



Stockholms
universitet

Antithetic variates in Asian options pricing: a numerical investigation

Emilio Dermani

Masteruppsats 2015:9
Matematisk statistik
Oktober 2015

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm



Mathematical Statistics
Stockholm University
Master Thesis **2015:9**
<http://www.math.su.se>

Antithetic variates in Asian options pricing: a numerical investigation

Emilio Dermani*

October 2015

Abstract

Asian options are a type of financial derivative constructed for assets with low liquidity (that are thus susceptible to price manipulation), such as crude oil. Depending on the averaging type they use in the payoff function, they may or may not have closed-form solutions for their prices. In this thesis, I examine the latter case (Monte Carlo simulation – MC) and a technique which reduces variance (Antithetic Variates – AV). I find that the use of the variance reduction is not justified by the stock distribution created by the two approaches, but rather by descriptive statistics of the two samples. Finally, I examine the rate of convergence for AV and pinpoint when its use is more effective than MC.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden.
E-mail: emilio.dermani@gmail.com. Supervisor: Joanna Tyrcha.

“We make ourselves significant by the courage of our questions, and the depth of our answers.”

Carl Sagan

Acknowledgements

I would like to thank my project adviser, *Joanna Tyrcha*, for constantly encouraging me and providing me with the liberty to write this thesis on my own terms, *Robert Fitzner* for suggesting the third method of testing in Chapter 2, and *Filip Lindskog* for rekindling my passion in Mathematical Statistics and for showing me that my true calling lies in Quantitative Risk Management.

Special thanks go to professor and colleague *Bo Larsson*, for providing me not only with the idea for this thesis, but also with food for thought and countless discussions for all academic matters, both within and beyond my reach.

Last but not least, I would like to express my appreciation to family and friends, for all the support you have given me since undertaking the ever-lasting journey of education.

Contents

Abstract	ii
Acknowledgements	iii
Contents	iv
1 The object, the problem, and the method	1
1.1 A brief description of options	1
1.2 The Asian option in detail	2
1.2.1 Status quo	3
1.2.2 The Law of Large Numbers	3
1.3 Pricing the option	4
1.3.1 The Monte Carlo method	5
1.3.2 The antithetic variates technique	6
1.4 The methodology ahead	7
2 Delineation of convergence	10
2.1 Statistical methodology	10
2.2 Anderson-Darling test	11
2.2.1 Description	11
2.2.2 Results	12
2.3 An unexpected brick wall	12
2.4 Epps-Singleton test	18
2.4.1 Description	18
2.4.2 Results	20
2.5 A numerical method	21
2.5.1 Description	21
2.5.2 Results	22
2.6 The final threshold	22
3 Implementing antithetic variates	23
4 Conclusions	28
A MATLAB code & further notes	30
Bibliography	38

*For Jon Snow, Stannis Baratheon, and all who defied
insurmountable odds. . .*

Chapter 1

The object, the problem, and the method

1.1 A brief description of options

Options have existed in financial transactions since ancient days. According to sources, the first mention of option buying is credited to Thales of Miletus in ancient Greece, and options have continued to exist ever since. However, with the establishment of the *Chicago Board Options Exchange* in 1973, the idea of options was reinvented for the finance world in a more detailed and standardized manner, so that everyone could deal with these instruments on the same basis. Ever since, options have continued to grow in both complexity and usage from investors and traders, since options are highly versatile financial instruments which can be used in a plethora of ways. Traders, being typically interested in maximizing profit, use options to speculate – a relatively risky practice; on the other hand, hedgers, who are focused on risk reduction, use options to reduce the inherent risk of holding an asset. This has incentivized the academia to the point where finding a "fair" price for an option, i.e. a price that does not allow for risk-free profit, has given birth to a special branch of Applied Mathematics dealing with this problem.

Options are most commonly divided in two large categories: vanilla and exotic. As indicated by their labels, vanilla options (namely American and European) are fairly "simple" in their description – though not necessarily as simple in their calculations – whereas exotic options offer more of a challenge, since they have more complicated formulations.

The pricing of one such exotic option will be the topic of this thesis. We will see how the "correct" price of the *Arithmetic Asian call option* can be found, what are the problems it poses, and how one can get around those problems.

1.2 The Asian option in detail

There are numerous permutations of an Asian option, but as mentioned above, we will work with the *call* version for the sake of simplicity, but the results can of course be extended by changing the payoff function in the algorithms used later.

Formally, an Asian call underwritten on asset S at $t = 0$, with time to maturity T and a fixed strike price K has payoff

$$C(T) = \max\{\bar{S}_{[0,T]} - K; 0\},$$

where $\bar{S}_{[0,T]}$ denotes the average of the asset (most commonly a stock) price over the time interval $[0, T]$. Here, we see immediately what separates an Asian option from vanilla options: the payoff function is dependent on the *average* of the stock price over the time period and not only on the final price S_T . This constitutes this type of option as *path-dependent*, i.e. we are interested in the entire course the stock has taken, not only an individual price as it is with European and American options.

Regarding the average $\bar{S}_{[0,T]}$, there are different ways of obtaining it. Typically, the choice is an *arithmetic average*. In the continuous case, this means that

$$\bar{S}_{[0,T]} = \frac{1}{T} \int_0^T S(t) dt.$$

However, since we are working in discrete monitoring (where the monitoring times are the subperiods $\frac{kT}{n}$, $k = 1, \dots, n$), the above translates to

$$\bar{S}_{[0,T]} = \frac{1}{N} \sum_{k=1}^N S(t_k).$$

Here, N signifies the total number of partition points of the $[0, T]$ interval. For reasons of simplicity but without any loss of generality, we choose $T = 1$. Needless to say, the efficiency and precision of the binomial tree (presented below) depends on an analogous relationship between N and T : the longer the time to maturity of the option, the more partitions are required to sufficiently path out the asset's trajectory.

In addition to the arithmetic, another case is that of the geometric average:

$$\bar{S}_{[0,T]} = \exp\left(\frac{1}{T} \int_0^T \ln(S(t)) dt\right).$$

One major difference between these two ways is that geometric averaging options can in fact be priced with a closed form (analytic) solution. This happens because, just as the underlying price process itself, its geometric average follows a lognormal distribution, a condition which no longer holds under arithmetic averaging. This result was presented in [1]. Because of these findings, this thesis focuses on pricing Asian options where the averaging is done arithmetically, since we are using Monte Carlo simulations and it still remains an issue with activity in the research area. Hence I will omit onward the word "arithmetic".

1.2.1 Status quo

As mentioned above, due to its nature, there is no closed-form solution for the arbitrage-free price of an Asian option. There have been developed various methods to this problem, such as the Effective Classical potential ([8]), or the PDE approach ([11]). The method we will focus on in this thesis however is the Monte Carlo simulations method. Simply put, the Monte Carlo method consists of simulating as many price paths of the stock as possible, in order to make use of the Law of Large Numbers (LLN) which states that, given enough ¹ simulations, we will get a reasonable price for our option.

1.2.2 The Law of Large Numbers

Without getting into too much detail, I present here the theorem that makes possible the pricing of an Asian option through Monte Carlo simulations. The (Weak) Law of Large Numbers states that the sample average of a random variable X converges (in probability) to the "true" mean:

$$\bar{X}_N \xrightarrow[N \rightarrow \infty]{p} \mathbb{E}[X] = \mu.$$

Translating this into our problem's terms, we have that if we perform N simulations (with N is sufficiently large), then according to the above, we will have

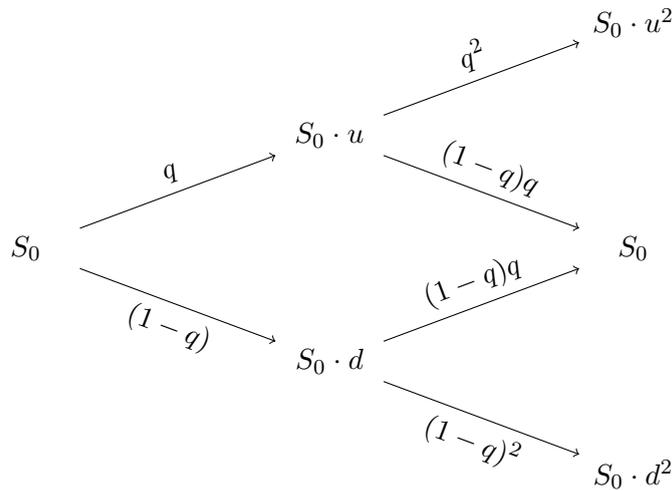
$$\bar{S}_N = \frac{1}{N} \sum_{i=1}^N S_i \xrightarrow[N \rightarrow \infty]{} \mathbb{E}[S].$$

¹Since the LLN provides the convergence as a limit to infinity, we have to find a finite number of simulations that still does the job within a sufficiently small numerical margin convergence for our price.

After this step, the price of the option is simply calculating $\max\{\bar{S}_N - K; 0\}$.

1.3 Pricing the option

In order to create the necessary simulations for the method described above, I will use the Cox-Ross-Rubinstein (CRR) method of the stock evolution over time, also known as the binomial model. This method makes use of certain characteristics (initial price of the stock at $t = 0$, time to maturity T , number of periods n , risk-free rate r , stock volatility σ) of a stock share in order to create a tree of possible price nodes that the stock can take. An illustration of a two-period tree is presented below:



where $u = e^{\sigma\sqrt{T/n}}$, $d = e^{-\sigma\sqrt{T/n}} = u^{-1}$, and $q = \frac{e^{rT} - d}{u - d}$.

As we can see, the tree is recombining, meaning that an "up" and then "down" movement leads to the same result as a "down" and then "up" one. Clearly, the tree will change based on the values that we give the parameters, in particular that of duration T and number of periods n . Since this tree will provide us with the stock prices required to calculate the option (and, in a sense, models the "reality" of the stock), we want to make it as detailed as possible while at the same time keeping it as small as possible. In other words, we want to have as many time periods n as possible – which will give us more possible final values for S_T at the end of the tree – but we also want to keep it small enough so that the computer does not waste too much memory on the calculation of the tree, so as to prioritize focus on the number of simulations N we will run.

1.3.1 The Monte Carlo method

The Monte Carlo (MC) method for pricing an Asian option is directly linked to the LLN. Using the CRR tree as the set of all possible events, the MC method consists of the following steps:

1. Define the jump factors u and d , the risk-free rate r , and the risk-neutral probability of an upward jump $q = \mathbb{Q}[S_{i+1} = S_i \cdot u]$, $i = 1, \dots, n - 1$.
2. Perform the desired amount of simulations.
3. Average the results accordingly so as to get the final price of the option.

Note that in this method, one does not actually have to construct the entire tree itself, which in fact we do not need. The u & d factors, along with q , ensure that the stock can only take prices from the tree that is created by said factors. The tree was only mentioned in order to help visualize the problem, since each simulation is in fact one path (out of all the possible outcomes) on the tree.

Let's say (for now) that we also chose our n (for example $n = 250$) and now we have a model for the stock evolution over the next year. We can then use the tree to simulate possible paths of the stock in the following way: at every time point $t = 1, \dots, n$, the price of the stock can be described as $S_t = S_{t-1} \cdot [Z_t u + (1 - Z_t) d]$, where $Z_i, i = 1, \dots, n$ are *iid*¹ *Bernoulli* distributed with probability of success q as calculated above. In other words, each simulation is fully characterized by the triplet $\{S_0, \mathbf{Z}, \{u, d\}\}$, where $\mathbf{Z} = (Z_1, \dots, Z_n)^T$ is the vector of the up-or-down jumps that decide how the stock price will move, and $u = u(r, \sigma, n), d = d(r, \sigma, n)$ are the jump factors, indicated here that they are also a function of the number n of time periods we have chosen for our model. Clearly, if one wishes to get one simulation of the final price S_T (and therefore also of the path that lead to that price), one needs a vector of n total Z_i 's, whereas if we want to do N simulations, one needs to simulate a matrix of $n \times N$; each column corresponds to one unique simulation path:

$$\mathbf{Z} = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \dots & Z_{1,T} \\ Z_{2,1} & Z_{2,2} & \dots & Z_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{n,1} & Z_{n,2} & \dots & Z_{n,N} \end{bmatrix},$$

which is a matrix of ones and zeros. In conclusion, n decides the "length" (or steps) of each simulation, whereas N decides how many simulations we get.

