WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

IB9JH0
C++ for Quantitative Finace

Monte Carlo Method for One-Factor Option Pricing

This is to certify that the work I am submitting is my own. All external references and sources are clearly acknowledged and identified within the contents. I am aware of the University of Warwick regulation concerning plagiarism and collusion.

No substantial part(s) of the work submitted here has also been submitted by me in other assessments for accredited courses of study, and I acknowledge that if this has been done an appropriate reduction in the mark I might otherwise have received will be made.

ID: 1991391

May 11, 2020

Contents

1 Introduction

In this project, we will price European and Asian options using two Monte Carlo methods and discuss how the framework can be extended to price other exotic options. The methods are coded in C++ with visualisations made with Python 3.7 using the `MatPlotLib` and `Numpy` libraries. For reference, any CPU time stated in this report is measured with the `Stopwatch` class on a 2016 Dell Latitude E7470 with an Intel(R) Core(TM) i7-6600U CPU @ 2.60 GHz processor and 16.0 GB of RAM.

We will follow the standard notation and denote stock price with S, strike price with K, interest rates with r, dividends with q, cost of carry b, volatility with σ and time to maturity with T.

We start by stating our assumptions, followed by some numerical analysis of the option prices and the pricing errors when compared to closed form solutions or approximations. We conclude the report by comparing the two main methods, the Euler method and the exact simulation, in particular the algorithm complexities and pricing errors.

2 Setup and assumptions

For this assignment, we will assume that the underlying asset is a geometric Brownian motion, i.e. S_t follows the stochastic differential equation

$$dS_t = S_t(r-q)dt + \sigma S_t dW_t$$

where $\sigma > 0$, r and q are constants and W_t is a Wiener process.

We will perform the following Monte Carlo methods and compare their accuracy to risk-neutral closed form results obtained by a Martingale pricing approach.

$$\text{Euler method: } \begin{cases} dS_t = S_t(r-q)dt + \sigma S_t dW_t \\ S_{t+\Delta t} = S_t + dS_t \end{cases}$$

$$\text{Exact simulation: } \begin{cases} X_0 = \log(S_0) \\ dX_t = \nu dt + \sigma dW_t; \quad \nu = r - q - \frac{\sigma^2}{2} \\ X_{t+\Delta t} = X_t + dX_t \\ S_t = \exp(X_t) \end{cases}$$

The option price, V_t, is calculated as the expected value of the discounted payoff

$$V_t = \exp(-rT)\,\mathbb{E}[h(S_T)] \approx \exp(-rT)\frac{1}{N}\sum_{i=1}^{M} h(S_{T_i})$$

where M is the number of simulations and h is the option's payoff function which can either depend on the final value of the asset price, S_T, or the entire path, e.g. Asian options or barrier options.

3 Numerical analysis

3.1 Standard deviations and variance reduction

The standard deviation and error of the option prices for both Monte Carlo methods is calculated with the following.

$$\text{Standard deviation:} \quad \hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{M} h(S_{T_i})^2}{M} - \left\{\frac{\sum_{i=1}^{M} h(S_{T_i})}{M}\right\}^2}$$

$$\text{Standard error:} \quad \frac{\hat{\sigma}}{\sqrt{M}}$$

The standard deviation of the option prices was reduced by using the antithetic variates technique. Since the path of the Wiener process has normally distributed increments with mean zero and a standard deviation of $\sqrt{dt}$, we know that the same applies to the antithetic path, i.e. the negated values of W_t. We therefore simulate the following

$$\text{Euler method:} \quad \begin{cases} dS_t^{\pm} = S_t(r-q)dt \pm \sigma S_t dW_t \\ S_{t+\Delta t} = S_t + dS_t^{\pm} \end{cases}$$

$$\text{Exact simulation:} \quad \begin{cases} X_0 = \log(S_0) \\ dX_t^{\pm} = \nu dt \pm \sigma dW_t; \quad \nu = r - q - \frac{\sigma^2}{2} \\ X_{t+\Delta t} = X_t + dX_t^{\pm} \\ S_t = \exp(X_t) \end{cases}$$

and observe that the mean option price remains the same

$$\mathbb{E}[V_{t_i}] = \exp(-rT)\,\mathbb{E}\left[\frac{h(S_{T_i}^+) + h(S_{T_i}^-)}{2}\right] = \exp(-rT)\,\mathbb{E}[h(S_T)]$$

whereas the variance is significantly reduced (at least halved) due to the negative covariance between $h(S_T^+)$ and $h(S_T^-)$, i.e.

$$\begin{aligned} \mathrm{Var}[V_{t_i}] &= \mathrm{Var}\left[\frac{h(S_{T_i}^+) + h(S_{T_i}^-)}{2}\right] \\ &= \frac{\mathrm{Var}[h(S_{T_i}^+)] + \mathrm{Var}[h(S_{T_i}^-)] + 2\mathrm{Cov}[h(S_{T_i}^+, h(S_{T_i}^-)]}{4} \\ &= \frac{\mathrm{Var}[h(S_{T_i})]}{2} + \frac{\mathrm{Cov}[h(S_{T_i}^+), h(S_{T_i}^-)]}{2} \end{aligned}$$

3.2 Finite difference methods and path recycling

To evaluate the delta and the gamma of the options, we used the explicit Euler method

$$\begin{aligned} \Delta &= \frac{\partial V_t}{\partial S_t} = \frac{V_{j+1} - V_{j-1}}{2dS_t} + \mathcal{O}((dS_t)^2) \\ \Gamma &= \frac{\partial^2 V_t}{\partial S_t^2} = \frac{V_{j+1} - 2V_j + V_{j-1}}{dS_t^2} + \mathcal{O}((dS_t)^2) \end{aligned}$$

for $j = 1, \ldots, M-1$. The variance of the two greeks is calculated using

$$\begin{aligned} \mathrm{Var}\left(\sum_{i=1}^{N} a_i X_i\right) &= \sum_{i=1}^{N} a_i^2 \mathrm{Var}[X_i] + 2 \sum_{1 \le i < j \le N} a_i a_j \mathrm{Cov}(X_i, X_j) \\ &= \sum_{i=1}^{N} a_i^2 \mathrm{Var}[X_i] + 2 \sum_{1 \le i < j \le N} a_i a_j \sqrt{\mathrm{Var}[V_i]\mathrm{Var}[V_j]}\mathrm{Corr}(X_i, X_j) \end{aligned}$$

and the following approximation: $\text{Var}[V_{j+1}] \approx \text{Var}[V_j] \approx \text{Var}[V_{j-1}]$.

