1. (a) Write a Matlab function to compute

$$\int_0^1 \frac{1}{\sqrt{1+\sin^2(t)}}\,\mathrm{d}t \tag{1}$$

using the Monte Carlo method with N samples. Your function should compute both the integral and an estimate of the error. [**40%**]

Answer:

```
function [answer, sd] = computeIntegral(N)
t = rand(1,N);
vals = sqrt( 1 + (sin( t ).^2) );
sd = 1/sqrt(N)*std( vals );
answer = mean(vals);
end
```

The first return value `answer` gives an unbiased estimate for the integral. By the central limit theorem, this value will be approximately normally distributed with a standard deviation given by the second return value `sd`. This allows one to construct confidence intervals for the answer. For example, answer $\pm\, 1.96$ sd would give a 95% confidence interval.

Notes: I am looking for it being a function, to have two return values and a parameter N. In the code, I am looking for the use of `.^` or a for loop, correct formulae and an explanation that the return value is the estimate and the standard error.

(b) Describe a better numerical method to compute the integral given in equation (1). Justify your answer. [**20%**]

Answer: The rectangle rule would be better. Simply take evenly spaced points instead of uniformly spaced points. The standard error of Monte Carlo converges at rate $O(N^{-1/2})$, the error of rectangle rule converges at rate $O(N^{-2})$.

Notes: this was all I was looking for. You might want to remark that the convergence of Monte Carlo follows from the Central Limit Theorem, while the convergence of the rectangle rule can be proved using Taylor's theorem so long as one has a bound on the third derivative of the integrand.

(c) Name a technique you can use to improve the accuracy of the Monte Carlo methods and describe briefly how you would apply it to this problem. [**20%**]

Answer: You could use the control variate technique. A good choice of control variate would be $\int_0^1 \frac{1}{\sqrt{1+t^2}}\,\mathrm{d}t = \sinh^{-1}(1)$. This would be a good choice of control variate since for small x $\sin(x) \approx x$, so one would expect the required integrand and the integrand of the control variate to be highly correlated.

The paragraph above was all I was actually looking for as the challenge is coming up with a good control variate. In other words the phrase "how you would apply it to this problem" is key to what I was looking for. Many students described the control variate method in general rather than how to apply it to this problem. I gave partial credit for doing this. Here is some extra detail on how to actually implement the control variate method in this case.

Let U be uniformly distributed on $[0,1]$. Let X be the random variable

$$\frac{1}{\sqrt{1+\sin(U)^2}}$$

and let Y be the random variable

$$\frac{1}{\sqrt{1+U^2}}.$$

We wish to estimate $E(X)$ and we know that $E(Y) = \sinh^{-1}(1)$. We generate N samples of U and hence obtain N samples for X and Y.

For any λ, the expectation of $X + \lambda(Y - \sinh^{-1}(1))$ will give us an unbiased estimate of $E(X)$. The optimal choice of λ is given by the formula:

$$\lambda = -\frac{\mathit{Cov}(X,Y)}{\mathit{Var}Y}.$$

We use our samples of X and Y to estimate the covariance and variance in this formula and hence approximate the optimal λ. The sample mean of $X + \lambda(Y - \sinh^{-1}(1))$ then gives us our control-variate estimate of $E(X)$.

(d) What one dimensional integral do you need to compute the price of a call option in the Black–Scholes model? **[20%]**

Answer: The price of a call option is given by:

$$\int_{-\infty}^{\infty} \exp(-rT)\max\{\exp(s) - K, 0\} q(s)\, \mathrm{d}s$$

where $q(s)$ is the p.d.f. of the log of the stock price in the risk neutral measure. In detail

$$q(s) = \frac{1}{\sigma\sqrt{2\pi T}} \exp(-(s - (\log(S_0) + (r - \frac{1}{2}\sigma^2)T))^2/(2\sigma^2 T))$$

Note: I've written the answer in two pieces to make it easy to give me marks for the central idea even if I haven't quite got the formula for q correct. I've *written the integrand in terms of the log of the stock price rather than the stock price because it is easier to compute the pdf of s than it is to compute the pdf of S. In fact, since I know that $s = \log S$ follows the process*

$$\mathrm{d}s = (r - \frac{1}{2}\sigma^2)\mathrm{d}t + \sigma \mathrm{d}W_t$$

I know that s_T is normally distributed with mean $s_0 + (r - \frac{1}{2}\sigma^2)T$ and standard deviation $\sigma\sqrt{T}$. So I can just write down $q(s)$.