¹Independent and identically distributed.

For each simulation k , we take the average price of the stock in that path, and calculate the value of the option as if that path were the only one:

$$C^k(T) = \max\{\bar{S}_{[0,T]}^k - K; 0\}.$$

We repeat the process for all the simulations we ran, and then we average the option prices we found. Discounting the result to its present value will give us the final price of the option:

$$C(0) = e^{-rT} \bar{C}(T) = e^{-rT} \left(\frac{1}{N} \sum_{k=1}^N C^k(T) \right)$$

1.3.2 The antithetic variates technique

While the MC method does solve the problem, it presents a significant issue: the number of simulations required is very high. Having done the method for an Asian option in a previous work, I found that even after setting the sample to $N = 200,000$ and the time steps to $n = 2,000$ (i.e. trading approximately every business hour of the day for 250 days), the time during which the code was running was fairly high (≈ 600 seconds), particularly when running the code on an Intel i7-3770K 8-core processor with CPU @ 3.50 GHz. Even though the five prices found (one price for each repetition of the algorithm) were consistent with each other to a certain extent, there were still divergences, namely a maximum difference of 0.076% and standard deviation 0.0274 for the option price. These are differences that, at least under certain circumstances, might be too large to be acceptable, especially when the simulation sample and trading frequency are that high. While the sample variance above might seem small, the bigger indicator that we can still improve the process is the maximum difference (in that particular case, 0.076%). This is an issue that cannot be ignored. So what can we do about it?

One advantage of working with Monte Carlo simulations is that they allow for a wide range of *variance reduction* techniques. The most commonly used such methods are: common random numbers, antithetic variates, control variates, importance sampling, and stratified sampling.

The weapon of choice for this thesis is the antithetic variates (AV). The reason I chose this method is because it offers a clever workaround for the problem of simulation size, and also because it can be implemented by making a very minor modification to the source code that I used to generate the simulations. In other words, one might say that the AV method occurs naturally when seeking to reduce variance of Asian options priced via Monte Carlo simulations.

Take the case of the CRR tree with n periods (so no simulations yet). Since the tree is recombining, the total amount of possible outcomes for S_T is $n+1$, so one needs to run as many simulations as possible to sufficiently cover that spectrum of prices, i.e. we want a very high N . Suppose we choose $N = N^*$ and create my sample, which we can call sample A (which was created using the matrix Z_A of 1's and 0's). Using antithetic variates, we can create a matrix Z_B which has – in comparison to Z_A – *half* the amount of unique columns (i.e. the vectors $Z_1, Z_2, \dots, Z_{N^*/2}$), and fill in the remaining half by mirroring these same vectors, i.e.

$$Z_B = \left[\begin{array}{cccc|cccc} \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ Z_1 & Z_2 & \dots & Z_{N^*/2} & \mathbb{1} - Z_1 & \mathbb{1} - Z_2 & \dots & \mathbb{1} - Z_{N^*/2} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \end{array} \right],$$

where $\mathbb{1} - Z_j$ means component-wise deduction, i.e. $\mathbb{1} - Z_j(i), i = 1, \dots, n$.

In other words, we use the first half of Z_B in order to mirror the initial "half-sample", by taking half¹ of its simulated paths, and then mirroring each path at each step so that wherever simulation #1 goes up (down), simulation $\# \frac{N^*}{2} + 1$ goes down (up) and so on. In other words, for sample B we use only half the amount of simulations we did in A, and we complete the other half by mirroring these paths.

As the name suggests however, this is a technique that not only reduces the size of samples to be generated, but also reduces the variance. Suppose that we want to estimate the quantity $\theta = \mathbb{E}[h(X)] = \mathbb{E}[Y]$, where $h : \mathbb{R} \rightarrow \mathbb{R}$ some deterministic function. To do this, we generate two samples, say Y_1 and Y_2 . An unbiased estimator of θ is $\hat{\theta} = (\hat{\theta}_1 + \hat{\theta}_2)/2$, whose variance is

$$\mathbb{V}[\hat{\theta}] = \frac{\mathbb{V}[Y_1] + \mathbb{V}[Y_2] + 2Cov(Y_1, Y_2)}{4}.$$

If the two samples are independent, then $Cov(Y_1, Y_2) = 0$ so $\mathbb{V}[\hat{\theta}] = \mathbb{V}(Y_1)/2$. In the antithetic variates however, the second sample is created in such a way that Y_1 and Y_2 are no longer *iid* and $Cov(Y_1, Y_2) < 0$, which results in a reduced $\mathbb{V}[\hat{\theta}]$. In the stock path case, this translates to the creation of Z_B as described above.

1.4 The methodology ahead

Of course, for the LLN to work, we need independence within our sample observations, and sample B is anything but that. However, for a large enough N^* , the two samples

¹Due to this division by two, we get the minor limitation that the number N of simulations has to be even, in order for the matrices to be created.

A & B may start "looking" similar, i.e. their distributions (or at least their descriptive statistics) will be similar. This is what I will examine in *Chapter 2*, namely, what is the minimum value of N^* such that the two samples A & B of final stock values S_T^A and S_T^B exhibit similar distribution properties. After examining this minimum N^* via three different methods of statistical tests, I move on to *Chapter 3*, where I will use only the AV method – which, as showed, provides a "similar" sample to the independence method but with half the simulation cost – in order to zero in as much as possible on the price of an Asian option.

Remark: Ideally, the problem would be solved by finding an analytic lower bound (n^*, N^*). However, due to the non-linear nature of the problem, and more importantly, the notion of equality by distribution for the two samples, such an analytic bound would be very difficult to find, if not impossible.

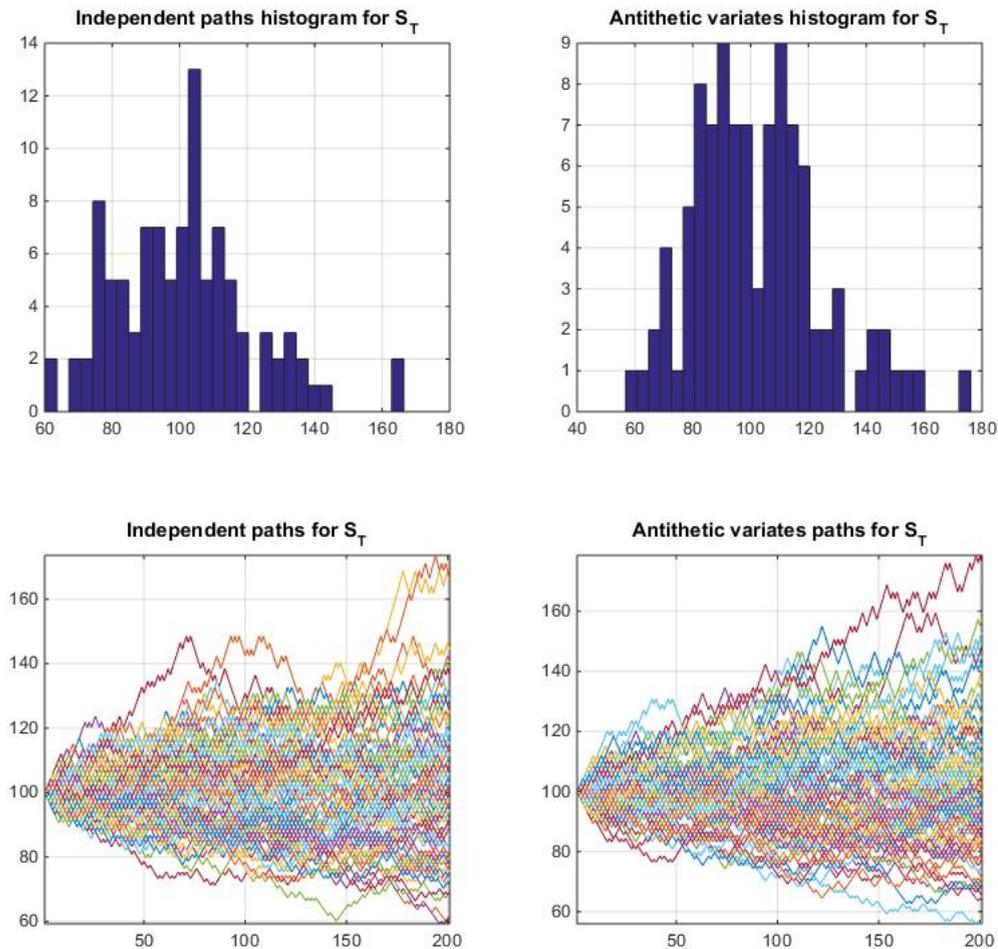


Figure 1.1: Results when $N < N^*$.

Figure 1.1 shows the differences between the MC (independent) and the AV method when $(n, N) = (200, 500)$. As one can see, the distributions of the two different vectors for S_T differ greatly, making it highly improbable that they are equal.

On the graph below however (fig. 1.2), we see that for $(n, N) = (500, 10000)$, the differences are fewer (from a statistical point of view). This reinforces the intuitive idea of the AV, that is, if we repeat the self-dependent process long enough, the end result will be similar to independence.

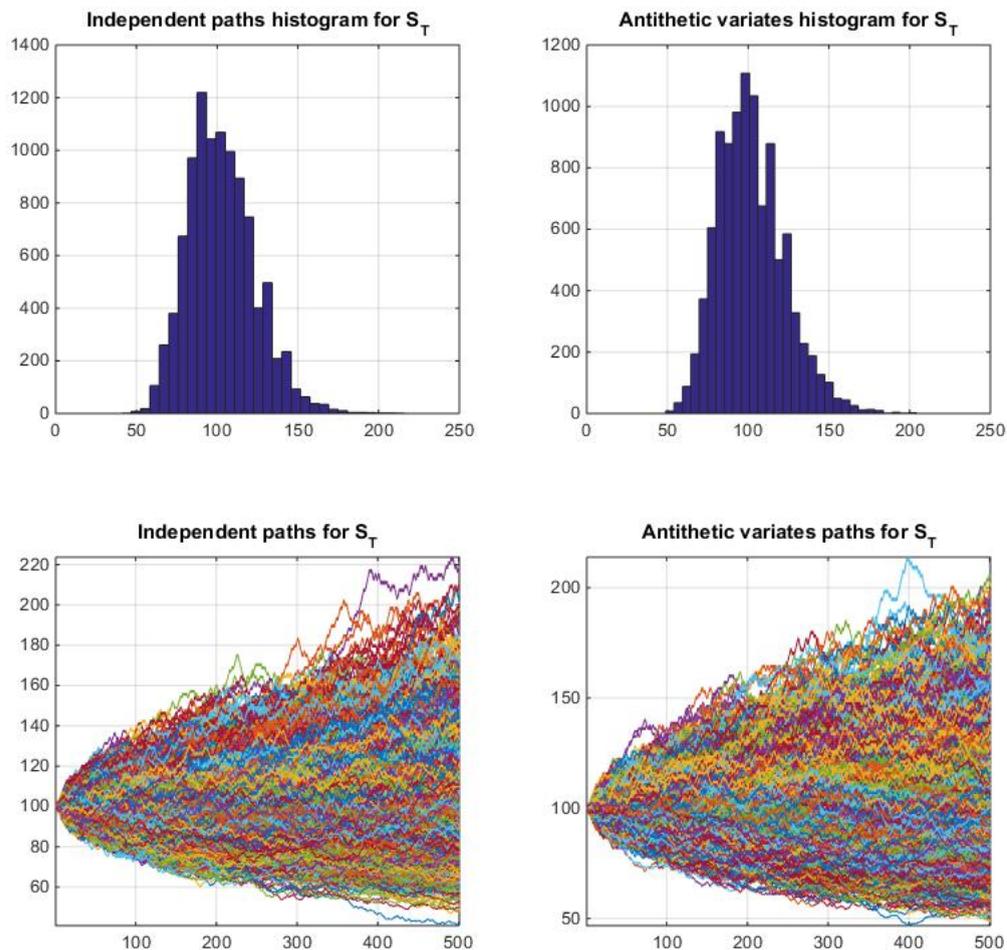


Figure 1.2: Results when $N > N^*$.