By using the same M paths for the Wiener processes when calculating the price of an option for a given stock price, we ensure that the values of V_{j+1}, V_j, and V_{j-1} are not obtained by independent Monte Carlo simulations. Hence they will be dependent on one another and have a positive correlation, resulting in reduced variance.

The main problem with this method is that the values tend to blow up as dS_t approaches zero. A safer alternative would be to use the pathwise derivative method instead of the finite difference method.

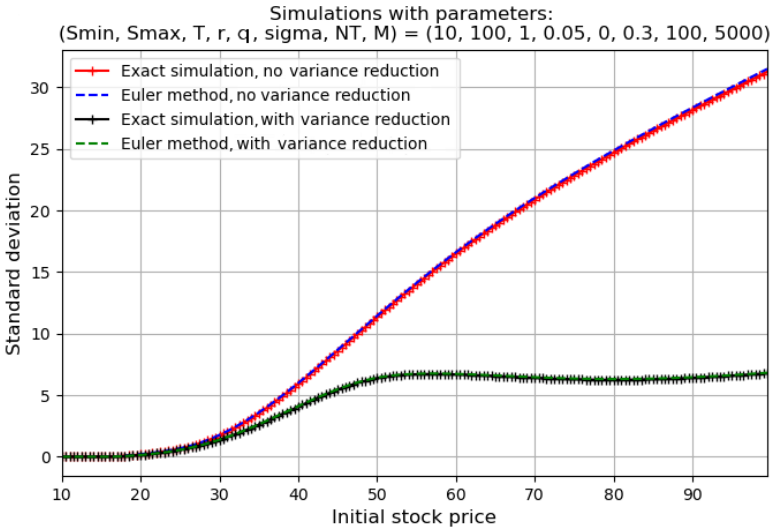


Figure 3.1: Demonstration of variance reduction for a European call option

3.3 User defined accuracy

To calculate the minimum number of simulations, N, needed for some user defined accuracy ϵ and some user defined confidence level $1-\alpha$, it is first necessary to perform M Monte Carlo simulations and observe the maximum standard deviation over the range of stock prices, σ_{MC}. Afterwards, the number of simulations required is

$$||V_{\text{MC}} - V||_\infty = \epsilon \geq \Phi^{-1}\left(1-\frac{\alpha}{2}\right) \cdot \frac{\sigma_{\text{MC}}}{\sqrt{M}} \Rightarrow N \geq \left(\Phi^{-1}\left(1-\frac{\alpha}{2}\right) \cdot \frac{\sigma_{\text{MC}}}{\epsilon}\right)^2$$

where Φ is the normal cumulative distribution function. This ensures that the price of the option for all stock prices in a given range is $100(1-\alpha)\%$ accurate to within $£\epsilon$.

4 Software design

The code was written with the single responsibility principle in mind. Each class was given a single purpose and then the main functionality was implemented in `MonteCarlo`. When applicable, a Standard Template Library (STL) algorithm or data structure was used and the main option parameters were stored in `OptionData`, which uses `boostparameter.hpp`.

