MATH7049 – Assignment 4
Joel Thomas 44793203

1.

Given dynamics:

$$dx_t = \mu(x_t, t)dt + \sigma(x_t, t)dW_t$$

$$\Delta t = \frac{T}{N}, \qquad t_n = n\Delta t, \qquad n = 0,1,\dots,N$$

Note N refers to the number of intermediate time steps and $m = 1,\dots,M$ refer to the m-th Monte Carlo sample below. Discrete approximation for x_t from SDE:

$$x_{t_{n+1}}^{(m)} = x_{t_n}^{(m)} + \mu\left(x_{t_n}^{(m)}, t_n\right)\Delta t + \sigma\left(x_{t_n}^{(m)}, t_n\right)\Delta W_{t_n}^{(m)}$$

$$\Delta W_{t_n}^{(m)} = W_{t_{n+1}}^{(m)} - W_{t_n}^{(m)} \sim N(0, t_{n+1} - t_n) = N(0, \Delta t)$$

$$\therefore \Delta W_{t_n}^{(m)} = \sqrt{\Delta t} Z_{t_n}^{(m)}, \qquad Z_{t_n}^{(m)} \sim N(0,1), m = 1,\dots,M$$

$$\rightarrow x_{t_{n+1}}^{(m)} = x_{t_n}^{(m)} + \mu\left(x_{t_n}^{(m)}, t_n\right)\Delta t + \sigma\left(x_{t_n}^{(m)}, t_n\right)\sqrt{\Delta t} Z_{t_n}^{(m)}$$

Finally, can approximate $\int_0^T \left(x_t^{(m)}\right)^2 dt$ using composite trapezoidal rule:

$$\int_0^T \left(x_t^{(m)}\right)^2 dt \approx \left(\frac{1}{2}x_{t_0}^{(m)} + \sum_{n=1}^{N-1} x_{t_n}^{(m)} + \frac{1}{2}x_{t_N}^{(m)}\right)\Delta t$$

Algorithm 1 to approximate price of financial instrument X using ordinary Monte Carlo with path simulation (Euler timestepping). Parameters $x_0, K, \mu(x_t, t), \sigma(x_t, t), T, M, N$ given.	
Step	**Task**
1	set $\Delta t = \frac{T}{N}$
2	for $m = 1,\dots,M$ {loop over all MC samples;}
3	for $n = 0,\dots,N-1$ {perform Euler timestepping on m-th sample;} compute Draw $Z_n^{(m)} \sim N(0,1)$ $\hat{x}_{t_{n+1}}^{(m)} = \hat{x}_{t_n}^{(m)} + \mu(\hat{x}_{t_n}^{(m)}, t_n)\Delta t + \sigma(\hat{x}_{t_n}^{(m)}, t_n)\sqrt{\Delta t} Z_{t_n}^{(m)}$ end inner for
4	compute and store $\int_0^T \left(\hat{x}_t^{(m)}\right)^2 dt = \left(\frac{1}{2}\hat{x}_{t_0}^{(m)} + \sum_{n=1}^{N-1} \hat{x}_{t_n}^{(m)} + \frac{1}{2}\hat{x}_{t_N}^{(m)}\right)\Delta t$ {approximate $\int_0^T \left(\hat{x}_t^{(m)}\right)^2 dt$;} end outer for
5	output $\hat{X}_M = \frac{1}{M}\sum_{m=1}^{M}\left(\int_0^T \left(\hat{x}_t^{(m)}\right)^2 dt - K\right)^+$ {ordinary MC estimate;}

Table 1: Algorithm 1 to price financial instrument X

2.

a)