Chapter 2

Delineation of convergence

2.1 Statistical methodology

The purpose of this chapter is, as mentioned earlier, to find the minimum N for which the MC and the AV methods show no statistical difference with regards to the distribution of S_T . The way I chose to go about this is as follows:

1. Choose a starting value N_0 for the number of simulations.
2. Perform 100 simulations for each of the two methods with this given N_0 .
3. Check if the distributions of the resulting S_T are the same (for each pair of the 100 simulations).
4. If more than 1 pairs of samples exhibit different distributions (i.e rejecting the null hypothesis), increase N_0 and repeat the process until at least 99 pairs of the total 100 have the same distribution. When that happens, the resulting threshold of the method is the minimum N^* is the number we are looking for.

As one can see from the above description, the methodology is nothing more than a loop with an *if* condition that will help us reach the conclusion we seek in order to continue exclusively with the AV method in Chapter 3. As described, we get one sample from MC and one from AV for the N_0 we chose, then check to see if they have the same distribution, and repeat this process 100 times to make use of the LLN. We stop when no more than 1% of the pairs differ in distribution. The beginning value for N_0 is of course what defines by how much we will increase it on step 4. As seen on the end of the previous chapter, there seems to be a very satisfactory behaviour for $N = 10,000$, but since we want to examine the rate of convergence (and at the same time find the

minimum N for which we get a good enough result), starting so high may be a waste of computational power, so the starting point is set to be $N_0 = 1,000$. Special consideration is given at step #3: to perform the check for distribution equality I will employ three different methods of statistical testing which are described below.

2.2 Anderson-Darling test

2.2.1 Description

The 1-sample Anderson-Darling (AD) test [3] is a statistical test that checks whether a given sample of data comes from a specified probability distribution. Even though my first idea was to use the Kolmogorov-Smirnoff (KS) test [4], the AD choice might a better alternative for the particular samples to be examined: according to theory (i.e given a high enough n), the distribution of S_T is lognormal, which is a heavy-tailed distribution – an assertion that can be easily confirmed visually by the graphs at the end of Chapter 1. I thus chose the AD test over the KS, since the former is more effective when dealing with heavy-tailed distributions as it takes more into account the tail events.

However, since I am not comparing one sample against a fixed distribution but two samples on whether they come from the same distribution, the version I had to use was the 2-sample AD test. This presents a challenge because the AD test assumes no parameters of the target distribution to be estimated, so the set of critical values is distribution-free. While this is a desired quality from a statistical viewpoint, it also means that the test's critical values have to be approximated. Luckily, I was able to find a previous work on this issue done by Bahr ([2]), in which he describes the AD test, its computational challenge, and also writes the code for the 2-sample version. If each sample has size $n_i, i = 1, 2$, the formula for the test is

$$AD = \frac{n_1 n_2}{(n_1 + n_2)^2} \int_{-\infty}^{\infty} w(x) [F_1(x) - F_2(x)]^2 dF_{12}(x)$$

where $F_1(x)$ (resp. $F_2(x)$) is the ecdf (empirical CDF) of the sample produced by the MC simulation method (resp. from the AV method), $F_{12}(x)$ is the joint ecdf of both samples, and

$$w(x) = [F_{12}(x)(1 - F_{12}(x))]^{-1}.$$

Perhaps it is noteworthy to mention here that the AD test is directly derived from the Cramér-von Mises test simply by adding the $w(x)$ factor, which increases when $F_{12}(x)$ approaches 0 or 1. Interpreting this fact, we get an intuitive explanation of the test formula: we want the AD quantity to be "small" (since we do not want to reject H_0),

which means we want $w(x)$ to be small as well. This happens when $F_{12}(x)$ is not too close to either 0 or 1, so if $F_{12,S_T}(x) = \mathbb{P}[S_T^1 \leq x, S_T^2 \leq x]$ is the theoretical joint cdf, we see that $w(x)$ stays away from edge probabilities when **both** S_T^1 and S_T^2 remain close to the same value x with the probability of deviating on either side of x being equal to $1/2$. This confirms the intuition that we expect the two samples to have come from the same distribution when their ecdf's do not look too different.

2.2.2 Results

Implementing the algorithm for the AD test gives the following results. As described in the methodology, after choosing N , I performed 100 repetitions of the simulation process, and in the second column I write the amount of cases where the distributions were deemed not equal. The reason for the running time being so high is that, for a chosen N , the amount of simulations is $150N$, and that is only with regard to the final values S_T , i.e. *without* taking into account the previous $n - 1$ pseudo-random binary realizations required to produce a single simulation for the stock path, which means that the total number of simulations is $150 \times n \times N$.

N	# of rejections	Running time
1,000	10	249.241 s
2,000	7	466.555 s
5,000	7	1118.292 s
10,000	7	1709.702 s
20,000	12	3735.779 s

Table 2.1: Analytics of the 2-samples Anderson-Darling test.

Interestingly enough however, when looking at the profile summary of the code, it is not the number of simulations itself that makes the time increase so much, but the procedure of testing with AD. As mentioned in the description above, the many-samples version of the test has - to the extent of my knowledge - no official function integrated into MATLAB, so the reason it increases the time is because of its repetitive structure for approximating the p -value via iterative loops (even with $N = 1000$ and a single 1×2 vector of samples to consider, the running time was high enough).

2.3 An unexpected brick wall

At this point the analysis starts breaking down. The custom AD test function written by Bahr begins to take so much time to analyze the 100, 2-pair samples that it becomes essentially impossible to increase the sample in search of a better result, at least with

this method, since we do not observe the number of rejections going down when $N \uparrow$, as it was expected.

To try and tackle this issue, I employed a battery of different approaches. First, I increased the total amount of simulations to the maximum capacity allowed by my computer: $(n, N) = (1000, 900000)$ (the running time to create the simulations alone was about 20 minutes and the size of the simulated data was over 3GB). This way, I wanted to make sure that if there are any desired effects in the data within this kind of numerical range, I would be able to capture them by analyzing the numbers. The next step was to perform a wide range of tests on the data to see if the conclusions would allow me to proceed further.

Without manipulating the data in any way, I wanted to see how the distribution of S_T looked for the two different methods:

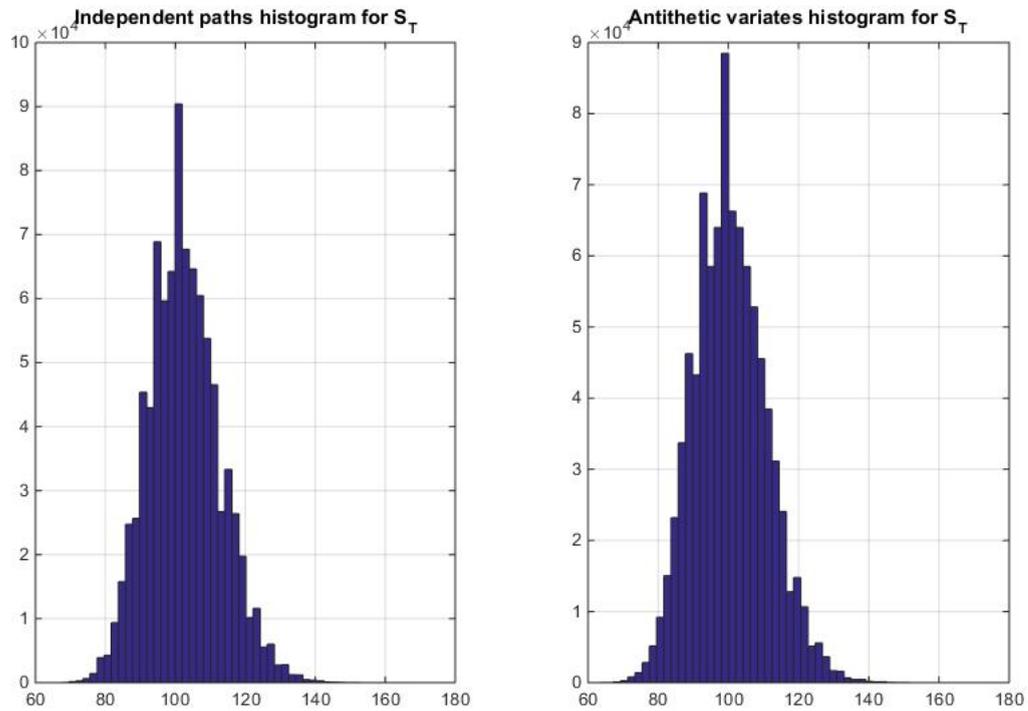


Figure 2.1: Histograms of S_T for MC (left) and AV (right).

Even though the graphs are not in the same scale, a quick eyeballing (by looking at the y -axis) would suggest that the distributions appear similar. This hypothesis however proved to be notoriously difficult to handle, as different methods of testing provide different answers. If we check the empirical CDF's of the two samples, we see that they differ around the center but not so much at the tails:

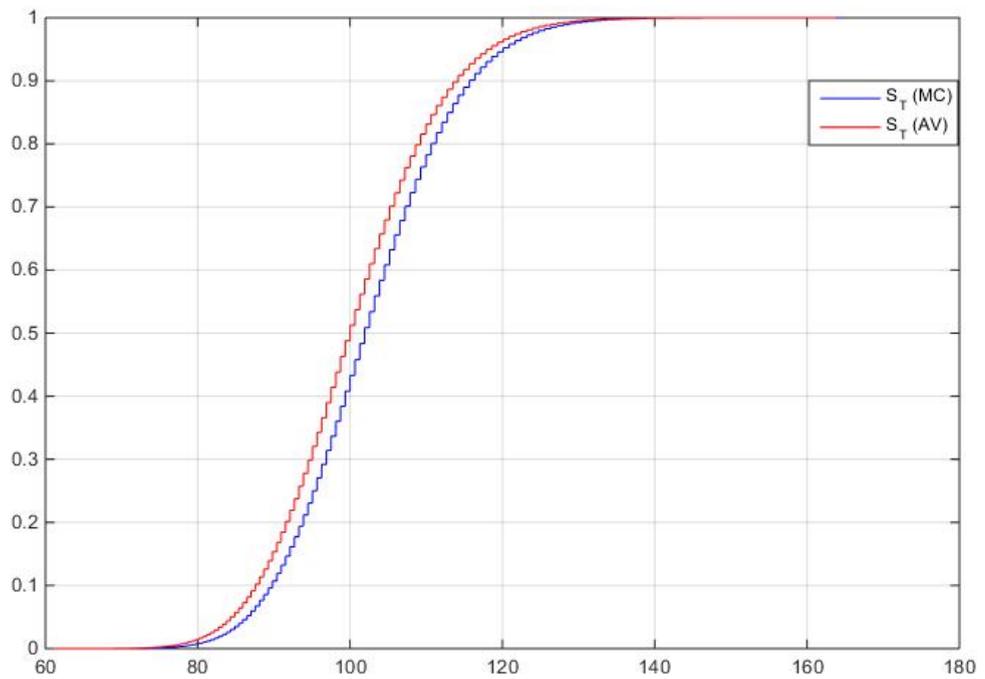


Figure 2.2: Empirical distribution functions for MC and AV.