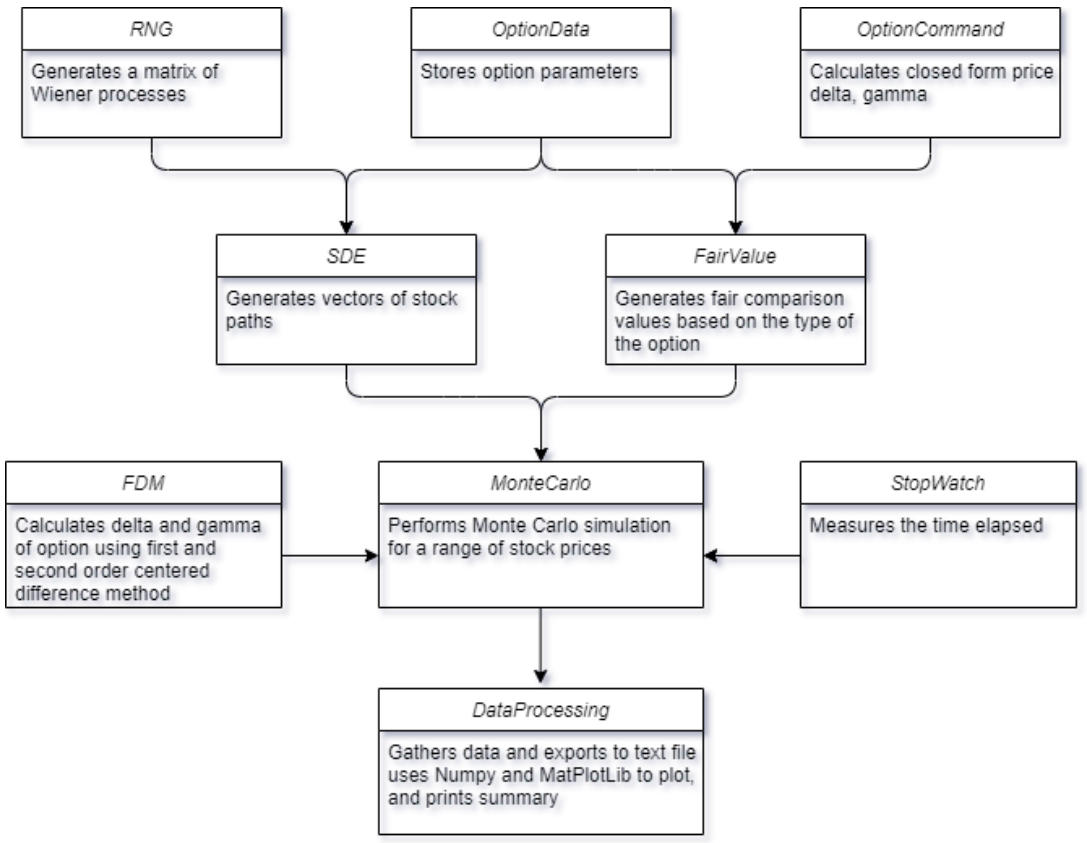


Figure 4.1: Class diagram showing the collaboration among the elements of the projects and describing the functionalities performed by each system

5 Option pricing

For both methods, the Wiener processes are generated and stored in a matrix for path recycling. They are then used to generate two matrices of stock price paths, S^+ and S^-, for initial stock prices ranging from $S_{\max}$ to $S_{\min}$ with increments dS_t.

5.1 Option payoffs

After the stock price paths have been generated, each vector is sent into the `OptionData` class to calculate the payoff, whether it is a path-dependent or not. The input of the payoff function is a vector and the output depends on the type of option.

$$\text{European options:} \begin{cases} \text{Call payoff:} & \max(S_T - K, 0) \\ \text{Put payoff:} & \max(K - S_T, 0) \end{cases}$$

$$\text{Asian options:} \begin{cases} \text{Arithmetic call payoff:} & \max\left(\frac{1}{N}\sum_{i=1}^{N} S_i - K, 0\right) \\ \text{Arithmetic put payoff:} & \max\left(K - \frac{1}{N}\sum_{i=1}^{N} S_i, 0\right) \\ \text{Geometric call payoff:} & \max\left(\sqrt[N]{\prod_{i=1}^{N} S_i} - K, 0\right) \\ \text{Geometric put payoff:} & \max\left(K - \sqrt[N]{\prod_{i=1}^{N} S_i}, 0\right) \end{cases}$$

The code is structured to allow for additional option types to be easily added. To this end, the option's payoff function should be added to the `OptionData::payoff` function, a closed form solution can be added for the option's price, delta and gamma in `OptionCommand` as derived classes (optional, only for accuracy comparison), and finally the `FairValue` constructor should reference the right pointers.

5.2 Comparison to closed form solutions

To compare the accuracy of our Monte Carlo simulations, we implemented the closed form solutions into a derived class in `OptionCommand`. For the European options, we used the Black Scholes formulas

$$\text{European call:} \begin{cases} \text{Price:} & C(S_t, t) = S_t \exp(-q(T-t))N(d_1) - K\exp(-r(T-t))N(d_2) \\ \text{Delta:} & \Delta_C = \exp(-q(T-t))N(d_1) \\ \text{Gamma:} & \Gamma_C = \exp(-q(T-t))N'(d_1) = \dfrac{\exp(-q(T-t))}{S_t\sigma\sqrt{T-t}\sqrt{2\pi}}\exp\left(\dfrac{-d_1^2}{2}\right) \end{cases}$$

$$\text{European put:} \begin{cases} \text{Price:} & P(S_t, t) = K\exp(-r(T-t))N(-d_2) - S_t \exp(-q(T-t))N(-d_1) \\ \text{Delta:} & \Delta_P = \exp(-q(T-t))(N(d_1) - 1) \\ \text{Gamma:} & \Gamma_P = \exp(-q(T-t))N'(d_1) = \dfrac{\exp(-q(T-t))}{S_t\sigma\sqrt{T-t}\sqrt{2\pi}}\exp\left(\dfrac{-d_1^2}{2}\right) \end{cases}$$

where q is a continuously compounded dividend yield and d_1 and d_2 are

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left(\log\left(\frac{S_t}{K}\right) + (r - q + \frac{\sigma^2}{2})(T-t)\right)$$
$$d_2 = d_1 - \sigma\sqrt{T-t}$$

For the geometric Asian options, it can be shown[1] that they have the following closed form solution shown

$$\text{Geometric Asian call:} \begin{cases} \text{Price:} & C_{\text{GAC}}(S_t,t) = S_t \exp((b-r)(T-t))N(d_1) - K\exp(-r(T-t))N(d_2) \\ \text{Delta:} & \Delta_{\text{GAC}} = \exp((b-r)(T-t))N(d_1) \\ \text{Gamma:} & \Gamma_{\text{GAC}} = \exp((b-r)(T-t))N'(d_1) = \dfrac{\exp((b-r)(T-t))}{S_t\sigma_G\sqrt{T-t}\sqrt{2\pi}}\exp\left(\dfrac{-d_1^2}{2}\right) \end{cases}$$

$$\text{Geometric Asian put:} \begin{cases} \text{Price:} & P_{\text{GAP}}(S_t,t) = K\exp(-r(T-t))N(-d_2) - S_t\exp((b-q)(T-t))N(-d_1) \\ \text{Delta:} & \Delta_{\text{GAP}} = \exp((b-r)(T-t))(N(d_1)-1) \\ \text{Gamma:} & \Gamma_{\text{GAP}} = \exp((b-q)(T-t))N'(d_1) = \dfrac{\exp((b-q)(T-t))}{S_t\sigma_G\sqrt{T-t}\sqrt{2\pi}}\exp\left(\dfrac{-d_1^2}{2}\right) \end{cases}$$

where

$$\begin{aligned} \sigma_G &= \frac{\sigma}{\sqrt{3}}, \qquad b = \frac{1}{2}\left(r - q - \frac{1}{2}\sigma_G^2\right) \\ d_1 &= \frac{1}{\sigma_G\sqrt{T-t}}\left(\log\left(\frac{S_t}{K}\right) + (b + \frac{\sigma^2}{2})(T-t)\right) \\ d_2 &= d_1 - \sigma_G\sqrt{T-t} \end{aligned}$$