Given dynamics:

$$dX_t = \kappa(\alpha - X_t)dt + \sigma dW_t$$

$$\Delta t = \frac{T_1}{N}, \qquad t_n = n\Delta t, \qquad n = 0,1,\dots,N$$

Use T_1 when calculating Δt because only need to simulate X_t up to time T_1. Note N refers to the number of intermediate time steps and $m = 1,\dots,M$ refer to the m-th Monte Carlo sample below. Discrete approximation for X_t from SDE:

$$X_{t_{n+1}}^{(m)} = X_{t_n}^{(m)} + \kappa\left(\alpha - X_{t_n}^{(m)}\right)\Delta t + \sigma \Delta W_{t_n}^{(m)}$$

$$\Delta W_{t_n}^{(m)} = W_{t_{n+1}}^{(m)} - W_{t_n}^{(m)} \sim N(0, t_{n+1} - t_n) = N(0, \Delta t)$$

$$\therefore \Delta W_{t_n}^{(m)} = \sqrt{\Delta t} Z_{t_n}^{(m)}, \qquad Z_{t_n}^{(m)} \sim N(0,1), m = 1,\dots,M$$

$$\rightarrow X_{t_{n+1}}^{(m)} = X_{t_n}^{(m)} + \kappa\left(\alpha - X_{t_n}^{(m)}\right)\Delta t + \sigma\sqrt{\Delta t} Z_{t_n}^{(m)}$$

After N intermediate steps:

$$S_{T_1}^{(m)} = e^{X_{T_1}^{(m)}} = e^{X_{t_N}^{(m)}}$$

Using (1):

$$F_{T_1}^{(m)} = e^{e^{-\kappa(T_2-T_1)}\log\left(S_{T_1}^{(m)}\right) + \left(1-e^{-\kappa(T_2-T_1)}\right)\alpha + \frac{\sigma^2}{4\kappa}\left(1-e^{-2\kappa(T_2-T_1)}\right)}$$

$$\therefore Y_m = e^{-rT_1}\left(F_{T_1}^{(m)} - K_1\right)^+$$

Full algorithm displayed in Table 2 on next page.

Algorithm 2 to approximate price $C(S_0, t_0)$ using ordinary Monte Carlo with path simulation (Euler timestepping). Parameters $S_0, K_1, r, \sigma, T_1, T_2, \kappa, \alpha, N$ given.	
Step	**Task**
1	set $\Delta t = \frac{T_1}{N}, X_0 = \log(S_0)$
2	estimate true M from pilot computation
3	for $m = 1, \dots, M$ {loop over all MC samples;}
4	for $n = 0, \dots, N-1$ {perform Euler timestepping on m-th sample;} compute Draw $Z_n^{(m)} \sim N(0,1)$ $\hat{X}_{t_{n+1}}^{(m)} = \hat{X}_{t_n}^{(m)} + \kappa\left(\alpha - \hat{X}_{t_n}^{(m)}\right)\Delta t + \sigma\sqrt{\Delta t}Z_{t_n}^{(m)}$ end inner for
5	compute $\hat{S}_{T_1}^{(m)} = e^{\hat{X}_{t_N}^{(m)}}$ $\hat{F}_{T_1}^{(m)} = e^{e^{-\kappa(T_2-T_1)}\log\left(\hat{S}_{T_1}^{(m)}\right)+\left(1-e^{-\kappa(T_2-T_1)}\right)\alpha+\frac{\sigma^2}{4\kappa}\left(1-e^{-2\kappa(T_2-T_1)}\right)}$ {simulate forward price at time T_1;} store $Y_m = e^{-rT_1}\left(\hat{F}_{T_1}^{(m)} - K_1\right)^+$ {discounted payoff at time 0;} end outer for
6	output $\hat{C}_M = \frac{1}{M}\sum_{m=1}^{M} Y_m$ {ordinary MC estimate;}

Table 2: Algorithm 2 to approximate $C(S_0, t_0)$

b)

In order to estimate how large M should be, require:

$$\frac{\hat{\sigma}_M}{\sqrt{M}} \leq 0.05$$

$$\rightarrow M = \left\lceil \left(\frac{\hat{\sigma}_M}{0.05}\right)^2 \right\rceil$$

This further requires we do a pilot computation, similar to as in Assignment 3, to get an initial estimate for $\hat{\sigma}_M$, the sample standard deviation of $Y\ \forall m = 1, \dots, M$. In the code, $M = 1e3$ was chosen as the initial guess.