A Q-Q plot confirms this, and the slight *S*-shape of the line shows that the MC method produces a range of S_T that has heavier tails than its AV counterpart:

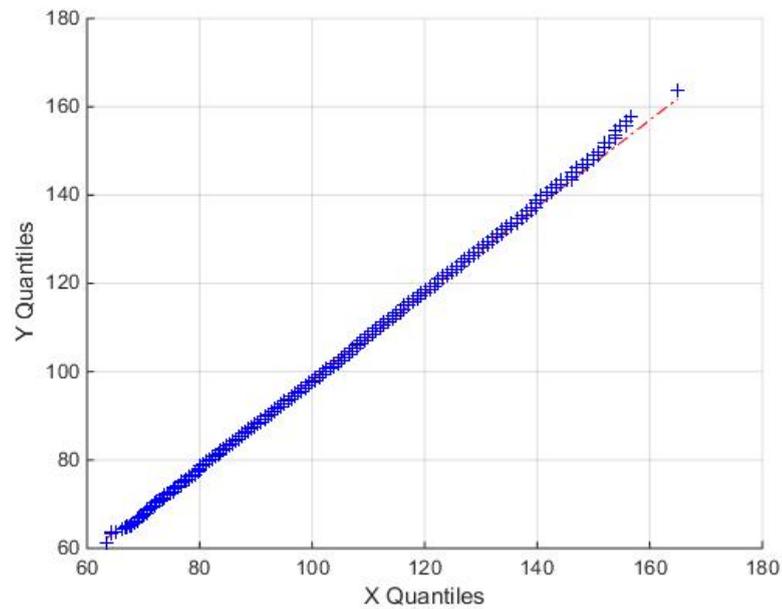


Figure 2.3: Q-Q plot of MC (*x*-axis) against AV (*y*-axis).

Since the visual inspection methods are susceptible to subjective interpretation, I proceeded to test the two samples using strictly statistical measures. The KS test rejected the null that the two samples have the same distribution, but other methods gave different results. Namely, the Kruskal-Wallis one-way ANOVA by ranks (a non-parametric test for checking whether samples come from the same distribution) does not reject the same H_0 :

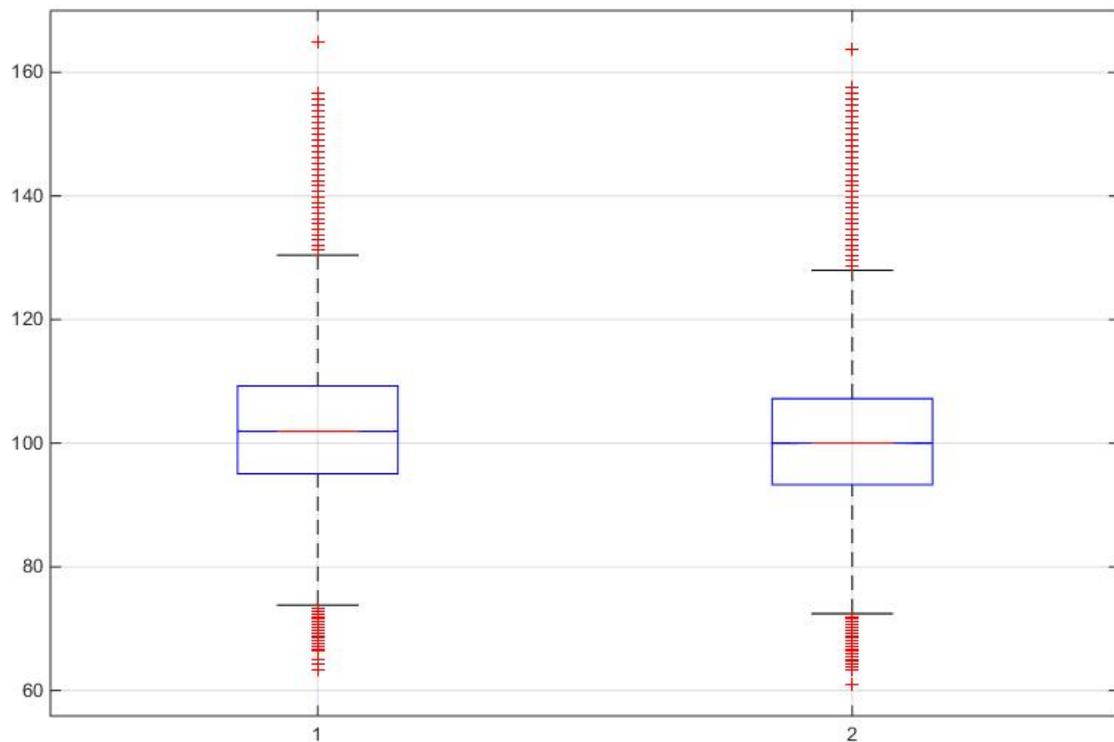


Figure 2.4: Kruskal-Wallis boxplot.

Kruskal-Wallis ANOVA Table					
Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	4.49198e+15	1	4.49198e+15	16636.96	0
Error	4.81508e+17	1799998	2.67505e+11		
Total	4.86e+17	1799999			

Figure 2.5: Kruskal-Wallis ANOVA table.

Another method I used to examine this discrepancy even further was to log the stock prices and then check the resulting samples for normality, since it is known that if $X \sim N(\mu, \sigma^2)$, then $Y := e^X \sim LN(\mu, \sigma^2)$.

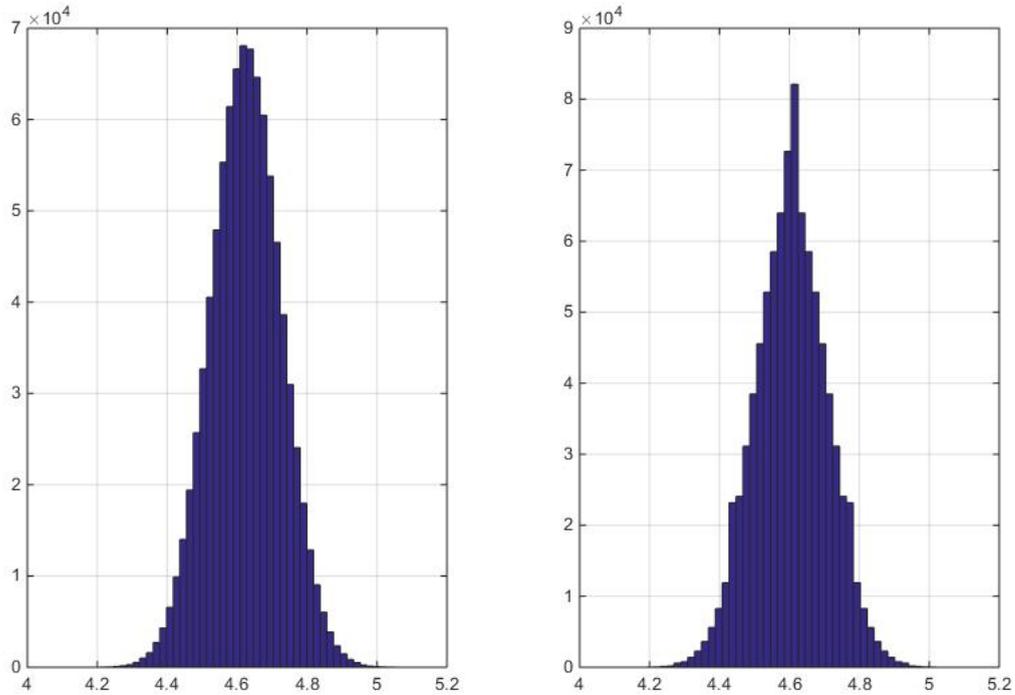


Figure 2.6: Histograms of $\log S_T$ for MC (left) and AV (right).

Different conclusions are reached here depending on the test one uses. The KS test rejects the null of the two samples coming from the same distribution, but it also rejects the null that each of the individual samples comes from a distinct normal distribution (even if we fit a normal distribution to the data and run the test against a distribution object with the resulting parameters). The Jarque-Bera test on the other hand, an omnibus, goodness-of-fit test which checks if the data have the skewness and kurtosis of a normal distribution (with unspecified parameters), does not reject H_0 for either of the two samples. This means that, according to the KS test, the resulting S_T 's are not lognormal in either case, whereas according to JB, both samples follow lognormal distributions.

Lastly, I fit a multitude of distributions to the data using the *allfitdist.m* function written by Mike Sheppard. The results here were also rather strange. The best fit for MC was a Birnbaum-Saunders distribution, although the lognormal fit had the same value of log-likelihood and the same values for the three information criteria (AIC, BIC, AICc)¹

¹For details regarding the information criteria as well as a brief description of the *allfitdist.m* function, check Appendix.

when compared to the best fit, so we may well consider the best fit to be a lognormal.

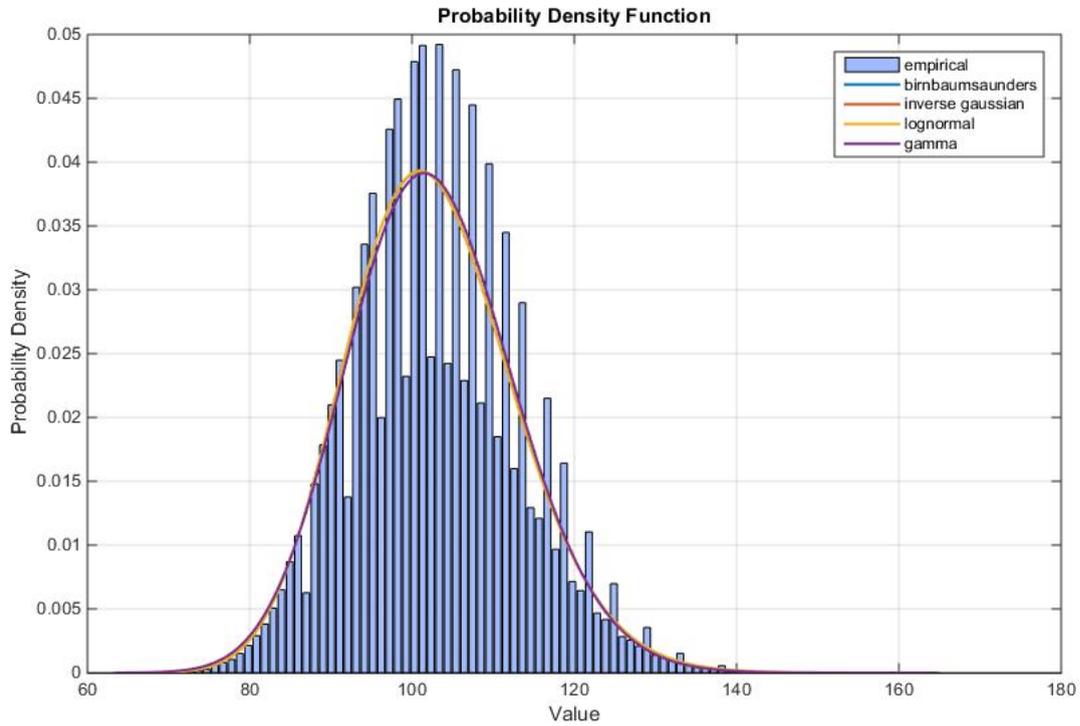


Figure 2.7: Fitted distributions of S_T (MC).