5.3 Comparison to the Turnbull-Wakeman approximation

For the arithmetic Asian option, there is no closed form solution so they were priced using the Turnbull-Wakeman approximation[2] below.

$$\begin{aligned} C &\approx Se^{(b_A-r)T}N(d_1) - Ke^{-rT}N(d_2) \\ P &\approx Ke^{-rT}N(d_2) - Se^{(b_A-r)T}N(d_1) \end{aligned}$$

where b_A is the adjusted cost of carry, σ_A^2 is the adjusted variance, and

$$\begin{aligned} d_1 &= \frac{\ln(S/K) + (b_A + \sigma^2/2)T}{\sigma_A\sqrt{T}} \\ d_2 &= d_1 - \sigma_A\sqrt{T} \\ \sigma_A &= \sqrt{\frac{\ln(M_2)}{T} - 2b_A} \\ b_A &= \frac{\ln(M_1)}{T} \end{aligned}$$

where M_1 and M_2 are the first and second moments of the arithmetic average and calculated with the following.

$$\text{If } b \neq 0 \quad \begin{cases} M_1 &= \dfrac{e^{bT} - e^{bt}}{b(T-t)} \\ M_2 &= \dfrac{2e^{(2b+\sigma^2)T}}{(b+\sigma^2)(2b+\sigma^2)(T-t)^2} + \dfrac{2e^{(2b+\sigma^2)t}}{b(T-t)^2}\left[\dfrac{1}{2b+\sigma^2} - \dfrac{e^{b(T-t)}}{b+\sigma^2}\right] \end{cases}$$

$$\text{If } b = 0 \quad \begin{cases} M_1 &= 1 \\ M_2 &= \dfrac{2e^{\sigma^2 T} - 2e^{\sigma^2 t}[1 + \sigma^2(T-t)]}{\sigma^4(T-t)^2} \end{cases}$$

[1] https://en.wikipedia.org/wiki/Asian_option#European_Asian_call_and_put_options_with_geometric_averaging

[2] 1997, Espen Gaarder Haug, The Complete Guide to Option Pricing formulas

6 Results

For the simulations, we will use options with the following parameters and take 100 time steps:

$$K = £50.0, \quad T = 1.0, \quad r = 5.0\%, \quad \sigma = 25.0\%, \quad q = 2.5\%$$

and stock prices ranging from £10.0 to £100.0.

6.1 Time and error measurements

The time elapsed and maximum error was measured for all types of options in `Test_measurements.cpp`. The results were as follows.

	European call option				Arithmetic Asian call option				Geometric Asian call option			
	Time		Max error		Time		Max error		Time		Max error	
N	Euler	Exact	Euler	Exact	Euler	Exact	Euler	Exact	Euler	Exact	Euler	Exact
10	0.0221	0.0233	5.1118	5.1094	0.0135	0.0177	4.4267	4.4695	0.0112	0.0144	5.4289	5.4744
20	0.0246	0.0304	3.4712	3.5621	0.0271	0.0412	1.9499	2.0217	0.0234	0.0291	2.8269	2.9015
40	0.0408	0.0627	1.4891	1.6312	0.0516	0.0759	0.2185	0.3008	0.0472	0.0797	1.1688	1.2515
80	0.0785	0.1205	0.9008	1.0057	0.1314	0.1511	0.3365	0.2721	0.0951	0.1219	0.6377	0.7024
160	0.2337	0.3724	0.6355	0.6689	0.2066	0.3121	0.5081	0.4912	0.1805	0.2974	0.4526	0.4682
320	0.5488	0.7386	0.3947	0.3964	0.4787	0.5451	0.7103	0.7133	0.4341	0.5263	0.2736	0.2733
640	0.8519	1.2237	0.2762	0.2815	0.8162	1.1077	0.8221	0.8208	0.6978	1.0512	0.1512	0.1523
1280	1.9202	2.5559	0.1907	0.2024	1.6911	2.3249	0.8515	0.8491	1.4927	2.1006	0.1373	0.1391
2560	3.9389	4.0578	0.1487	0.1558	3.4616	4.6483	0.8803	0.8785	3.0887	4.2562	0.0955	0.0973
5120	5.5861	8.2371	0.0542	0.0567	7.1319	9.5274	0.9429	0.9458	6.1263	8.7861	0.0191	0.0166
10240	11.0085	16.1674	0.0598	0.0604	13.8388	19.0434	0.9418	0.9416	12.1017	17.2474	0.0279	0.0296
20480	23.7898	31.7962	0.0455	0.0461	28.0382	39.2913	0.9513	0.9516	24.9118	33.9007	0.0234	0.0246
40960	44.5425	70.3425	0.0461	0.0483	56.2148	75.9265	0.9526	0.9531	48.0284	70.7302	0.0226	0.0239
81920	91.2649	123.2112	0.0302	0.0314	113.0542	145.2184	0.9583	0.959	119.9241	136.9182	0.0203	0.0213
163840	189.082	306.5581	0.0184	0.0191	271.0041	295.2401	0.9608	0.9613	199.8751	264.8861	0.0158	0.0163
327680	363.271	523.3785	0.0157	0.0162	475.3839	615.3332	0.9632	0.9637	434.4481	538.9022	0.0132	0.0132

