Series 2a

1. Monte Carlo Approximation for Numerical Integration

Let $A, B \subseteq \mathbb{R}^2$ be the sets defined by

$$A = \left\{(x,y) \in \mathbb{R}^2 \colon (x-2)^2 + y^2 \leq 4\right\} \quad \text{and} \quad B = \left\{(x,y) \in \mathbb{R}^2 \colon x^2 + (y-2)^2 \leq 4\right\},$$

and consider the function $f \colon \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(x,y) = \mathbb{1}_{A \cap B}(x,y) \cdot |x|^{2/3}.$$

Let $(\Omega, \mathcal{F}, P)$ be a probability space, and let $Y_n, Z_n : \Omega \to \mathbb{R}, n \in \mathbb{N}$, be independent $\mathcal{U}_{(0,1)}$-distributed random variables. For each $N \in \mathbb{N}$, define the random variable $I_N \colon \Omega \to \mathbb{R}$ by

$$I_N = \frac{4}{N} \sum_{n=1}^{N} f(2Y_n, 2Z_n).$$

The random variables $I_N, N \in \mathbb{N}$, are thus Monte Carlo approximations of $\mathbb{E}_P[4f(2Y_1, 2Z_1)]$.

a) Prove or disprove the following statement: The random variables I_N, $N \in \mathbb{N}$, are P-unbiased estimators of the integral

$$\int_0^2 \int_0^2 f(x,y) \, \mathrm{d}x \mathrm{d}y.$$

b) Write a Matlab function `MonteCarlo(`N`)` with input $N \in \mathbb{N}$ and output a realization of I_N.

c) Write a Matlab function `MonteCarloPlot()` which, for each $k \in \{2,3,4,5,6\}$, plots five realizations of I_{10^k}, each represented by a blue star in a coordinate plane. Plot the values of k on the x-axis and the corresponding realizations of I_{10^k} on the y-axis. The plot should contain a total of 25 blue stars.

2. Approximative Realizations of a One-Dimensional Standard Brownian Motion

In this exercise, we do not distinguish between pseudo-random numbers and actual random numbers. Let A be the set defined by

$$A = \bigcup_{n=1}^{\infty} \left\{\mathbf{t} = (t_1, \ldots, t_n) \in [0,\infty)^n \colon \#_{\mathbb{R}}(\{t_1, \ldots, t_n\}) = n\right\},$$

and define the function length$\colon A \to \mathbb{N}$ such that for all $n \in \mathbb{N}$ and all $\mathbf{t} = (t_1, \ldots, t_n) \in [0,\infty)^n \cap A$,

$$\operatorname{length}(\mathbf{t}) = n.$$

Let further $Q : A \to \bigcup_{n=1}^{\infty} \mathbb{R}^{n \times n}$ be the function defined such that for all $n \in \mathbb{N}$ and for all $\mathbf{t} = (t_1, \ldots, t_n) \in [0,\infty)^n \cap A$, we have

$$Q(\mathbf{t}) = \left(\min\{t_i, t_j\}\right)_{(i,j) \in \{1,\ldots,n\}^2}.$$

Write a Matlab function `StandardBrownianMotion(t)` with input $\mathbf{t} \in A$ and output a realization of an $\mathcal{N}_{0,Q(\mathbf{t})}$-distributed random variable. The Matlab function `StandardBrownianMotion(t)` may use at most length$(\mathbf{t})$ realizations of an $\mathcal{N}_{0,I_{\mathbb{R}}}$-distributed random variable.