<i>Distribution</i>	Parameter 1	Parameter 2
Birnbaum-Saunders	$\beta = 102.0135$ (scale)	$\gamma = 0.1000$ (shape)
Inverse Gaussian	$\mu = 102.5240$ (scale)	$\lambda = 1.022 \times 10^4$ (shape)
Lognormal	$\mu = 4.6251$ (log location)	$\sigma = 0.0999$ (log scale)

Table 2.2: Fitted distributions to S_T (MC) and their estimated parameters.

Similar findings appear when running the analysis for the AV sample of S_T . Here, the lognormal distribution seems to be the first choice, and one can notice how close the parameters are to those from the MC sample, especially when comparing the lognormals. One detail which seems to come up often is that the AV sample creates a more compact (i.e more density in the center) range than that of the MC, as noted earlier in the Q-Q plots. This is a result that is to be expected, since the AV method spreads the prices in a more even way by definition. This leads me to believe that even more simulations may be required to achieve a good behavior just from the MC method, which is unfortunately outside the possible range of computational abilities for my hardware.

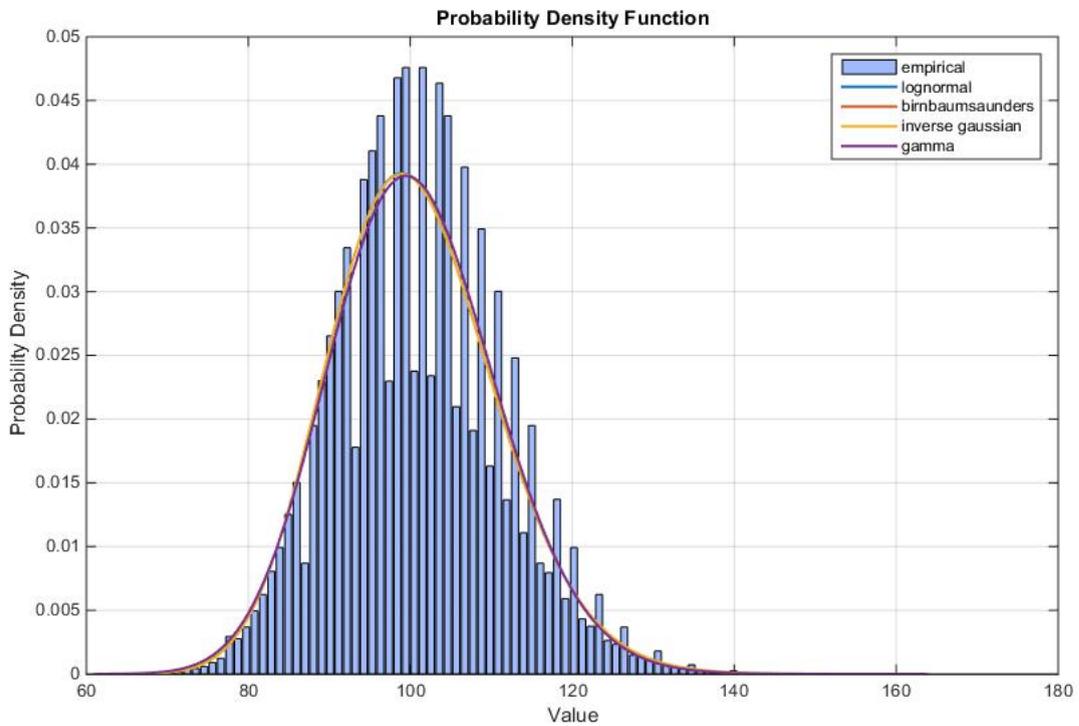


Figure 2.8: Fitted distributions of S_T (AV).

<i>Distribution</i>	Parameter 1	Parameter 2
Lognormal	$\mu = 4.6052$ (log location)	$\sigma = 0.1021$ (log scale)
Birnbaum-Saunders	$\beta = 100.0000$ (scale)	$\gamma = 0.1022$ (shape)
Inverse Gaussian	$\mu = 100.5221$ (scale)	$\lambda = 0.9602 \times 10^4$ (shape)

Table 2.3: Fitted distributions to S_T (AV) and their estimated parameters.

Due to the findings in page 12, I conclude that the AD test I used appears to be too time consuming (and possibly unstable) for usage in large data. Therefore I cannot conclude with certainty a sufficiently good value for N given the machine I use, at least with the tests and methods conducted so far. Note however that this applies not only to AD but the whole variety of tools used above. Thus, the results of the two next sections are not definitive values for N , rather different pieces of the same puzzle.

2.4 Epps-Singleton test

2.4.1 Description

The Epps-Singleton test (ES) [5] has the same null hypothesis with AD, but approaches the problem in a different way. In statistical testing, the p -value of a test is the probability of falsely rejecting H_0 (in this case that both samples are drawn from the same

population). The ES test checks for discrepancies not by comparing the ecdf's F_1 and F_2 but their Fourier transforms instead, i.e the empirical characteristic functions (cf) $\phi_1(t)$ and $\phi_2(t)$, and one of its advantages is that it belongs in the class of omnibus moment tests, which assess simultaneously whether *skewness* and *kurtosis* of the data samples are consistent with each other (in the univariate case, the assessment is done against the target distribution). For more details on this and the next page, please check [6].

Remark: One important arising issue is that the application of the ES test requires all observations to be independent, both within and across the samples. The later condition is easily met: the matrices generating the simulations for MC and AV are independent from one another by design. The first condition, however, is more problematic, since the nature of the AV method makes the corresponding sample inherently dependent on itself (we generate the first half of the paths and fill the remaining half by inverting the first one). Despite this, I believe it is not a breach of logic to apply the test nevertheless, since we are hoping to achieve a "pseudo-independence" after sufficiently many simulations with the AV method. In other words, it is expected that the test will reject H_0 in the beginning of the algorithm, since the dependence will create too much interference which will mess up the distribution of S_T . As indicated by the previous section though, this interference might decay (depending on which test we "ask") after a certain point of simulations, which is exactly what we are trying to pinpoint in the first place. The null hypothesis of the test is

$$H_0 : \phi_1(t) = \phi_2(t), -\infty < t < \infty.$$

The cf is defined as $\phi_k(t) = \int_{-\infty}^{\infty} e^{itx} dF_{n_k}(x)$, where $t \in \mathbb{R}$ and $i = \sqrt{-1}$. If each sample k has size n_k , we denote by X_{km} the m -th observation of sample k and by $F_{n_k}(x)$ the ecdf of that sample, so the (empirical) cf becomes

$$\phi_{n_k}(t) = \int_{-\infty}^{\infty} e^{itx} dF_{n_k}(x) = n_k^{-1} \sum_{m=1}^{n_k} e^{itX_{km}}$$

For its use, the ES test requires the use of a set of parameters t_1, t_2, \dots, t_k (in this case $k = 2$), which then need to be calibrated. In the original publication, Epps and Singleton find that $t_1 = 0.4$ and $t_2 = 0.8$, which they then standardize by an estimate of scale $\hat{\sigma}$, so the test is implemented with $\tilde{t}_j = t_j/\hat{\sigma}$, $j = 1, 2$ and $\hat{\sigma}$ is the semi-interquartile range:

$$\hat{\sigma} = \frac{1}{4} [(Y_U + Y_{U-1}) - (Y_L + Y_{L-1})],$$

where Y is the merger of both samples, in ascending order, and L is the largest integer $(n_1 + n_2)/4$ and $U = n_1 + n_2 - L$. Each X_{km} requires a transformation by

$$g(X_{km}) = (\cos t_1 X_{km}, \sin t_1 X_{km}, \cos t_2 X_{km}, \sin t_2 X_{km})',$$

which makes it into a 4×1 vector. We then average the transformed samples into g_k , $k = 1, 2$:

$$g_k = n_k^{-1} \sum_{m=1}^{n_k} g(X_{km}).$$

The ES test statistic is given by

$$W = (n_1 + n_2)(g_1 - g_2)' \hat{\Omega}^{-1} (g_1 - g_2),$$

where $\hat{\Omega}$ is an estimator of the covariance matrix of $\sqrt{(n_1 + n_2)}(g_1 - g_2)$:

$$\hat{\Omega} = \frac{1}{2} \frac{(\hat{S}_1 + \hat{S}_2)(n_1 + n_2)^2}{n_1 n_2},$$

with \hat{S}_1 and \hat{S}_2 being the sample covariance matrices of g_1 and g_2 , as given by

$$\hat{S}_k = n_k^{-1} \sum_{m=1}^{n_k} g(X_{km})g(X_{km})' - g_k g_k'$$

We have that $W \sim A\chi^2(r)$, where $r = \text{rank}(\hat{\Omega}^{-1})$. As seen from its formula, W measures the statistical difference between the ecdf's of both samples, standardized by the covariance matrices, with the cfs portraying the underlying distributions of the two samples. Epps and Singleton compared their test with the Anderson-Darling, the Cramér-Von Mises and the Kolmogorov-Smirnoff tests by performing simulations, and found [5] that:

- If the data used are realizations of discrete random variables, apply the ES test.
- If the data comes from a continuous distribution, the KS test usually has a lower power than the ES test.
- Sometimes, the Anderson-Darling and the Cramér-von Mises tests can have a higher power than the ES test.

2.4.2 Results

After running the algorithm with the ES test for $(n, N) = (250, \cdot)$, the results are as follows:

N	# of rejections	Running time
1,000	3	12.878 s
2,000	2	23.615 s
5,000	1	59.116 s
10,000	5	118.644 s
20,000	16	235.804 s

Table 2.4: Analytics of the 2-samples Epps-Singleton test.

It is clear that the erratic behavior seen in the AD test continues here as well. The only difference is that the running times are significantly lower, which allows for an increase in the parameter values. Taking advantage of this, I increased n to 500, and the results I got were

N	# of rejections	Running time
20,000	20	476.713 s
30,000	45	700.667 s
40,000	83	1177.321 s

Table 2.5: Extended analytics of the 2-samples Epps-Singleton test.

It is clear that these tests do not give the results that I was expecting to get; if anything, one might say they give the exact opposite ones, adding further evidence to the notion that the two samples cannot be of the same distribution. Even in this case though, it is perhaps still possible to credit this discrepancy between tests to the relatively small amount of simulations I was able to create for the analysis. In the end, what matters is that the AV sample achieves lognormality, which seems to be the case, as the results in section 2.3 show.

2.5 A numerical method

2.5.1 Description

The last method for comparing the samples is not an exact statistical test as the previous cases were; instead, it is a more crude approach to the issue, but given the indecisive answers from the previous sections, I felt the need to examine the problem from more than one aspect, so I also included this method (provided to me by *Robert Fitzner*) for the sake of curiosity.

If we define $\mu_{MC} = \bar{S}_T^{MC}$ and $\sigma_{MC} = \frac{1}{N-1} \sum_{l=1}^N (S_T^{MC}(l) - \bar{S}_T^{MC})^2$ (and identically for the AV case), then we create the following metric:

$$Q = \frac{|\mu_{MC} - \mu_{AV}|}{\max\{\sigma_{MC}, \sigma_{AV}, |\sigma_{MC} - \sigma_{AV}|\}}$$

and we preselect a certain numerical threshold q such that we deem the two samples as being "too different" whenever $Q > q$. Since this is an "artificial" method, the tricky part is to pick the proper value for q such that the analysis makes sense and that we do not make mistaken decisions due to being off scale.

If we examine Q , we see that the term $|\sigma_{MC} - \sigma_{AV}|$ will be very close to 0, so we can ignore it. Since both σ_{MC} and σ_{AV} are strictly positive, we find that $Q \in [0, \infty)$. We want the sample means to be as close to each other as possible (sample standard deviations were found to relatively deviate very little), we would ideally see $\mu_{MC} \approx \mu_{AV}$, so with enough simulations we will (hopefully) have $Q \rightarrow 0^+$. Thus, a reasonable room for error could be $q = .05$.

