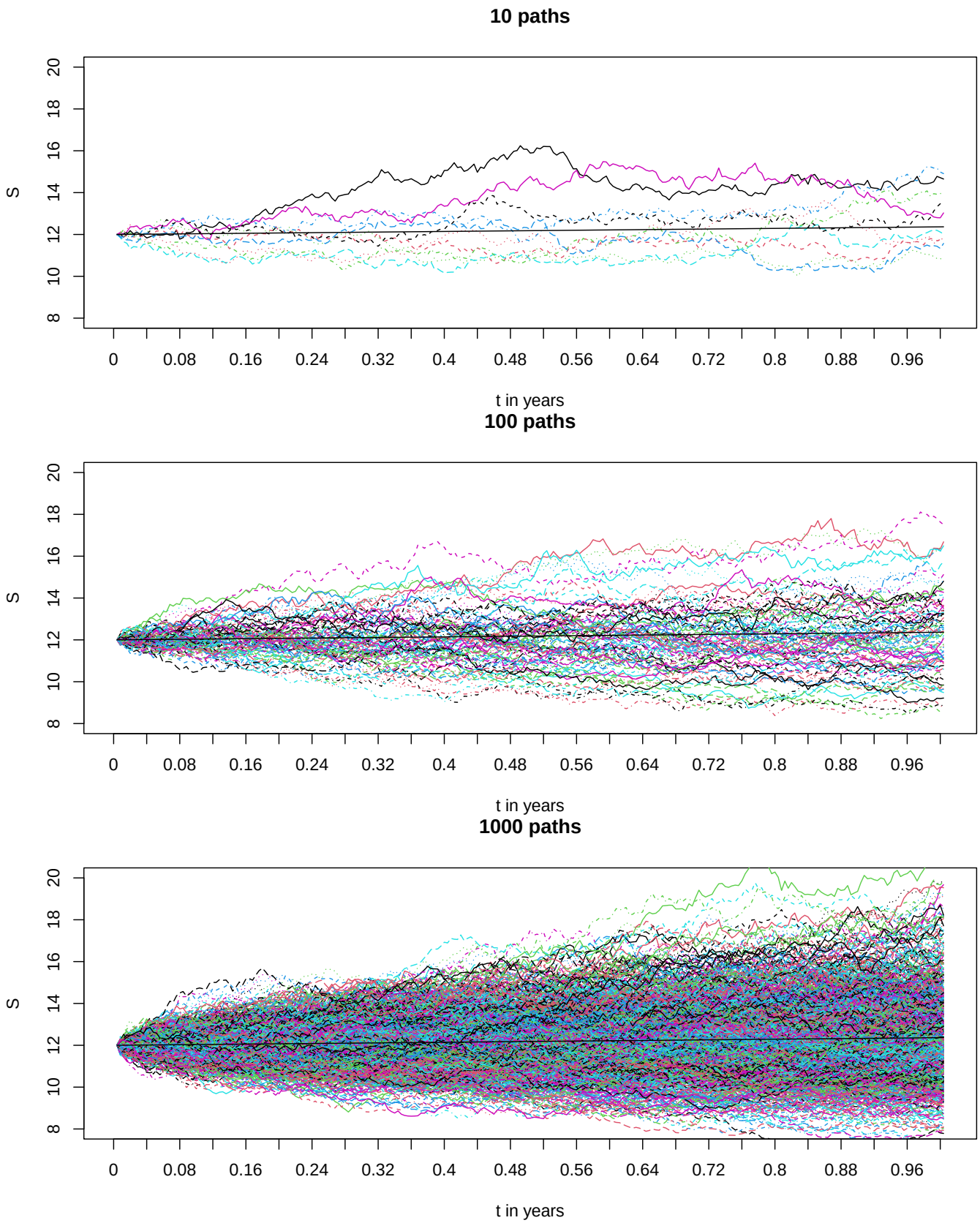
where X represents the Brownian motion, n is the number of steps in the discretization and $h = \frac{T}{n}$ is the step size.