Table 1: Time and maximum error measurements for a call option

	European put option				Arithmetic Asian put option				Geometric Asian put option			
	Time		Max error		Time		Max error		Time		Max error	
N	Euler	Exact	Euler	Exact	Euler	Exact	Euler	Exact	Euler	Exact	Euler	Exact
10	0.0118	0.0137	3.7653	3.7655	0.0139	0.0203	3.4586	3.4497	0.0106	0.0168	3.7325	3.728
20	0.0197	0.0353	1.7911	1.782	0.0322	0.0423	1.5468	1.5343	0.0258	0.0401	1.855	1.8475
40	0.046	0.0702	0.9208	0.9065	0.0628	0.1157	0.6387	0.6284	0.0466	0.0636	0.9521	0.9438
80	0.0846	0.1657	0.4451	0.4346	0.1358	0.2355	0.3933	0.3709	0.1085	0.1577	0.4707	0.4642
160	0.1659	0.3026	0.2151	0.2083	0.3119	0.3663	0.4088	0.4033	0.2548	0.2995	0.3189	0.3148
320	0.4125	0.5227	0.2729	0.2708	0.5197	0.6906	0.3081	0.3072	0.4122	0.5664	0.2173	0.2182
640	0.6828	1.0174	0.1732	0.1721	1.0459	1.3736	0.2727	0.2737	0.8502	1.1671	0.1138	0.1138
1280	1.3963	2.0776	0.1032	0.1023	2.1177	2.7811	0.3046	0.3052	1.6881	2.2819	0.0964	0.0964
2560	3.1044	4.2701	0.0739	0.0737	4.3784	5.7250	0.3183	0.3189	3.3907	4.6643	0.0667	0.0672
5120	5.7166	8.7981	0.0175	0.0194	8.7917	11.0914	0.3205	0.3204	6.8152	9.6020	0.0096	0.0092
10240	11.3467	16.6787	0.0241	0.0248	16.1335	24.6046	0.3247	0.3248	16.6663	18.5968	0.0188	0.0198
20480	25.5077	40.8022	0.0249	0.0255	34.8832	46.5811	0.3256	0.3257	26.9879	37.9284	0.0171	0.0178
40960	56.7249	110.1831	0.0233	0.0249	73.8792	92.7087	0.3264	0.3265	55.6896	82.1826	0.0159	0.0167
81920	134.014	127.2842	0.0167	0.0177	115.6872	234.9428	0.3263	0.3263	115.633	171.203	0.0146	0.0153
163840	188.643	348.8599	0.0102	0.0105	332.9863	343.6927	0.3263	0.3263	237.367	316.893	0.0108	0.0109
327680	504.294	685.5742	0.0086	0.0091	590.2691	764.5718	0.3262	0.3263	531.763	774.065	0.0092	0.0087

Table 2: Time and maximum error measurements for a put option

6.2 Time complexity analysis

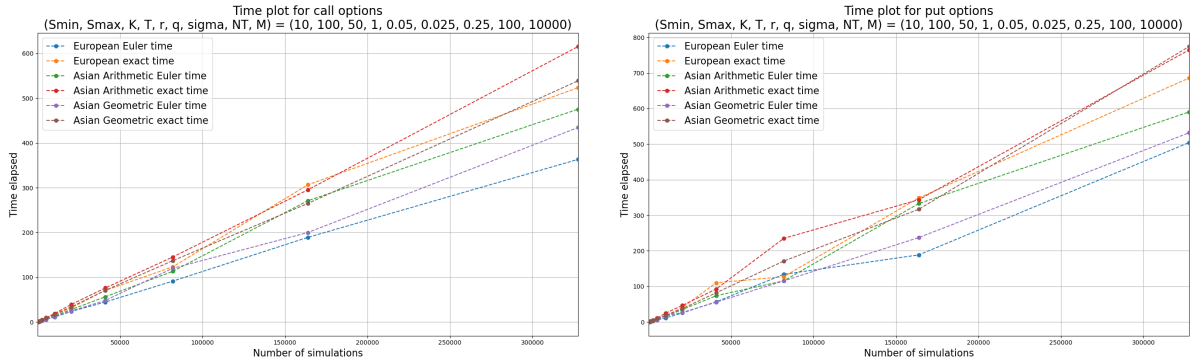
We assume that the relationship between the number of simulations for a fixed number of time steps N and time elapsed is $T(N) = \alpha N^{\beta}$. We then take the natural logarithm of both sides and fit a best line through it to find α and β. These calculations were carried out in `time_and_error_analysis.py`. The results were as follows:

	Call option	Put option
European Euler time complexity:	$0.00158 \cdot N^{0.97076}$	$0.00097 \cdot N^{1.02792}$
European exact time complexity:	$0.00198 \cdot N^{0.98399}$	$0.00153 \cdot N^{1.02204}$
Arithmetic Asian Euler time complexity:	$0.00133 \cdot N^{1.00631}$	$0.00154 \cdot N^{1.01144}$
Arithmetic Asian exact time complexity:	$0.00187 \cdot N^{0.99813}$	$0.00232 \cdot N^{1.00019}$
Geometric Asian Euler time complexity:	$0.00111 \cdot N^{1.01147}$	$0.00116 \cdot N^{1.02089}$
Geometric Asian exact time complexity:	$0.00160 \cdot N^{1.00439}$	$0.00166 \cdot N^{1.01694}$