2.5.2 Results

Due to the lack of intensive statistical testing, this method allows for higher starting values for (n, N) . I start with $n = 500$ and reach the following numbers:

N	# of rejections	Running time
5,000	6	117.840 s
10,000	3	233.861 s
20,000	0	463.633 s
50,000	0	1175.881 s

Table 2.6: Analytics of the means comparisons for $q = 0.05$.

2.6 The final threshold

The table above shows convergence to happen rather quickly, so this method is the only one that gave an explicit answer to the question we posed, namely that $N = 20,000$. This is thus the number from which I will start trying to apply the AV method in order to calculate the "right" price of the Asian call option. The efficiency of this lower bound however will be tested in accordance to how different the prices resulting from the algorithm will be. This means that a large increase of N may likely be required before we start seeing a strong enough convergence.

Chapter 3

Implementing antithetic variates

The purpose of this chapter is rather straightforward: to take the low enough N from the previous chapter and start increasing it into simulations until we get a good enough price. This however presents us with other challenges. The results from Chapter 2 are deemed inconclusive, so the starting point of $N = 20,000$ might be inefficient. In addition, what defines a "good enough" price? In other words, when does convergence of the option price start being sufficiently good?

To answer these two issues, let's look at them by the order in which they appear:

1. The methodology I will follow is to simulate $k = 100$ times for a given N (resulting in 100 prices) and compare the prices with each other to see how much they differ. For bigger values of N (where hopefully convergence has started to take effect). If the value of N is too low, it will show up immediately by the distribution of the prices: the ideal distribution is highly centered around the "true" price with very few deviations from it.
2. Obviously, convergence is in relation to this real price of the option I mention above. The idea was to use real data, but Asian options are not traded in the same volume as vanilla options, so an internet search on market prices for Asian options gave back zero results (after consulting with *Jonas Gudmundsson*, Head of Risk & Valuation Services at SEB, he told me that I would not be able to get access to that kind of data from any public database). As such, it is not possible to compare against a target value, and I am thus limited to checking statistical properties of the results.

Despite this, there are certain theoretical limits to the option price, which we can at least check to see if the resulting prices make sense. Since a geometric average is always lower than an arithmetic average ([12]), we know that a geometric Asian option's price will be a lower bound for that of the respective arithmetic one. In [1], Kemna & Vorst show that the price of an arithmetic Asian option cannot be computed analytically because the distribution of the arithmetic sum is unknown, which is not true for the geometric case, for which they proceeded to find a closed-form solution. They also find that the price of an average-value option will always be lower than that of the corresponding European option, which comes as a result of $\bar{S}_{[0,T]}$ having reduced volatility in comparison with S_T . This means that even though I have no immediate value for which to check my results, there is at least a range in which the prices should be, and both values of the range can be computed analytically. Lastly, if the price vector components resulting from the larger values of N have a known distribution, I will try to create a confidence interval for the mean value $\bar{C}_0(N) = \frac{1}{k} \sum_{i=1}^k C_0(i, N)$. Convergence is considered to be achieved when the prices have a maximum range of around 5%, i.e when $\max\{C_0(N)\} - \min\{C_0(N)\} \approx 0.05$.

The parameters used were

<i>Name</i>	<i>Value</i>
T (time to maturity, starting at 0)	1 year
r (risk-free rate)	2%
σ (stock volatility)	20%
S_0 (starting price of the stock)	100
K (strike price of the option)	S_0
n (nodes for the tree)	1000
k (repetitions for each value of N)	200

Table 3.1: Parameter values.

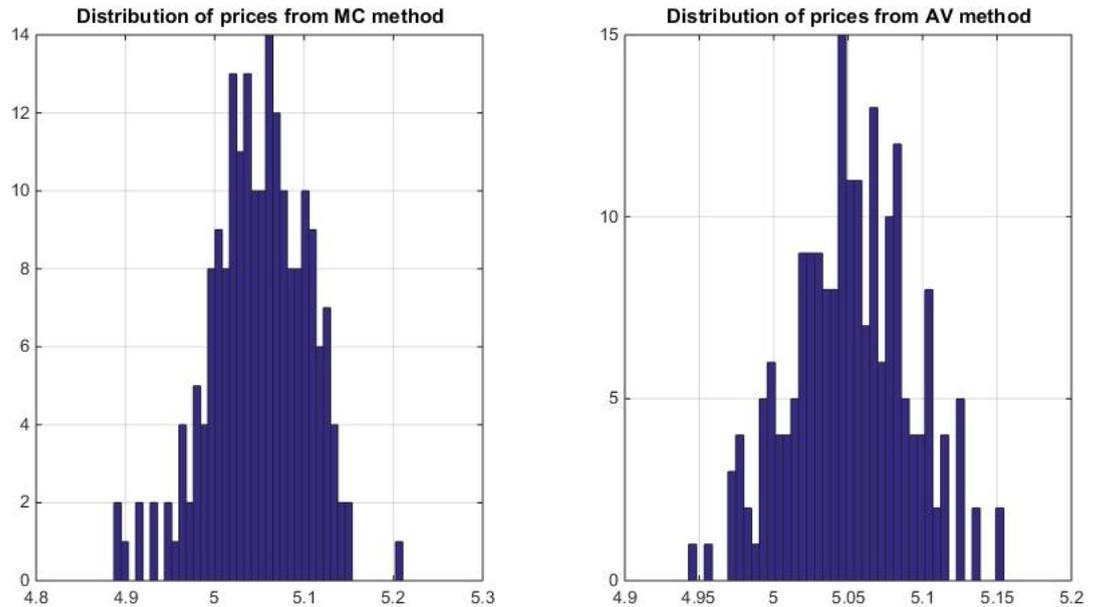
For these values, we have that the lower bound (as calculated by the Kemna-Vorst method) and the upper bound of the corresponding European option (as calculated by the famous Black-Scholes formula) for the arithmetic Asian call price are

$$[Price_{KV}, Price_{BS}] = [4.8633, 8.9160]$$

Of course, as described above, these two values serve simply as flags on whether or not the computations are correct, but they do not offer any additional information regarding the real price of the option, particularly since arithmetic averaging is used more frequently in practice than geometric.

Let us now finally see how the algorithm works for a range of values for N .

Starting at $N = 20,000$, I find that the benefits of the AV method are already present. When comparing with the MC method, the resulting samples come from the same distribution (according to a 2-sample KS test) and after I ran a 1-sample AD test on them individually, both samples seem to come from a normal distribution (with unspecified parameters).



However, the AV method provides a shorter range of prices when compared to MC, and their computational times were 299.7430 s for MC and 233.2455 s for AV. It should be reminded here that the N one feeds in the code is the number of **total** simulations, so the MC methods creates N total paths whereas the AV only $N/2$. Therefore, it seems irrelevant from this point forward to examine the MC method, so after that I only examine AV.

For $N = 50,000$, the differences become even smaller, and normality is again not rejected. However, the distance between the highest and lowest price is 0.1323, so according to the criterion named on p.24, we are still not close to a good enough solution.

To try and narrow down the price even more, I did three more repetitions of the code, with $N = \{1, 2, 3\} \cdot 10^5$. Normality starts becoming more and more accented in the distribution of C_0 , but the running time also increases significantly. For $N = 2 \cdot 10^5$, the difference $\max(C_0) - \min(C_0) = 0.0806$ which is still not close enough, and the running time was almost 42 minutes, so the trial at $3 \cdot 10^5$ is probably the limit before the system starts running out of memory.

The results of the last run of the algorithm show a maximum difference of .0604 between the computed prices which can be considered reasonable enough, therefore we see that a substantial amount of computations is required to sufficiently zero in on the price of an Asian call (MATLAB running time for $N = 3 \cdot 10^5$ was approx. 45 minutes).

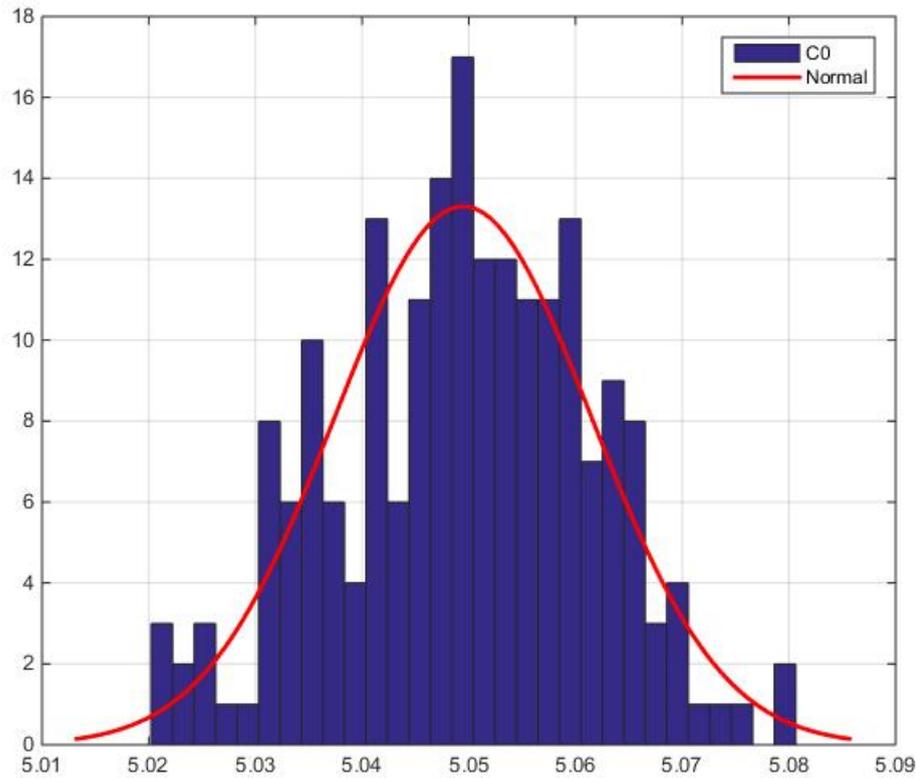


Figure 3.1: Distribution of prices with a fitted normal.

The AD test confirms here as well that the prices follow a normal distribution, with $\hat{\mu} = 5.0495$ and $\hat{\sigma} = 0.0121$.

Since the arithmetic Asian options do not have a closed-form solution, one might expect that different institutions value the same option in a different way, resulting in different prices. Therefore, we might consider the price of the option itself as a random variable. Another assumption that sounds reasonable is that any institution who would sell such a product can afford to perform at least 20,000 simulations as I did above, at which point normality was already present (although this number may vary depending on the parameters). This means that we can treat the price as a random variable coming from a normal distribution, but with unknown σ^2 , since as we saw, different values of N provided different ranges for the price.