2. (a) A stock price S_t follows the stochastic process given by:

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

where μ and σ are constants and W_t is a Wiener process. Find a function $f(S,t)$ such that $f(S_t,t)$ follows a Brownian motion with drift 0 and volatility σ. **[20%]**

Answer: By Itô's lemma, we know that $s = \log(S)$ obeys the SDE

$$\mathrm{d}s = (\mu - \frac{1}{2}\sigma^2)\mathrm{d}t + \sigma \mathrm{d}W_t$$

Hence if we let

$$f(S,t) = \log(S) - (\mu - \frac{1}{2}\sigma^2)t$$

then $f(S,t)$ will follow Brownian motion with drift 0 and volatility σ.

(b) The Black–Scholes PDE is:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

What change of variables would you use to transform this to the heat equation. Justify your answer. **[20%]**

Answer: The change of variables to use is:

$$x = \log(S) - (r - \frac{1}{2}\sigma^2)t$$

$$W = \exp^{-rt} V$$

$$\tau = -t$$

Given a payoff function f, by Feynman-Kac, the Black-Scholes PDE is the PDE obeyed by

$$V(S,t) = E(e^{-r(t-T)} f(S) | S_t = S)$$

where S obeys geometric Brownian motion with drift r and standard deviation σ. By the first part, under our first two changes of variables we obtain:

$$W(x,t) = E(f(x)|x_t = x)$$

where x obeys driftless Brownian motion with standard deviation σ. Hence by Feynman-Kac, W obeys the backwards heat equation. The final change of variables transforms the backwards heat equation to the heat equation.

(c) Suppose that you wish to price the following options using the explicit finite difference method, which equation would you solve and what would be the boundary conditions? Justify your answers.

(i) A European put option. **[30%]**

Answer: In this case, I would use the backwards heat equation as it has better scaling properties than the Black-Scholes PDE. Writing x_t for the transformed variable, I'd put the top boundary at say $x_0 + 8\sigma\sqrt{T}$ and the bottom boundary at $x_0 - 8\sigma\sqrt{T}$ to ensure that the probability of the stock leaving the region is negligible. (A move of 8 standard deviations is very unlikely). At the top boundary $V_t = 0$ because the put will be essentially worthless. At the bottom boundary we ignore the unlikely event of the stock ending up out of the money and so find that $V_t = e^{-r(T-t)}K - S_t$. At the final time $V_t = \max\{K - S_t, 0\}$. The boundary conditions written in terms of W then follow from the formula $W = \exp^{-rt} V$.

(ii) An up and out knockout call option. **[20%]**

Answer: In this case I would use the Black–Scholes PDE as this matches the boundary conditions better than the heat equation. I'd put the top boundary at $S_t = B$ and the bottom boundary at 0. At the top and bottom boundaries $V = 0$. At the final time $V_t = \max\{S_t - K, 0\}$.

(d) What are the pros and cons of the implicit and explicit methods of solving partial differential equations by finite differences? **[10%]**

Answer: The explicit method is easier to code and can be used to price American options rather easily. The implicit method for the heat equation is numerically stable whatever discretization one chooses.

3. (a) The stochastic differential equation (SDE) for geometric Brownian motion is:

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

(i) Write down the difference equation for the Euler scheme for this stochastic differential equation. **[10%]**

Answer:

$$S_{t+\delta t} = S_t + S_t(\mu \delta t + \sigma \sqrt{\delta t} \epsilon_t)$$

where ϵ_t are i.i.d. normally distributed with mean 0 and standard deviation 1. δt denotes the chosen time step.

(ii) Describe a better method to simulate S_t. Explain your answer. **[10%]**

Answer: It is better to use the Euler method for the log of the stock price. This is because the log of the stock price follows Brownian motion with drift for which the Euler scheme yields the exact solution of the SDE.