Table 3: Time complexity comparison of between the methods

6.3 Comparison

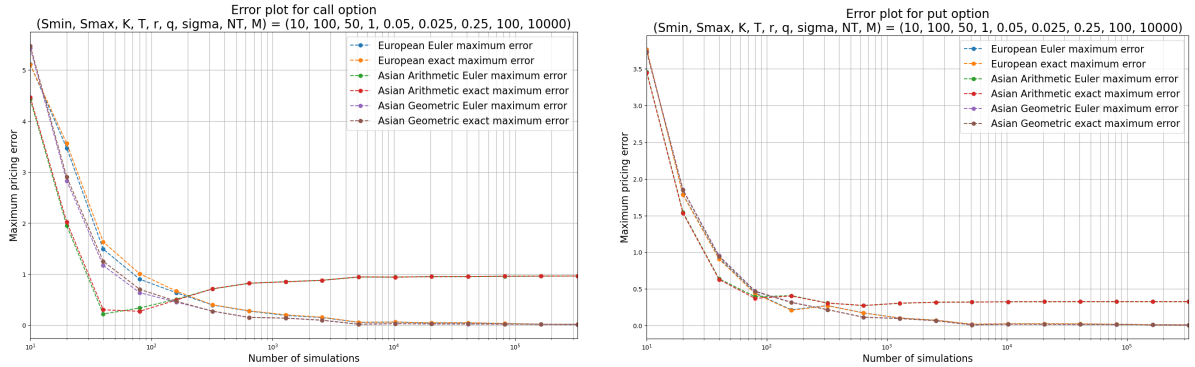
The Euler method and the exact simulation produced similarly accurate results but the Euler method was faster on average, as was expected since it required fewer steps to calculate. For both the arithmetic Asian call and put the values fail to converge to the correct price using both methods.



(a) Call option

(b) Put option

Figure 6.1: Time measurement comparison for all the methods



(a) Call option

(b) Put option

Figure 6.2: Maximum error measurement comparison for all the methods.

6.4 Problems encountered

When calculating the geometric sum of the Asian option, the standard `double` data type quickly exceeded its size limit[3] of 8 bytes when the simulations used either high stock prices, a multitude of small time steps or both. To avoid the sum going to infinity, the simulations were performed on stock prices ranging from £10 to £100 using $NT = 100$ time steps and the geometric sum was stored as a `long double`, which has bit width of 12 bytes.

	`double`	`long double`
Minimum	2.22507e-308	3.3621e-4932
Maximum	1.79769e+308	1.18973e+4932

Table 4: Comparison of the size limits of the data types `double` and `long double`

The gamma values of the arithmetic Asian option blow up, as can be seen in figures B.5 and B.6. To avoid this, it would be better to calculate the pathwise derivative of the options as mentioned earlier.

In figures 6.2(a) and (b), it is clear that the Turnbull-Wakeman approximation does not converge to the correct value so we can conclude that either the approximation is not suffeciently accurate or that the implementation was unsuccesful.

6.5 Future work

The code was written with future extensions in mind. It may be of interest to add more stochastic differential equation schemes, e.g. Milstein, and to observe more types of options, such as barrier, digital, American or even multi-asset options such as basket options.