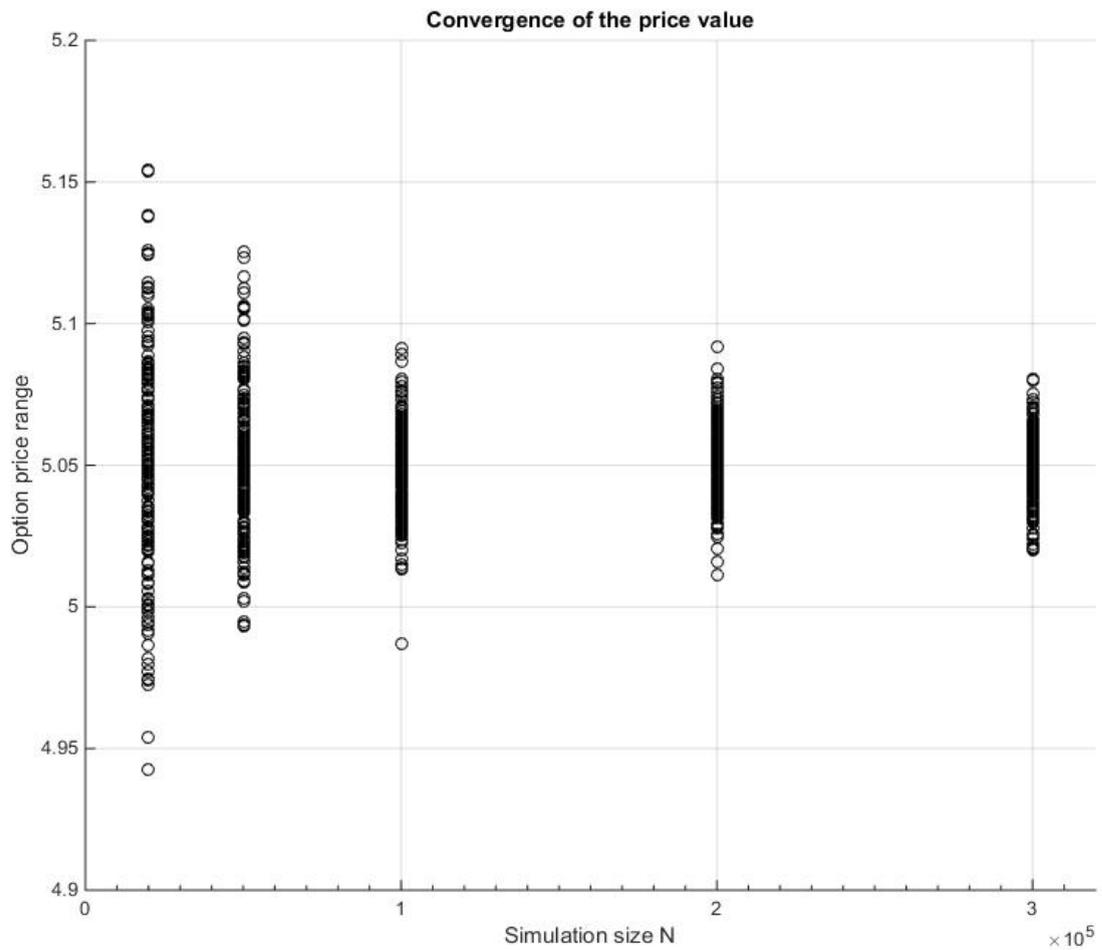
We can thus construct a $100(1-\alpha)\%$ confidence interval on μ by

$$\bar{x} - t_{1-\alpha/2, n-1} \frac{s}{\sqrt{n}} \leq \mu \leq \bar{x} + t_{1-\alpha/2, n-1} \frac{s}{\sqrt{n}},$$

and in this case, $n = k$, $\bar{x} = \frac{1}{k} \sum_{i=1}^k C_0(i)$, $s = \hat{\sigma}$. For $\alpha = .01$, the resulting CI for the mean value of the option price is

$$[5.0472, 5.0517]$$

The following picture illustrates how convergence occurs as a result of $N \uparrow$.



Chapter 4

Conclusions

In general, the method was found to provide the desired results, but not in the exact way I was looking for. While I was expecting to find that, given enough simulations, the MC and the AV methods of generating paths would produce results with statistically identical distributions, I found out that the opposite actually happens: when concerned about the distribution of S_T , the two methods produce different distributions, which MC usually generating more density in the tails – a result which is to be expected, given the construct of AV which offsets extreme movements towards any direction by mirroring them the other way.

One suggestion for future work would be to examine the distribution of $\bar{S}_{[0,T]}$ instead of simply S_T over the two methods, since it is the average that matters and not the final price when computing the price of the Asian option. In addition, one important factor I found out is that the parameters (n, N) cannot be separated indefinitely: for any given n , there is a certain range of values for N for which the analysis can produce meaningful results. This happens because, in the recombining binomial tree such as the one I used, the number of possible outcomes for S_T is $n + 1$. In other words, it is excessive to perform e.g. $N = 10,000$ simulations when $n = 50$, just as it is inefficient to perform $N = 1,000$ simulations when $n = 250$. The relationship between these two parameters has to do with how quickly the possible value spectrum for S_T is covered, i.e when does the sample size become sufficiently large (at which point one should increase n to $n + \delta$, since increasing N for the initial value of n would offer no useful information and would be just a waste of resources). This is a question that can be examined on future work by fixing n_0 , testing the distribution of S_T for different N 's, and after finding the minimum N for which said distribution is sufficiently good, increasing n_0 and repeating. Note that this is not a trivial matter void of applications: the value of T directly influences

the starting value of n ($n = 250$ is not the same for $T = 1$ and $T = 10$ years), and there are plenty of options, both vanilla and exotic, that have long-term maturities.

Lastly, the simulations throughout the entire analysis were generated using **MATLAB** commands that generate pseudorandom numbers. An alternative approach to this is the *quasi-Monte Carlo method*, which uses low-discrepancy sequences (also known as quasi-random or sub-random sequences). As the name suggests, a low-discrepancy sequence is one with the property that for all values of N (note that this is not the notation I use to denote simulation size), its subsequence $\alpha_1, \dots, \alpha_N$ has a low discrepancy, which is defined for the sequence $\{s_1, s_2, s_3, \dots\}$ with respect to the interval $[a, b]$ as

$$D_N = \sup_{a \leq c \leq d \leq b} \left| \frac{|\{s_1, \dots, s_N\} \cap [c, d]|}{N} - \frac{d - c}{b - a} \right|.$$

The notation $|\{s_1, \dots, s_N\} \cap [c, d]|$ denotes how many of the first n elements of the sequence are between c and d . Loosely speaking, the discrepancy of a sequence is low if the ratio of points in the sequence falling into an arbitrary set B is approximately proportional to the measure of B . The advantage of using these sequences is a faster rate of convergence: quasi-Monte Carlo has a rate of convergence close to $O(1/N)$ while the respective Monte Carlo method rate is $O(1/\sqrt{N})$.

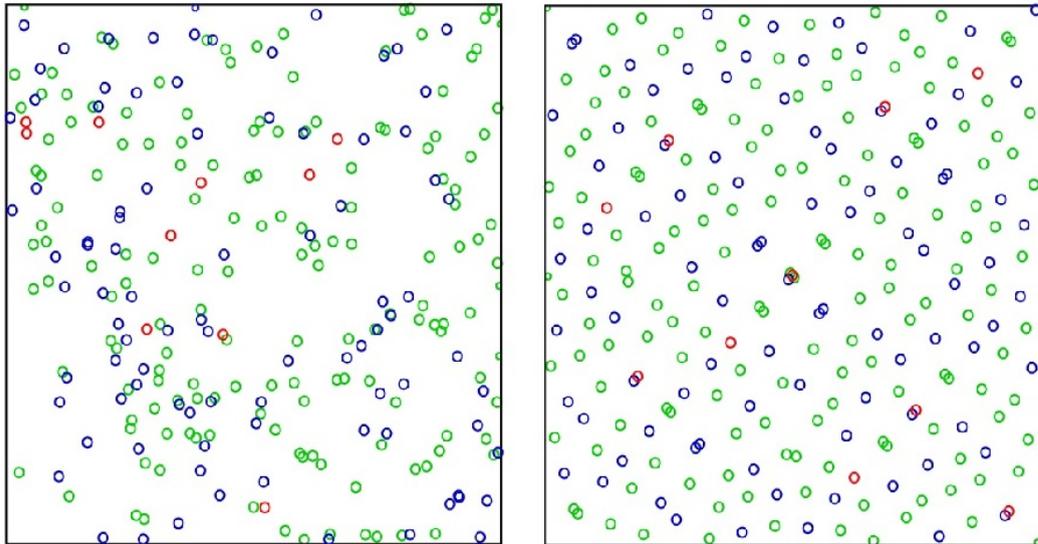


Figure 4.1: 256 points from a pseudorandom number source to the left, and Sobol sequence to the right (red=1, ..., 10, blue=11, ..., 100, green=101, ..., 256). Points from Sobol sequence are more evenly distributed. Image taken from Wikipedia.

Appendix A

MATLAB code & further notes

`allfitdist.m`

A direct link to download the script is found in the bibliography. The function takes a vector of the data to be analyzed as input (with specification as "discrete" or "continuous", depending on the type of data), and then fits a wide range of known distributions against the data, ordering the results by three information criteria presented below:

1) AIC. Introduced in [13], the Akaike Information Criterion is a measure of the relative quality of statistical models, given a set of data. It offers a relative estimate on the information that is lost when a given model is chosen as a representative of the "true" process that generates the data. If we have a certain statistical model for some data, with L the maximized value of the likelihood function and k the number of estimated parameters in the model, then the AIC value of the model is

$$AIC = 2k - 2\ln(L)$$

Given a set of candidate models for the data, the model to choose (according to the criterion) is the one with the minimum AIC value. Thus, we can say that AIC examines the relationship between goodness-of-fit and amount of parameters, penalizing the model when the former is not offset by the later (in a standard linear regression, more explanatory variables always increase the variation of outcomes explained by the model; AIC examines whether or not more variables are worth this increase they provide).

2) AICc. The corrected version of the AIC was originally introduced in [14] for linear regression only, but instigated a series of works that started with [15] and extended the cases in which it could be applied. AICc is a correction of the AIC for finite sample sizes, and its formula depends on the statistical model at hand. If the model is univariate,

linear, and has normally-distributed residuals, the formula is

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}$$

where n denotes the sample size and k is the number of parameters as before. As the formula demonstrates, AICc is simply AIC but with a greater penalty for extra parameters, thus rewarding parsimonious models. The use of AICc over AIC is preferred when n is not many times larger than k^2 , otherwise the probability of overfitting (i.e. selecting models with too many parameters) is significant. Generally, as AICc converges to AIC as $n \uparrow$, it should be used regardless.

3) BIC. Contrary to the two previous criteria who rely on classical statistics (frequentist inference) to maximize the likelihood function, the Bayesian Information Criterion (or Schwarz criterion) [16] uses Bayesian inference to calculate that quantity in its formula:

$$BIC = -2\ln(\hat{L}) + k\ln(n)$$

where \hat{L} is the maximized value of the likelihood function of the model M , i.e. $\hat{L} = p(x|\hat{\theta}, M)$, and $\hat{\theta}$ are the parameter values that maximize the likelihood function. The BIC is an asymptotic result derived under the assumptions that the distribution of the data is from the exponential family, and it is used in model selection problems where adding a constant to the BIC does not change the result.

The two main limitations of the BIC are that

1. the approximation under which it is derived is only valid in cases where n is much larger than k (number of parameters in the model).
2. it cannot handle complex collections of models as in the variable selection problem in higher dimensions.

Analytic descriptions of these criteria are beyond the scope of this thesis. For further reading on the topic of information criteria, there is a highly extensive database of academic articles available on the Internet, such as those cited in the bibliography.