(iii) Write the MATLAB code to simulate S_t. **[20%]**

Answer:

```
function S = simulateStock( S0, mu, sigma, T, nSteps, nPaths )
    dt = T/nSteps;
    epsilon = randn( nPaths, nSteps );
    ds = (mu-0.5*sigma^2)*dt + sigma*sqrt(dt)*epsilon;
    s = cumsum(ds,2) + log(S0);
    S = exp(s);
end
```

(b) A trader believes that the Black–Scholes model holds. She writes a European call option at the Black–Scholes price with strike K and maturity T and delta hedges her position at the times $(0, \delta t, 2\delta t, 3\delta t, \ldots)$.

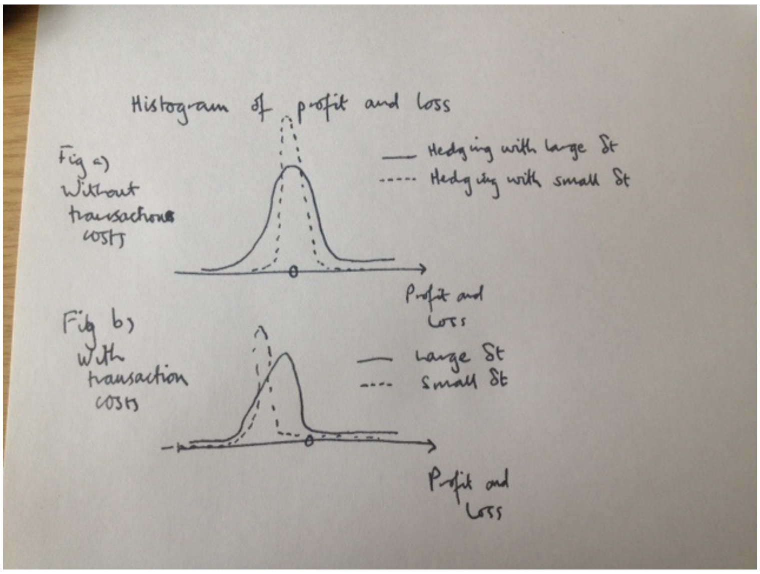
(a) Derive difference equations for the value in her risk free account at times $i\delta t$

Answer: Let Δ_t denote the delta of the stock at time t. Write b_t for the bank balance at time t. Let C denote the amount she charges her customer. At time 0, she receives C but purchases Δ_0 units of stock at price S_0. Hence

$$b_0 = C - (\Delta_0)S_0.$$

At intermediate times t she received interest on her bank account and purchases $\Delta_{t+\delta t} - \Delta_t$ units of stock at price S_t. Hence

$$b_{t+\delta t} = e^{r\delta t} b_t - (\Delta_{t+\delta t} - \Delta_t) S_{t+\delta t}$$



At the final time, she receives interest, cashes in her $\Delta_{T-\delta t}$ units of stock and if necessary pays the customer the payoff of the option. Hence

$$b_T = e^{r\delta t} b_t + (\Delta_{T-\delta t}) S_T - \max\{S_T - K, 0\}$$

[**30%**]

(b) Sketch a graph showing how you would expect her profit and loss to be distributed if she is correct. How will your graph change as δt is reduced? [**10%**]

Answer: The graph should be a histogram of profit and loss. It should be bell shaped around 0. δt is reduced, the histogram will become more concentrated around 0.

(c) Suppose that in fact there is a 1% bid-ask spread at all times that she has forgotten to take into account. How would the graphs change? Explain your answer. [**20%**]

Answer: The histogram will move to the left since she is sure to lose money on average due to transaction costs. For smaller δt, the further the graph will be moved to the left due to the increasing effect of transaction costs.

Note: many students blindly drew the wrong graph showing the rate of convergence as δt tends to 0. This was not what was asked

4. (a) Define the term *pseudo square root.* **[10%]**

Answer: If M is square, symmetric matrix, a pseudo square root A of M is a matrix which satisfies $M = AA^t$.

(b) Define the term *Cholesky decomposition.* **[10%]**

Answer: If M is square, symmetric and positive definite, the Cholesky decomposition is the unique lower triangular pseudo square root of M with positive diagonal.

(c) Explain why Cholesky decomposition is useful for simulating stochastic processes. Give a financial example of when you might use it. **[30%]**

Answer: Cholesky decomposition allows us to compute multivariate normal distributions with desired covariance matrix Σ. If A is pseudo square root of Σ and ϵ is a vector of i.i.d. standard normal variables, then $A\epsilon$ will be normally distributed with mean 0 and have the desired covariance. Thus given SDEs with noise terms given by correlated Brownian motions, we can use Cholesky decomposition to simulate the correlated Brownian motions and then use a numerical scheme such as the Euler scheme to simulate the SDE.