[3] `https://www.oreilly.com/library/view/c-cookbook/0596007612/ch03s08.html`

Appendix A European call option

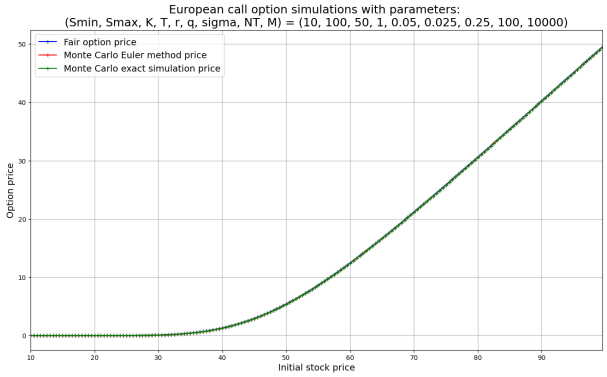


Figure A.1: European call option prices

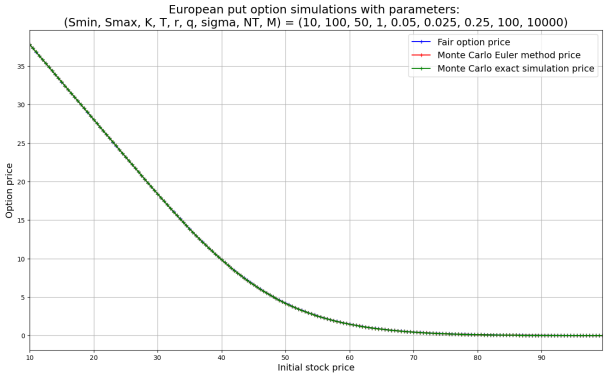


Figure A.2: European put option prices

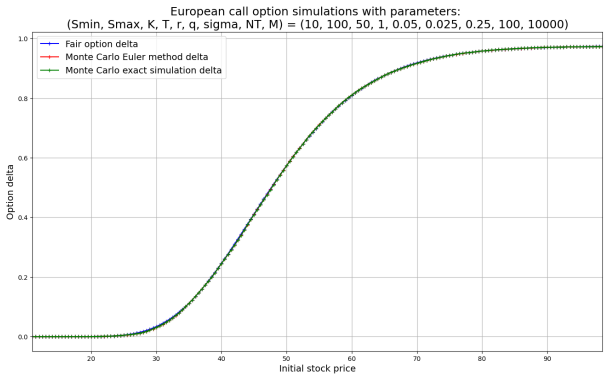


Figure A.3: European call option deltas

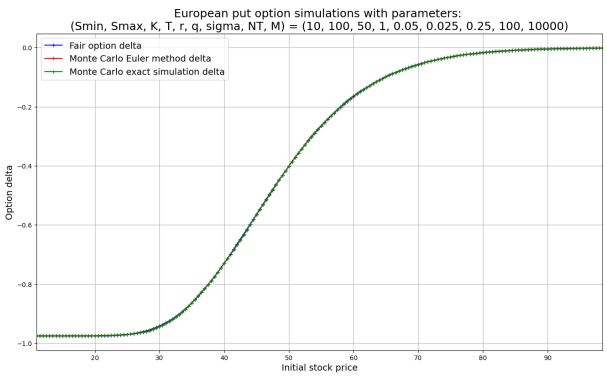


Figure A.4: European put option deltas

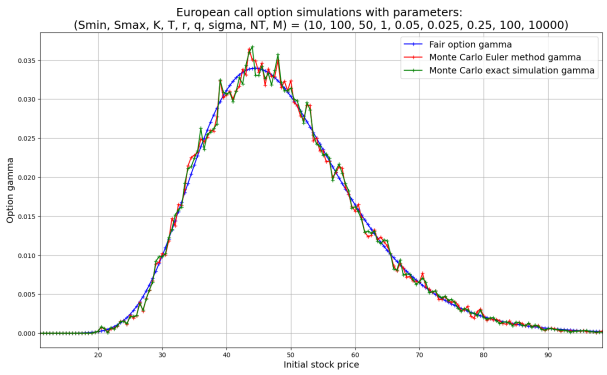


Figure A.5: European call option gammas

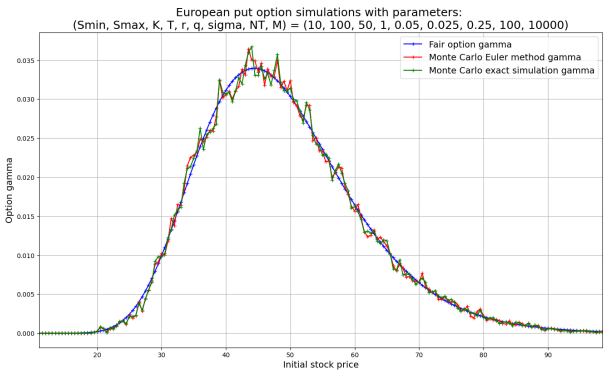


Figure A.6: European put option gammas

Appendix B Arithmetic Asian put option

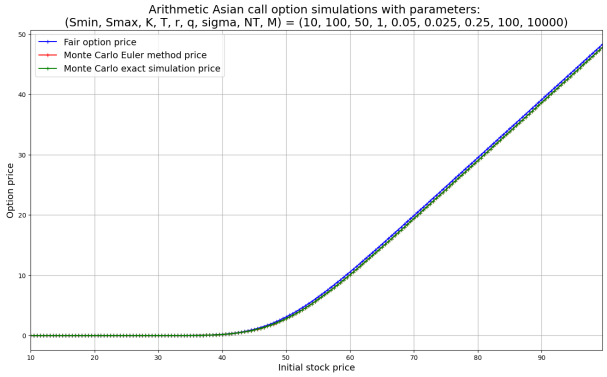


Figure B.1: Arithmetic Asian call option prices

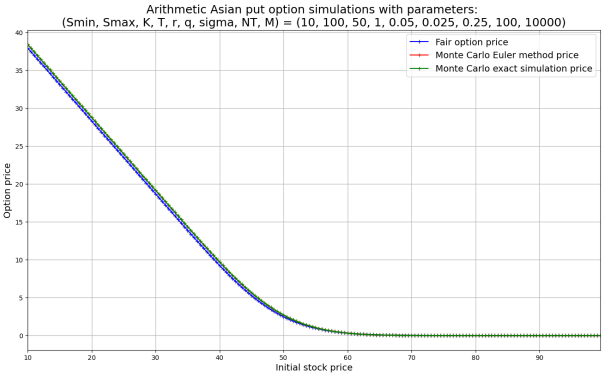


Figure B.2: Arithmetic Asian put option prices

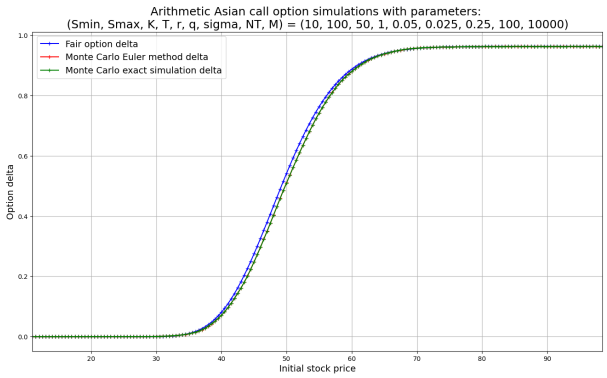


Figure B.3: Arithmetic Asian call option deltas

