MA124 Maths by Computer: Assignment 3

A. Monte Carlo Integration (8 marks)

Consider the integral

$$\int_a^b f(x)\,dx$$

where $f(x) = \sin(x^2)$, $a = 0$ and $b = \sqrt{2\pi}$. As noted here, $\sin(x^2)$ does not have an elementary antiderivative so we use numerical methods to evaluate this integral.

Assignment: Estimate the above integral using two Monte Carlo methods.

1. Use Monte Carlo to estimate $\bar{f}$, the mean value of f over the interval $[a, b]$. From this estimate the integral.

1. Use the fact that

   $$\int_a^b f(x)\,dx = \text{area under the curve}$$

   where "under the curve" means the *signed area* between the curve $y = f(x)$ and the x axis. Use Monte Carlo to estimate the integral by estimating areas.

Details:

- Approach 1 is straightforward. You can use 10^6 samples to get a reasonably accurate estimate for the integral, but DO NOT PLOT THE SAMPLES! Report the value of the integral and produce a plot showing $y = f(x)$ and a box $[a, b] \times [0, \bar{f}]$.
- Approach 2 requires Monte Carlo samples in a rectangular region $[a, b] \times [y_{min}, y_{max}]$. The graph of $y = f(x)$ partitions the rectangle into three regions. From the areas of these regions estimate the integral and report the value. For a sample size of $N = 2000$ (NOT LARGER!) produce a plot showing the Monte Carlo points coloured according to the region they are in. There should be three colours. Include on the plot the curve $y = f(x)$ and the x axis.
- (Optional) Plot a small square with the average colour of the plot of Approach 2.

B. Geometric Brownian Motion (12 marks)

The stochastic differential equation

$$\dot{S}(t) = \mu S(t) + \sigma S(t)\xi(t)$$

plays a central role in mathematical finance. This equation describes what is known as Geometric Brownian Motion. $S(t)$ is taken to be positive and represents the value of a stock at time t. The deterministic ODE $\dot{S}(t) = \mu S(t)$, describes exponential growth in the value of a stock. We consider $\mu > 0$ and refer to μ as the growth rate. The term $\sigma S(t)\xi(t)$ describes fluctuations in the value of a stock. $\xi(t)$ is a Gaussian random variable. The parameter $\sigma > 0$ is known as the volatility. Notice that the fluctuating term contains $S(t)$ itself. This models the fact that fluctuations in a stock price are proportional to the price.

Time t is measured in years. Typical values for μ are between 0.01 and 0.10, corresponding to 1% and 10% annual growth (non-compounded). We assume values for σ are between 0.1 and 0.4.

This and this are simple articles on investopedia.com that might be of interest.

Assignment: Write a Python function to solve the above SDE by Euler's method. Then call the function, plot and analyse the solution for one choice of parameter values μ and σ.

Details:

- Implementation:
  - You must write a function `SDE_GBM(S0, tf, mu, sigma, Npaths)` that compute `Npaths` paths of the SDE and returns `t, S`. The function should not plot the results.
  - `Nsteps` is not included as an argument to the function. `Nsteps` should be set to `365 * tf` corresponding to a time step of 1 day. (In practice it is common to use the number of trading days in a year, but we will use 365.)
  - The implementation is a straightforward generalisation of the example SDEs in the Week 6 notebooks. Copy and modify those examples as needed. You need to include S in the fluctuating term, but still use $\sqrt{\Delta t}$ as usual.

- Parameters:
  - Use a final time of $t_f = 5$, corresponding to 5 years.
  - Decide the values you want to use for the parameters μ and σ consistent with the discussion above.
  - Take the initial stock price to be $S_0 = 100$.

- Plots:
  - Plot `Npaths` sample paths $S(t)$. For this plot `Npaths` should not exceed 100. You can choose how many. Useful values are between `Npaths = 10` and `Npaths = 100`.
  - Now call your function with `Npaths = 2000`. Make a plot showing the mean +/- standard deviation for the 2000 paths as a function of time, as discussed in the Week 6 notebooks.
  - With the same data as the previous item, plot histograms at three times, one of which should be the final time. (You might not want equally spaced times.) Either plot all three histograms on the same plot, or else in three plots all with the same limits on all plots (so that they can be compared). Print the mean and standard deviation for each of the histograms.
  - (Optional) The histograms correspond to a known distribution. Find it and use SciPy to generate the corresponding distribution and plot it on the histograms.

Further 5 marks

A further 5 marks will be awarded for each assignment based on overall quality and clarity of the submission; the level of understanding demonstrated; originality, creativity and engagement.

Submission

You will submit **two Jupyter notebooks**, one for each part. **These must be `.ipynb` files, not pdf files or any other file type**. You should make sure that your notebook **executes without error before submitting it**. From the Kernel menu, you should: `Restart Kernel and Run All Cells` as the last step before saving your notebook and submitting it.

Your submission should be such that:

- If the notebook is run and all code cells are collapsed, each notebook should be readable as a **short** report, primarily consisting of
  - A short introduction describing what problem is being solved and briefly how.
  - In the case of the Geometric Brownian Motion, specify the parameter values used.
  - Describe each of the figures (this is very important).
  - End each notebook with a short summary of what has been found.
- Use the example notebooks as a guide for Python style. Make sure your functions have comments describing what they do. One assumes the reader understands Python. Add comments to set off blocks of code or to note anything tricky. In most cases Python code explains itself.