Price	Standard Error	Radius	Confidence Interval
1.632044	0.046765	0.091658	[1.540386, 1.723701]

Table 3: Results from implementing algorithm proposed in a) to estimate $C(S_0, t_0)$

c)

Limit definitions of first (delta) and second (gamma) derivatives for an option:

$$\frac{\partial C}{\partial S} = \lim_{\Delta S \to 0} \frac{C(S + \Delta S, t) - C(S, t)}{\Delta S}$$

$$\frac{\partial^2 C}{\partial S^2} = \lim_{\Delta S \to 0} \frac{C(S + \Delta S, t) - 2C(S, t) + C(S - \Delta S, t)}{(\Delta S)^2}$$

In this question, we can approximate $\frac{\partial C}{\partial S}$ and $\frac{\partial^2 C}{\partial S^2}$ using $\Delta S = 0.001$. We can do this via Euler timestepping for $C(S+\Delta S, t)$ and $C(S-\Delta S, t)$ in addition to $C(S,t)$ as described in the modified algorithm 2 below. In order to estimate how large M should be again, repeat the same pilot computation from b).

Modified algorithm 2 to approximate price $C(S_0, t_0), \frac{\partial C}{\partial S}, \frac{\partial^2 C}{\partial S^2}$ using ordinary Monte Carlo with path simulation (Euler timestepping). Parameters $S_0, K_1, r, \sigma, T_1, T_2, \kappa, \alpha, N$ given.

Step	Task
1	set $\Delta t = \frac{T_1}{N}, \Delta S = 0.001, X_0 = \log(S_0), X_0^{(\Delta S)} = \log(S_0 + \Delta S), X_0^{(-\Delta S)} = \log(S_0 - \Delta S)$
2	estimate true M from pilot computation
3	for $m = 1, \dots, M$ {loop over all MC samples;}
4	for $n = 0, \dots, N-1$ {perform Euler timestepping on m-th sample;} compute Draw $Z_n^{(m)} \sim N(0,1)$ $\hat{X}_{t_{n+1}}^{(m)} = \hat{X}_{t_n}^{(m)} + \kappa\left(\alpha - \hat{X}_{t_n}^{(m)}\right)\Delta t + \sigma\sqrt{\Delta t} Z_{t_n}^{(m)}$ $\hat{X}_{t_{n+1}}^{(\Delta S, m)} = \hat{X}_{t_n}^{(\Delta S, m)} + \kappa\left(\alpha - \hat{X}_{t_n}^{(\Delta S, m)}\right)\Delta t + \sigma\sqrt{\Delta t} Z_{t_n}^{(m)}$ $\hat{X}_{t_{n+1}}^{(-\Delta S, m)} = \hat{X}_{t_n}^{(-\Delta S, m)} + \kappa\left(\alpha - \hat{X}_{t_n}^{(-\Delta S, m)}\right)\Delta t + \sigma\sqrt{\Delta t} Z_{t_n}^{(m)}$ end inner for
5	compute $\hat{S}_{T_1}^{(m)} = e^{\hat{X}_{t_N}^{(m)}}$ $\hat{F}_{T_1}^{(m)} = e^{e^{-\kappa(T_2-T_1)} \log\left(\hat{S}_{T_1}^{(m)}\right) + \left(1-e^{-\kappa(T_2-T_1)}\right)\alpha + \frac{\sigma^2}{4\kappa}\left(1-e^{-2\kappa(T_2-T_1)}\right)}$ $\hat{S}_{T_1}^{(\Delta S, m)} = e^{\hat{X}_{t_N}^{(\Delta S, m)}}$ $\hat{F}_{T_1}^{(\Delta S, m)} = e^{e^{-\kappa(T_2-T_1)} \log\left(\hat{S}_{T_1}^{(\Delta S, m)}\right) + \left(1-e^{-\kappa(T_2-T_1)}\right)\alpha + \frac{\sigma^2}{4\kappa}\left(1-e^{-2\kappa(T_2-T_1)}\right)}$ $\hat{S}_{T_1}^{(-\Delta S, m)} = e^{\hat{X}_{t_N}^{(-\Delta S, m)}}$ $\hat{F}_{T_1}^{(-\Delta S, m)} = e^{e^{-\kappa(T_2-T_1)} \log\left(\hat{S}_{T_1}^{(-\Delta S, m)}\right) + \left(1-e^{-\kappa(T_2-T_1)}\right)\alpha + \frac{\sigma^2}{4\kappa}\left(1-e^{-2\kappa(T_2-T_1)}\right)}$ {simulate forward price at time T_1;} store $Y_m = e^{-rT_1}\left(\hat{F}_{T_1}^{(m)} - K_1\right)^+$ $Y_m^{(\Delta S)} = e^{-rT_1}\left(\hat{F}_{T_1}^{(\Delta S, m)} - K_1\right)^+$ $Y_m^{(-\Delta S)} = e^{-rT_1}\left(\hat{F}_{T_1}^{(-\Delta S, m)} - K_1\right)^+$ {discounted payoff at time 0;} $\frac{\partial \hat{C}_m}{\partial \hat{S}^{(m)}} = \frac{Y_m^{(\Delta S)} - Y_m}{\Delta S}$ $\frac{\partial^2 \hat{C}_m}{\partial \left(\hat{S}^{(m)}\right)^2} = \frac{Y_m^{(\Delta S)} - 2Y_m + Y_m^{(-\Delta S)}}{(\Delta S)^2}$ {approximate delta and gamma;} end outer for
6	output $\hat{C}_M = \frac{1}{M}\sum_{m=1}^{M} Y_m, \quad \frac{\partial C}{\partial S} = \frac{1}{M}\sum_{m=1}^{M} \frac{\partial \hat{C}_m}{\partial \hat{S}^{(m)}}, \quad \frac{\partial^2 C}{\partial S^2} = \frac{1}{M}\sum_{m=1}^{M} \frac{\partial^2 \hat{C}_m}{\partial \left(\hat{S}^{(m)}\right)^2}$ {ordinary MC estimates;}