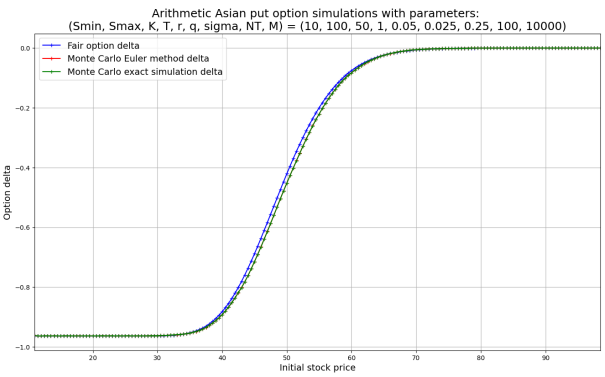


Figure B.4: Arithmetic Asian put option deltas

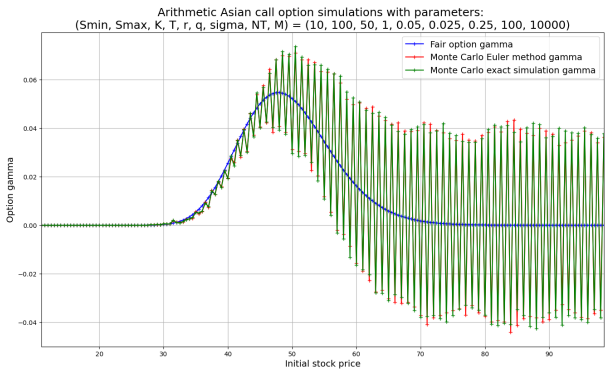


Figure B.5: Arithmetic Asian call option gammas

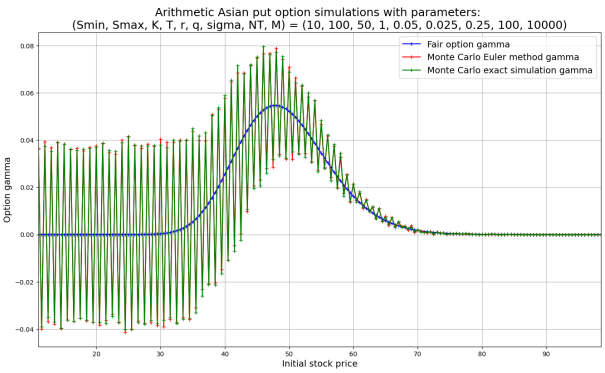


Figure B.6: Arithmetic Asian put option gammas

Appendix C Geometric Asian put option

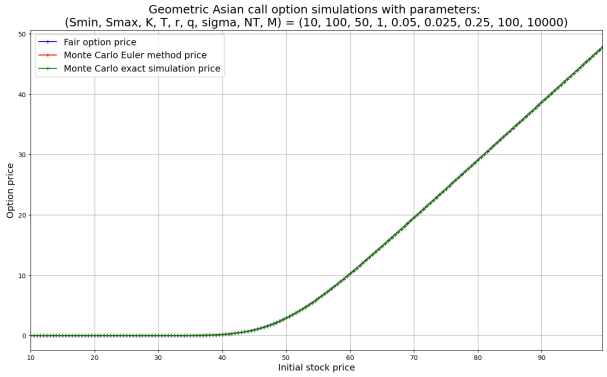


Figure C.1: Geometric Asian call option prices

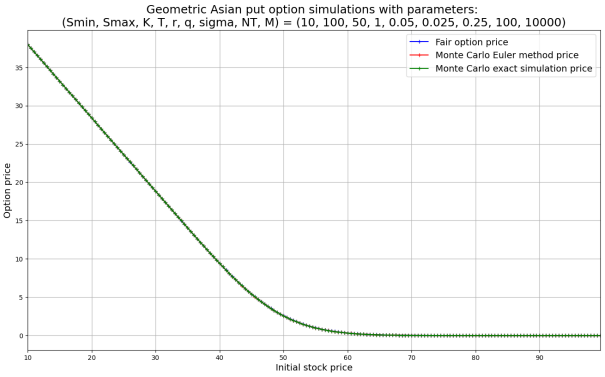


Figure C.2: Geometric Asian put option prices

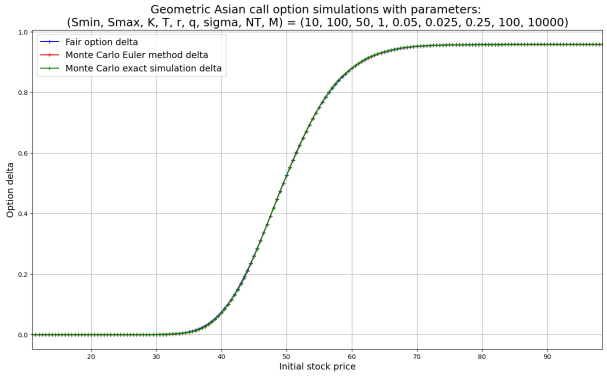


Figure C.3: Geometric Asian call option deltas

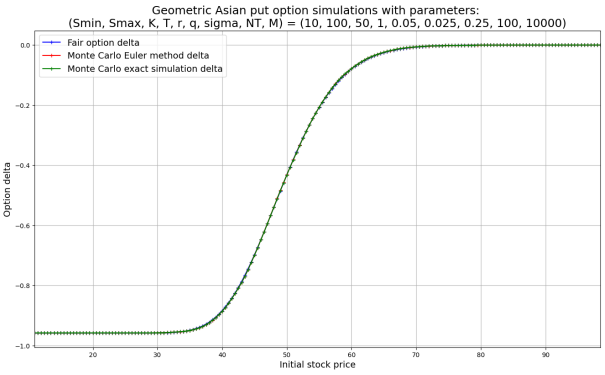


Figure C.4: Geometric Asian put option deltas

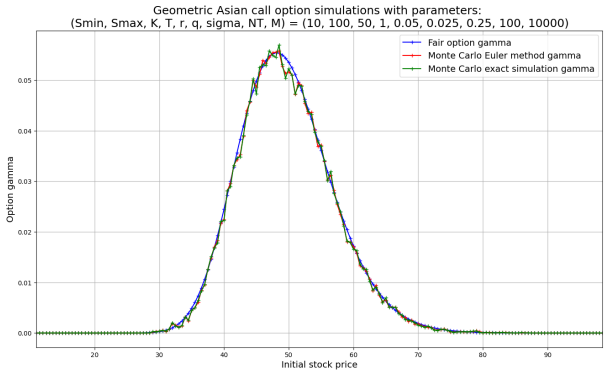


Figure C.5: Geometric Asian call option gammas

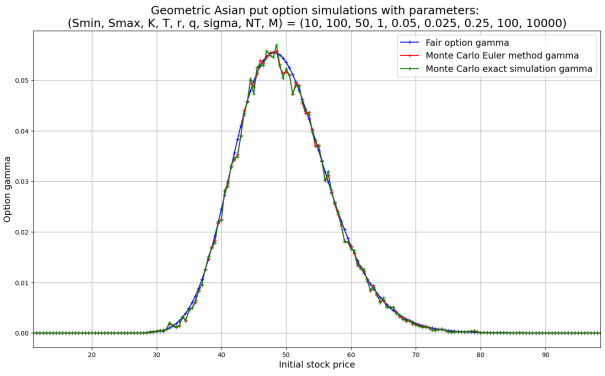


Figure C.6: Geometric Asian put option gammas

List of Tables

List of Figures