For example, in the Heston model one parameter is the correlation between the noise driving the stochastic volatility term and the noise driving the stock price process.

(d) Find the Cholesky decomposition of the following matrix

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

where $(-1 < \rho < 1)$. **[20%]**

Answer: Expand

$$\begin{pmatrix} a & 0 \\ b & c \end{pmatrix} \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

with $a > 0$ and $c > 0$ to find:

$$\begin{aligned} a^2 &= 1 \\ ab &= \rho \\ b^2 + c^2 &= 1 \end{aligned}$$

Hence $a = 1$, $b = \rho$, $c = \sqrt{1 - \rho^2}$.

(e) Find two more pseudo square roots of this matrix. **[30%]**

Answer: Two obvious answers are:

$$\begin{pmatrix} -1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ \rho & -\sqrt{1-\rho^2} \end{pmatrix}$$

Other less obvious answers include:

$$\begin{pmatrix} \sqrt{1-\rho^2} & \rho \\ 0 & 1 \end{pmatrix}$$

or diagonalizing to give:

$$\frac{1}{\sqrt{2}}\begin{pmatrix} \sqrt{1+\rho} & \sqrt{1-\rho} \\ \sqrt{1+\rho} & -\sqrt{1-\rho} \end{pmatrix}$$

Remark: Despite actually being a very easy question, students struggled with this, probably because it was unexpected.

5. (a) Write pseudo code to show how you would compute the price of an up and out call option with strike K and barrier B by the Monte Carlo method in the Black–Scholes model. [**40%**]

Answer:

```
function price = priceByMonteCarlo( S0, r, sigma, T, K, B, nSteps, nPaths )
stockPaths = generatePaths(S0, r, sigma, T, nSteps, nPaths );
% Note we use the Q-measure model
payoff = computeKnockoutPayoff( stockPaths, K, B);
price = exp(-r*T) * mean( payoff );
end

function payoff = computeKnockoutPayoff( stockPaths, K, B )
hitBarrier = max( stockPaths > B, [], 2 ); % max over time dimension
    finalPrice = stockPaths(:,end);
payoff = ~hitBarrier .* max( finalPrice - K, 0 );
end

function S = generatePaths( S0, mu, sigma, T, nSteps, nPaths )
    dt = T/nSteps;
    epsilon = randn( nPaths, nSteps );
    ds = (mu-0.5*sigma^2)*dt + sigma*sqrt(dt)*epsilon;
    s = cumsum(ds,2) + log(S0);
    S = exp(s);
end
```

If you are asked to write pseudo code then that means I'm not going to be very fussy about the language of the code you write, more the ideas. So most of the marks here go for computing a discounted mean in a Q-measure model. I am also hoping for an intelligent division of the code into 3 functions, as this is important to making your code testable.

(b) Describe how you could compute the delta of the option by the Monte Carlo method. [**20%**]

Answer: Use the central estimate:

$$\frac{\text{price}(S_0 + h) - \text{price}(S_0 - h)}{2h}$$

and compute each of the prices using Monte Carlo. It is important to use the same numbers for each simulation when computing the prices by Monte

Carlo. You should choose h to be reasonably small, but not ridiculously so - say $h = S_0 \times 10^{-6}$.

(c) How would the accuracy of your computation of the delta be related to the size of the Monte Carlo simulation? Give the mathematical reason for your answer. **[10%]**

Answer: Let N be the number of samples in our Monte Carlo calculation. The accuracy will be

$$O(h^2) + \text{random error}$$

with the first error coming from the fact that we are only approximating the derivative and the second being an approximately normally distributed random error with mean 0 and a standard deviation proportional to $\frac{1}{\sqrt{N}}$. We expect this random error to be the main source of error. The central limit theorem explains the dependence on

(d) Describe how you would test your computation of the delta. **[30%]**

Answer: I would test computeKnockoutPayoff in isolation. I would generatePaths in isolation (e.g. for example I would check that it produced stock price paths with the expected mean and standard deviation). Test that priceByMonteCarlo replicates European call prices when barrier is high and is 0 when barrier is low. I would also repeat this test for the computation of the delta. I would also seed the random number generator to ensure tests are reliable.