Table 4: Modified algorithm 2 to approximate $C(S_0, t_0), \frac{\partial C}{\partial S}, \frac{\partial^2 C}{\partial S^2}$

Essentially, in table 4, for each MC replication, we simulate three different paths to time T_1 – one starting from S_0, one starting from $S_0 + 0.001$ and one starting from $S_0 - 0.001$. Given these three different S_{T_1}, we can obtain three different forward prices and hence three different option prices - $C(S_0, t_0)$, $C(S_0 + 0.001, t_0)$ and $C(S_0 - 0.001, t_0)$. From this, we can approximate $\frac{\partial C}{\partial S}$ and $\frac{\partial^2 C}{\partial S^2}$ as required.

	Value	Standard Error	Radius	Confidence Interval
$C(S_0, t_0)$	1.604256	0.048693	0.095437	[1.508819, 1.699693]
$\frac{\partial C}{\partial S}$	0.271777	0.005560	0.010898	[0.260879, 0.282675]
$\frac{\partial^2 C}{\partial S^2}$	-0.003803	0.000078	0.000153	[-0.003956, -0.003651]

Table 5: Results from implementing algorithm proposed on previous page to estimate $C(S_0, t_0)$

3.

Let:

$C =$ expected discounted payoff of an European option whose underlying has CEV dynamics.

$Y =$ discounted payoff of an European option whose underlying has CEV dynamics.

$C^* =$ expected discounted payoff of an European option whose underlying has GBM dynamics.

$Y^* =$ discounted payoff of an European option whose underlying has GBM dynamics.

We can simulate the SDE (2) via Euler timestepping by discretising for S_t similar as in questions 1 and 2. For S^*, we can use the closed form solution of a standard GBM to directly calculate the time T price:

$$S_{t_{n+1}}^{(*,m)} = S_{t_n}^{(*,m)} e^{\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} Z_{t_n}^{(m)}}$$