Programming scripts

Below are the scripts that I wrote for my analysis. Sources for the functions I didn't write myself (such as the 2-sample AD and ES tests, and alldistfit.m) can be found in the bibliography. Lines 204 - 225 of the code can be found on the "Pricing Asian Options" page of the *Financial Instruments Toolbox* documentation in MATLAB.

```

1  clc; clear all; close all;
2  e = exp(1); alpha = .01; % level of significance for the test
3
4  T = 1;
5  r = .02;
6  sigma = .2;
7  s = 100;
8  K = s;
9
10 n = 500;
11 N = 10000;
12
13
14 S_A = [s*ones(1,N); zeros(n,N)];
15 S_B = S_A;
16 %F = zeros(N,1);
17 dt = T/n; % <-- important: the smaller the T, the lesser the n needed
18 u = e^(sigma*sqrt(dt));
19 d = 1/u;
20 q = (e^(r*dt)-d)/(u-d);
21 Q = binornd(1,q,[n,N]);
22 B = binornd(1,q,[n,N/2]);
23 J = ones(n,N/2);
24 %AV = [B(:,1:N/2), J-B(:,1:N/2)];
25 AV = [B, J-B];
26 for j=1:N
27     for i=1:n
28         Z = Q(i,j);
29         S_A(i+1,j) = S_A(i,j)*(u*Z+d*(1-Z));
30         Y = AV(i,j);
31         S_B(i+1,j) = S_B(i,j)*(u*Y+d*(1-Y));
32     end
33     %F(j) = max(mean(S(:,j))-K,0);
34 end
35 SA = S_A(end,:);
36 SB = S_B(end,:);
37 corr(SA(1:N/2),SA(N/2+1:end))
38 corr(SB(1:N/2),SB(N/2+1:end))

```

```

39 %X = [SA ones(N,1);SB 2*ones(N,1)];
40 r1 = log(SA);
41 subplot(1,2,1)
42 hist(r1,50)
43 grid on
44 r2 = log(SB);
45 subplot(1,2,2)
46 hist(r2,50)
47 grid on
48 figure
49 subplot(1,2,1)
50 hist(SA,50)
51 title('Independent paths histogram for S-T')
52 grid on
53 subplot(1,2,2)
54 hist(SB,50)
55 title('Antithetic variates histogram for S-T')
56 grid on
57
58 function [Q] = paths(n, N, alpha)
59 % N must be an even number, and greater than 8;
60 if mod(N,2) == 0
61     e = exp(1);
62     T = 1; r = .025; sigma = 0.2; s = 100;
63     S_A = [s*ones(1,N); zeros(n,N)];
64     S_B = S_A;
65     dt = T/n;
66     u = e^(sigma*sqrt(dt));
67     d = 1/u;
68     q = (e^(r*dt)-d)/(u-d);
69     P = binornd(1,q,[n,N]);
70     B = binornd(1,q,[n,N/2]);
71     J = ones(n,N/2);
72     AV = [B(:,1:N/2),J-B(:,1:N/2)];
73     for j=1:N
74         for i=1:n
75             Z = P(i,j);
76             S_A(i+1,j) = S_A(i,j)*(u*Z+d*(1-Z));
77             Y = AV(i,j);
78             S_B(i+1,j) = S_B(i,j)*(u*Y+d*(1-Y));
79         end
80     end
81     SA = S_A(end,:);
82     SB = S_B(end,:);
83     % [r, Had] = ADtest2(S1, S2, alpha); % AD test
84     % [r, Hes] = EStest2(S1, S2, alpha); % ES test
85     % [h, p, k2stat] = kstest2(S1,S2,'Alpha',alpha); % KS test
86     muMC = mean(SA); muAV = mean(SB);

```

```
87     sigmaMC = std(SA); sigmaAV = std(SB);
88     M = [sigmaMC, sigmaAV, abs(sigmaMC-sigmaAV)];
89     F = abs(muMC-muAV)/max(M);
90     if F > alpha
91         Q = 1;
92     else
93         Q = 0;
94     end
95 else
96     error('The number N of simulations must be even');
97 end
98 end
99
100 [f1,y1] = ecdf(SA);
101 [f2,y2] = ecdf(SB);
102 %[f3,y3] = ecdf(x);
103 figure
104 A = plot(y1,f1,'b-');
105 hold on
106 B = plot(y2,f2,'r-');
107 legend([A B], 'S-T (MC)', 'S-T (AV)', 'Location', 'NE')
108 grid on
109 [parmhat, parmci] = lognfit(SB);
110 pd = makedist('Lognormal', 'mu', parmhat(1), 'sigma', parmhat(2));
111 figure
112 qqplot(SA, SB)
113 grid on
114 figure
115 qqplot(SA, pd)
116 grid on
117
118 A = mean(S_A(:,1:end));
119 B = mean(S_B(:,1:end));
120 figure
121 subplot(1,2,1)
122 hist(A,50)
123 grid on
124 subplot(1,2,2)
125 hist(B,50)
126 grid on
127 figure
128 qqplot(A,B)
129 grid on
130
131 [D1, PD1] = allfitdist(SA, 'PDF');
132 [D2, PD2] = allfitdist(SB, 'PDF');
133
134 [Dr1, PDr1] = allfitdist(r1, 'PDF');
```

```

135 [Dr2, PDr2] = allfitdist(r2, 'PDF');
136
137 [kek1, kek2] = normfit(r1);
138 j = random(noru,800000,1);
139 figure
140 qqplot(r1,j)
141 grid on
142 [kek3, kek4] = normfit(r2);
143 nyt = random(noru,800000,1);
144 figure
145 qqplot(r1,nyt)
146 grid on
147
148 muMC = mean(SA); muAV = mean(SB);
149 sigmaMC = std(SA); sigmaAV = std(SB);
150 M = [sigmaMC, sigmaAV, abs(sigmaMC-sigmaAV)];
151 Q = abs(muMC-muAV)/max(M)
152
153 function V0 = mc_asian_call(K,s,r,sigma,T,n,N)
154 e = exp(1);
155 S = [s*ones(1,N); zeros(n,N)];
156 F = zeros(N,1);
157 dt = T/n;
158 u = e^(sigma*sqrt(dt));
159 d = 1/u;
160 q = (e^(r*dt)-d)/(u-d);
161 Q = binornd(1,q,[n,N]);
162 for j=1:N
163     for i=1:n
164         Z = Q(i,j);
165         S(i+1,j) = S(i,j)*(u*Z+d*(1-Z));
166     end
167     F(j) = max(mean(S(:,j))-K,0);
168 end
169 V0=e^(-r*T)*mean(F);
170 end
171
172 function V0 = av_asian_call(K,s,r,sigma,T,n,N)
173 e = exp(1);
174 S = [s*ones(1,N); zeros(n,N)];
175 %F = zeros(N,1);
176 dt = T/n;
177 u = e^(sigma*sqrt(dt));
178 d = 1/u;
179 q = (e^(r*dt)-d)/(u-d);
180 B = binornd(1,q,[n,N/2]);
181 J = ones(n,N/2);
182 AV = [B(:,1:N/2),J-B(:,1:N/2)];

```

```
183 for j = 1:N
184     for i = 1:n
185         Z = AV(i,j);
186         S(i+1,j) = S(i,j)*(u*Z+d*(1-Z));
187     end
188 end
189 Payoff = max(mean(S)-K,0);
190 V0 = exp(-r*T)*mean(Payoff);
191 end
192
193 clc; clear all; close all;
194 e = exp(1);
195 k = 200;
196 T = 1;
197 r = .02;
198 sigma = 0.2;
199 s = 100;
200 K = s;
201 n = 1000;
202 N = 300000;
203 dt = T/n;
204 %% Create RateSpec and StockSpec
205 %this section can be found on the "Pricing Asian Options" help page...
206 ... of MATLAB's documentation for the Financial Instruments Toolbox
207 StartDates = '25-August-2015';
208 EndDates = '25-August-2016';
209 Rates = r;
210 Compounding = -1;
211 Basis = 1;
212
213 RateSpec = intenvset('ValuationDate', StartDates, 'StartDates', ...
214     StartDates, ...
215     'EndDates', EndDates, 'Rates', Rates, 'Compounding', Compounding, ...
216     'Basis', Basis);
217 StockSpec = stockspec(sigma,s);
218
219 % Define the Asian option
220 Settle = '25-August-2015';
221 ExerciseDates = '25-August-2016';
222 Strike = K;
223 OptSpec = 'call';
224 % Kemna-Vorst closed form model
225 PriceKV = asianbykv(RateSpec, StockSpec, OptSpec, Strike, Settle, ...
226     ExerciseDates);
227 PriceLevy = asianbylevy(RateSpec, StockSpec, OptSpec, Strike, Settle,...
228     ExerciseDates);
229 PriceBS = blsprice(s, K, r, T, sigma);
230 %% Run the Antithetic Variates model
```

```
228 C = [];  
229 tic  
230 C0 = zeros(k,1);  
231 for i = 1:k  
232     C0(i) = av_asian_call(K,s,r,sigma,T,n,N);  
233 end  
234 toc  
235 figure  
236 hist(C0,40)  
237 grid on  
238 C = [C C0];  
239 [A, B] = size(C);  
240 Nvec = [2*10^4 5*10^4 10^5 2*10^5 3*10^5];  
241 figure  
242 for i = 1:B  
243     for j = 1:k  
244         hold on  
245         scatter(Nvec(i),C(j,i),[],'k')  
246     end  
247 end  
248 %plot(PriceKVvector,'r')  
249 hold off  
250 grid on  
251 title('Convergence of the price value')
```

Bibliography

- [1] Kemna, A.G.Z., Vorst, A.C.F. (1990). “A pricing method for options based on average asset values”, *Journal of Banking and Finance* 14, pp. 113-129.
- [2] Bahr, G. (2009). “Sustainability – is it contagious? Fairness in an intergenerational three-person dictator game with social interaction”.
- [3] Anderson, T.W., Darling, D.A. (1952). “Asymptotic theory of certain ”goodness-of-fit” criteria based on stochastic processes”, *Annals of Mathematical Statistics*, Vol. 23, No. 2, pp. 193-212.
- [4] Kolmogorov, A. (1933). “Sulla determinazione empirica di una legge di distribuzione”, *Giorn. Ist. Ital. Attuari*, Vol. 4, pp. 83–91.
- [5] Epps, T. W., Singleton, K.J. (1986). “An omnibus test for the two-sample problem using the empirical characteristic function”, *Journal of Statistical Computation and Simulation*, Vol. 26, pp. 177–203.
- [6] Goerg, S.J., Kaiser, J. (2009). “Nonparametric testing of distributions – the Epps-Singleton two-sample test using the empirical characteristic function”, *The Stata Journal* Vol.9, No. 3, pp. 454-465.
- [7] Klassen, T.R., “Simple, fast, and flexible pricing of Asian options”.
- [8] Feynman, R.P., Kleinert, H. (December 1986). “Effective classical partition functions”, *Physical Review A*, Vol.34, No. 6.
- [9] Cox, J., Ross, S., and Rubinstein, M. (1979). “Option pricing: A simplified approach”, *Journal of Financial Economics*, 7, pp. 229-224.
- [10] Black, F., Scholes, M. (1973). “The Pricing of Options and Corporate Liabilities”, *Journal of Political Economy*, Vol. 81, No. 3, pp. 637-654.
- [11] Rogers, L.C.G., Shi, Z. (December 1995). “The Value of an Asian Option”, *Journal of Applied Probability*, Vol. 32, No. 4, pp. 1077-1088.

-
- [12] Beckenbach, E.F., Bellman, R.E. (1971). “Inequalities”, *Volume 30 of Ergebnisse der Mathematik und ihrer Grenzgebiete*.
- [13] Akaike, H. (1973). “Information theory and an extension of the maximum likelihood principle”, In Petrov, B. N., Csaki, B. F., *2nd International Symposium on Information Theory*, Academiai Kiado: Budapest, pp. 267–281.
- [14] Sugiura, N. (1978), “Further analysis of the data by Akaike’s information criterion and the finite corrections”, *Communications in Statistics - Theory and Methods*, Vol. A7, pp. 13–26.
- [15] Hurvich, C. M., Tsai, C.-L. (1989), “Regression and time series model selection in small samples”, *Biometrika*, Vol. 76, pp. 297–307.
- [16] Schwarz, G. E. (1978), “Estimating the dimension of a model”, *Annals of Statistics*, Vol. 6, No. 2, pp. 461–464.
- [17] <http://www.mathworks.com/matlabcentral/fileexchange/34943-fit-all-valid-parametric-probability-distributions-to-data/content/allfitdist.m>