VIL HAN HA EULER-MARUYAMA HER OG IKKE DENNE FORMELEN? DET ER ENKELT Å BYTTE UT, MEN MÅ SPØRRE (FOR EM ER IMPLEMENTERT LENGER NEDE UANSETT!)

```
# Price path stochastic process.
price.path <- function(s0, t, mu, sigma){
  #Wt <- rnorm(n = length(t), sd = sqrt(t[2]-t[1])) # Draw length(t) normally distributed
  #variables with mean 0 and standard deviation h.
  #W2 <- cumsum(Wt) # Step size is constant, grid is uniform. We calculate the cumulative
  #sum of the standard normal draw to simulate the Wiener process, which is N(0,t) at
  #time t, i.e. the variance increases when the process is run further and further away
  #from t = 0. Notice that we also multiply by the (uniform) stepsize used on the
  #(uniform) grid, in order to transform the N(0,1) values to N(0,t).
  #s0*exp(mu*t)*exp(sigma*W2-sigma^2/2*t)

  # Byttet ut her i tilfelle det var det han ville!
  # t-aksen trengs ikke i dette tilfellet da! Bedre å bare bruke n!
  n <- length(t)-1
  Z <- rnorm(n)
  S.values <- rep(NA, length.out = n+1)
  S.values[1] <- s0
  h <- t[2]-t[1]
  for (j in 2:(n+1)){
    S.values[j] <- S.values[j-1] + mu*S.values[j-1]*h + sigma*S.values[j-1]*sqrt(h)*Z[j-1]
  }
  S.values
}

# VIL HAN EGENTLIG HA EULER-MARUYAMA HER TIL Å LAGE HVER PATH I STEDET!? ELLER ER DETTE OK?
```



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 $(1-\alpha)\% = (1-0.05)\% = 95\%$ 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 $1-\alpha$ CI, which means it becomes closer 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.	12.80569	12.26352	12.42701	12.32369	12.36488
Lower CI	11.92015	11.90351	12.29141	12.28267	12.35180
Upper CI	13.69123	12.62354	12.56260	12.36471	12.37795

Now, what is the analytical solution for $\mathbb{E}(S_T)$? 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 $W_t \sim N(0, t)$, it follows that

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

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 analytical solution for $\mathbb{E}(S_T)$ is

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

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 analytical solution when the number of paths M is increased. For M_5 the estimation is precise to the first 3 decimals, 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.	12.28459	12.35087	12.33414	12.43521
Lower CI	12.15974	12.22149	12.20175	12.30558
Upper CI	12.40945	12.48025	12.46654	12.56485

We can make several similar observations in this case;

Notice however that is is the number of paths M, and not the number of discretization points n, that yields dramatic differences when the value is increased. In other words, the estimates move closer to 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 observation or discussion.

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 needed 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 lessen 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	0.5821167	0.5566707	0.6125545
M = 100	0.0255317	0.0021910	0.3364722
M = 1000	0.0522963	0.0273627	0.0263294
M = 10000	0.0066211	0.0508549	0.0291815
M = 100000	0.0035652	0.0019417	0.0029947

Table 4: Absolute Errors

	n = 10	n = 100	n = 1000
M = 10	1.8336091	1.7534568	1.9294852
M = 100	0.0804222	0.0069013	1.0598536
M = 1000	0.1647280	0.0861897	0.0829349
M = 10000	0.0208557	0.1601878	0.0919187
M = 100000	0.0112299	0.0061162	0.0094331

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 the errors decreases 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 ≈ -0.0019, 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 leads to dramatic changes (often of an order of magnitude) in the errors, as already noted in Problem A, where a change in n does not lead to the same dramatic changes BE SURE THAT THIS MAKES SENSE!

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).

WHAT ELSE CAN I COMMENT ON HERE?!

Moreover, notice that all 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 five 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 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 gain 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] 1.174624
```

As we can see from the result above, the price estimated via the MC estimator is 1.1746. THIS PRICE CAN PERHAPS BE CHECKED WITH A CALCULATOR ONLINE!

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 $(s_0, T, r, \sigma, K) = (22, 2, 0.02, 0.29, 20)$ THIS K IS NOT SPECIFIED IN THE PROBLEM DESCRIPTION, BUT I WILL JUST LEAVE IT HERE NOW!

```
#> [1] 2.367147e-18
```

As we can see from the result above, the price estimated via the MC estimator is 0. THIS PRICE CAN PERHAPS BE CHECKED WITH A CALCULATOR ONLINE!

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