Manually calculated a few iterations:

$$S_{t_1}^{(*,m)} = S_{t_0}^{(*,m)} e^{\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} Z_{t_0}^{(m)}}$$

$$S_{t_1}^{(*,m)} = S_0 e^{\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} Z_{t_0}^{(m)}}$$

$$S_{t_2}^{(*,m)} = S_{t_1}^{(*,m)} e^{\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} Z_{t_1}^{(m)}}$$

$$S_{t_2}^{(*,m)} = S_0 e^{\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} Z_{t_0}^{(m)}} \cdot e^{\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} Z_{t_1}^{(m)}}$$

$$S_{t_2}^{(*,m)} = S_0 e^{2\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} \sum_{n=1}^{2} Z_{t_n}^{(m)}}$$

$$\therefore S_{t_N}^{(*,m)} = S_0 e^{N\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} \sum_{n=0}^{N-1} Z_{t_n}^{(m)}}$$

In this way, we can directly simulate the final price $S_T^{(*,m)}$, using the N normal random draws for the m-th path, in a way that produces the correct distribution of $S_T^{(*,m)}$.

Algorithm 3 to approximate price $C(S_0, t_0)$ for underlying that has CEV dynamics using ordinary Monte Carlo with a control variate and with path simulation (Euler timestepping). Parameters $S_0, K, r, \sigma, T, \alpha, M, N$ given.	
Step	**Task**
1	set $\Delta t = \frac{T}{N}, \sigma^* = \sigma S_0^{\alpha-1}$
2	for $m = 1, \ldots, M$ {loop over all MC samples;}
3	for $n = 0, \ldots, N-1$ {perform Euler timestepping on m-th sample;} compute Draw $Z_n^{(m)} \sim N(0,1)$ $\hat{S}_{t_{n+1}}^{(m)} = \hat{S}_{t_n}^{(m)} \left(1 + r\Delta t + \sigma\left(\hat{S}_{t_n}^{(m)}\right)^{\alpha-1} \sqrt{\Delta t} Z_{t_n}^{(m)}\right)$ {simulate time T price of underlying with CEV dynamics;} end inner for
4	Compute $S_T^{(*,m)} = S_0 e^{N\left(r - \frac{(\sigma^*)^2}{2}\right)\Delta t + \sigma^* \sqrt{\Delta t} \sum_{n=0}^{N-1} Z_{t_n}^{(m)}}$ {simulate time T price of underlying with GBM dynamics;} store $Y_m = e^{-rT}\left(K - \hat{S}_T^{(m)}\right)^+$ $Y_m^* = e^{-rT}\left(K - \hat{S}_T^{(*,m)}\right)^+$ {discounted payoffs at time 0;} end outer for
5	compute $\hat{\beta} = \frac{cov(Y, Y^*)}{var(Y^*)} = \frac{\sigma_{Y,Y^*}}{\sigma^2{}_Y^*}$ {optimal coefficient;}
6	output: $\hat{C}_M = \frac{1}{M} \sum_{m=1}^{M} Y_m$ {ordinary MC estimate;} $\bar{Y}_M = \frac{1}{M} \sum_{m=1}^{M} Y_m$ $C^* = C^{BS}\left(S_0, K, T, R_{grow}, R_{disc}, \sigma\right) = C^{BS}(S_0, K, T, r, r, \sigma)$ $\bar{Y}_M^* = \frac{1}{M} \sum_{m=1}^{M} Y_m^*$ $\hat{C}_M^{cv,\hat{\beta}} = \bar{Y}_M + \beta(C^* - \bar{Y}_M^*)$ {MC estimate using a control variate;}

Table 6: Algorithm 3 to approximate $C(S_0, t_0)$ using ordinary MC and MC with a control variate

	Value	Standard Error	Radius	Confidence Interval
$\hat{C}_M$	3.275621	0.087117	0.170746	[3.104875, 3.446366]
$\hat{C}_M^{cv,\hat{\beta}}$	3.286091	0.003679	0.007210	[3.278881, 3.293301]

Table 7: Results from implementing algorithm proposed above to estimate $C(S_0, t_0)$

From lecture slides, we know that a control variate Y^* is a good choice for estimating $\hat{C}_M^{cv,\hat{\beta}}$ provided $cov(Y, Y^*) > \frac{1}{2} var(Y^*)$ i.e. Y^* and Y are highly correlated. A good choice here yields a large reduction in variance in the estimate of the true option price. From MATLAB, simply using the $corr$ correlation function on Y, Y^* results in $corr(Y < Y^*) > 0.99$ on every instance that the code is run. Thus, this explains why we can be so much more confident in the estimate $\hat{C}_M^{cv,\hat{\beta}}$ (tighter confidence interval) than $\hat{C}_M$ after using the control variate Monte Carlo technique.

4.

a)

Finding $g_{S_T}(s)$:

$$g_{S_T}(s) = c \cdot h(s) f_{S_T}(s)$$

$$h(s) = e^{-rT}(s-K)^+$$

$$c = \frac{1}{\int_R h(s) f_{S_T}(s) ds}$$

where $R =$ a region such that $h(s) \cdot f_{S_T}(s) > 0$

$$\therefore\ g_{S_T}(s) = \frac{h(s) f_{S_T}(s)}{\int_R h(s) f_{S_T}(s) ds}$$

$$g_{S_T}(s) = \frac{e^{-rT}(s-K)^+ f_{S_T}(s)}{e^{-rT}\int_R (s-K)^+ f_{S_T}(s) ds}$$

$$g_{S_T}(s) = \frac{e^{-rT}(s-K)^+ f_{S_T}(s)}{C_0}, \qquad C_0 = C^{BS}\left(S_0, K, T, R_{grow} = r, R_{disc} = r, \sigma, call\right)$$

b)

$$\mathbb{E}_{g_{S_T}}[S_T] = \int_R s \cdot g_{S_T}(s)\, ds$$

where $R =$ a region such that $s \cdot g_{S_T}(s) > 0$. Substituting in the expression for $g_{S_T}(s)$ from a):

$$\mathbb{E}_{g_{S_T}}[S_T] = \int_R s \cdot \frac{e^{-rT}(s-K)^+ f_{S_T}(s)}{C_0}\, ds$$

$$\mathbb{E}_{g_{S_T}}[S_T] = \frac{1}{C_0} \cdot e^{-rT} \int_R s(s-K)^+ f_{S_T}(s)\, ds$$

$$\mathbb{E}_{g_{S_T}}[S_T] = \frac{1}{C_0} \cdot \mathbb{E}_{f_{S_T}}[e^{-rT} S_T (S_T - K)^+]$$

$$\mathbb{E}_{g_{S_T}}[S_T] = \frac{\tilde{C}_0}{C_0}, \qquad \tilde{C}_0 = C^{BS}\left(S_0, K, T, R_{grow} = r + \sigma^2, R_{disc} = 0, \sigma, call\right)$$

where the last line follows from the hint provided for the question.

c)

Modified BM:

$$W_t = W_t^* + \lambda t, \qquad \lambda > 0$$

$$\therefore\ dW_t = d(W_t^* + \lambda t) = dW_t^* + \lambda dt$$

Modified GBM:

$$dS_t = rS_t dt + \sigma S_t dW_t$$

$$dS_t = rS_t dt + \sigma S_t dW_t^* + \sigma\lambda S_t dt$$

$$dS_t = (r + \sigma\lambda)S_t dt + \sigma S_t dW_t^*$$

So we are essentially simulating another GBM that has changed drift but similar diffusion coefficients to the original GBM. The closed form solution for the original GBM is given by:

$$S_T = S_0 e^{\left(r-\frac{\sigma^2}{2}\right)T+\sigma W_T}, \qquad W_T \sim N(0,T)$$

$$W_T = \sqrt{T}Z, \qquad Z \sim N(0,1)$$

$$\therefore S_T = S_0 e^{\left(r-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}Z}$$

The closed form solution for the modified GBM is thus:

$$S_T = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T+\sigma W_T} = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}Z}$$

Finding λ:

$$\mathbb{E}_{g_{S_T}}[S_T] = \mathbb{E}^*[S_T]$$

$$\mathbb{E}_{g_{S_T}}[S_T] = \mathbb{E}^*\left[S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}Z}\right]$$

$$\mathbb{E}_{g_{S_T}}[S_T] = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T}\,\mathbb{E}^*\left[e^{\sigma\sqrt{T}Z}\right]$$

Using the expression from b) for $\mathbb{E}_{g_{S_T}}[S_T]$ and evaluating the expectation of $e^{\sigma\sqrt{T}Z}$:

$$\rightarrow \frac{\tilde{C}_0}{C_0} = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T} \int_{-\infty}^{\infty} e^{\sigma\sqrt{T}z} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}\, dz$$

$$\frac{\tilde{C}_0}{C_0} = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2-2\sigma\sqrt{T}z}{2}}\, dz$$

Completing the square for z inside the exponential inside the integral:

$$\frac{\tilde{C}_0}{C_0} = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2-2\sigma\sqrt{T}z+\sigma^2T-\sigma^2T}{2}}\, dz$$

$$\frac{\tilde{C}_0}{C_0} = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T} \cdot e^{\frac{\sigma^2T}{2}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(z-\sigma\sqrt{T})}{2}}\, dz$$

Note that the expression inside the integral is just the PDF of a normal random variable $Z \sim N(\mu, \sigma^2) = N(\sigma\sqrt{T}, 1)$. Since we are calculating the CDF using the integral over the entire possible range, the area is simply equal to 1 i.e.:

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{(z-\sigma\sqrt{T})}{2}}\, dz = 1$$

$$\therefore \frac{\tilde{C}_0}{C_0} = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T+\frac{\sigma^2T}{2}} \cdot 1$$

$$\frac{\tilde{C}_0}{C_0} = S_0 e^{(r+\sigma\lambda)T}$$

$$(r+\sigma\lambda)T = \ln\left(\frac{\tilde{C}_0}{S_0 \cdot C_0}\right)$$

$$\sigma\lambda = \frac{\ln\left(\frac{\tilde{C}_0}{S_0 C_0}\right)}{T} - r$$

$$\lambda = \frac{\frac{\ln\left(\frac{\tilde{C}_0}{S_0 C_0}\right)}{T} - r}{\sigma}$$

$$\therefore \lambda = \frac{\ln\left(\frac{\tilde{C}_0}{S_0 C_0}\right) - rT}{\sigma T}$$

d)

Algorithm 4 to approximate price $C(S_0, t_0)$ using an importance sampling Monte Carlo algorithm. Parameters $S_0, K, r, \sigma, T, M, N$ given.	
Step	**Task**
1	set $C_0 = C^{BS}(S_0, K, T, R_{grow} = r, R_{disc} = r, \sigma, call), \tilde{C}_0 = C^{BS}(S_0, K, T, R_{grow} = r + \sigma^2, R_{disc} = 0, \sigma, call)$, $\lambda = \frac{\ln\left(\frac{\tilde{C}_0}{S_0 C_0}\right) - rT}{\sigma T}$
2	for $m = 1, \dots, M$ {loop over all MC samples;} compute Draw $Z^{(m)} \sim N(0,1)$ $\hat{S}_T^{(m)} = S_0 e^{\left(r+\sigma\lambda-\frac{\sigma^2}{2}\right)T+\sigma\sqrt{T}Z^{(m)}}$ {simulate time T price of underlying directly using closed form solution to GBM;} store $Y_m = e^{-rT}(S_T^m - K)^+$ {discounted payoffs at time 0;} end for
5	output $\hat{C}_M^{is} = \frac{1}{M}\sum_{m=1}^{M} Y_m$ {MC estimate using importance sampling;}

Table 8: Algorithm 4 to approximate $C(S_0, t_0)$ using an importance sampling MC algorithm

e)

Price	Standard Error	Radius	Confidence Interval
0.000109	0.000056	0.000109	[-0.000001, 0.000218]

Table 9: Results from implementing algorithm proposed in a) to estimate $C(S_0, t_0)$

From table 9, it is evident that the MC estimate based on using importance sampling is very close to the true option price given by $C^{BS}(S_0, K, T, R_{grow} = r, R_{disc} = r, \sigma, call) = 0.00010624$. The reason why the standard error is very small but non-zero is because the standard deviation of the estimate is non-zero. This is expected since from c), we are choosing λ such that the expectations of S_T under $\mathbb{P}^*$ and g_{S_T} agree but not their distributions.