Test your implementation by calling the following Matlab commands:

```
% Set random number generator to default
rng('default');

% Define parameters
N = 10^3;
preimage = linspace(0, 1, N+1);

% Generate and plot realizations of the standard Brownian motion
X = StandardBrownianMotion(preimage);
plot(preimage, X, 'b');
hold on;

X = StandardBrownianMotion(preimage);
plot(preimage, X, 'r');

X = StandardBrownianMotion(preimage);
plot(preimage, X, 'g');

% Add labels and a title
xlabel('t');
ylabel('X(t)');
title('Realizations of Standard Brownian Motion');
legend('BM (blue)', 'BM (red)', 'BM (green)');

hold off;
```

3. Brownian Motion and Monte Carlo Simulation

Let $(\Omega, \mathcal{F}, P)$ be a probability space and $W: [0,T] \times \Omega \to \mathbb{R}$ be a standard Brownian motion ($T > 0$).

a) For $N \in \mathbb{N}$, define a temporal discretization (or mesh) on the time interval $[0,T]$ by:

$$0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T, \quad t_n := \frac{n}{N}T. \tag{1}$$

Write a MATLAB function `BrownianMotion(`T`, `N`)` with inputs $T \in (0, \infty)$ and $N \in \mathbb{N}$, and output a realization of a $\big(W_0, W_{t_1}, \ldots, W_{t_{N-1}}, W_T\big)\big(P_{\mathcal{B}(\mathbb{R}^{N+1})}\big)$-distributed random variable. (You may use your results from Problem 2.) Now, assume $T = 1$. Write a MATLAB function `BrownianMotionPlot()` which uses the function `BrownianMotion(`T`, `N`)` to generate and plot 5 sample paths of the process $\widetilde{W}^{1000}$ over the time interval $[0,1]$, where, for $N \in \mathbb{N}$, the process $\widetilde{W}^N: [0,T] \times \Omega \to \mathbb{R}$ is defined by the linear interpolation of W at the mesh points $\{t_0, t_1, \ldots, t_N\}$, that is:

$$\widetilde{W}_t^N = \left(n + 1 - \frac{tN}{T}\right) W_{t_n} + \left(\frac{tN}{T} - n\right) W_{t_{n+1}},$$

for all $n \in \{0, 1, \ldots, N-1\}$ and each $t \in [t_n, t_{n+1}]$.

b) Let $X = (X_t)_{t \in [0,1]}$ be the stochastic process defined by $X_t = 1 + t + 3W_t$. First, prove that

$$\mathbb{E}\big[e^{\sigma W_t}\big] = \exp\big(\tfrac{1}{2}\sigma^2 t\big), \quad \text{for all} \quad \sigma \in \mathbb{R} \quad \text{and} \quad t \in [0,1].$$

Then, perform the following tasks:

(i) Plot 10 sample paths of X, using the discretization (1) for $N = 10^3$.

(ii) Compute $\mu := \mathbb{E}[\exp(X_1)]$ exactly and then approximate it using a Monte Carlo simulation with $M = 10^5$ samples. Provide a 95% confidence interval for your estimate based on the Central Limit Theorem (CLT), and compare the exact value of μ with your simulation result.

Submission Deadline: Wednesday, 23 October 2024, by 2:00 PM.

(1)

$A, B \subseteq \mathbb{R}^2$, $A = \{(x,y) \in \mathbb{R}^2 : (x-2)^2 + y^2 \leq 4\}$ and
$B = \{(x,y) \in \mathbb{R}^2 : x^2 + (y-2)^2 \leq 4\}$

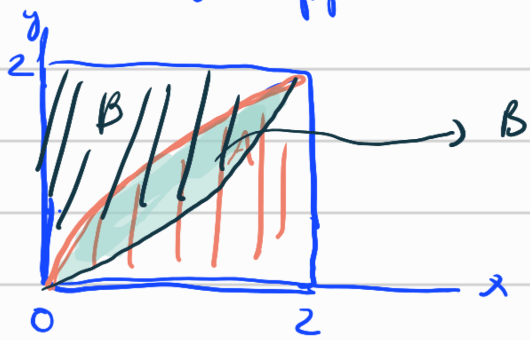
and consider $f : \mathbb{R}^2 \to \mathbb{R}$ tq $f(x,y) = \mathbb{1}_{A\cap B}(x,y) \cdot |x|^{2/3}$

$Y_n, Z_n : \Omega \to \mathbb{R}$ $U_{(0,1)}$-indep. r.v. For each $N \in \mathbb{N}$:

$$I_N = \frac{4}{N} \sum_{n=1}^{N} f(2Y_n, 2Z_n)$$

are monte carlo approximations of $\mathbb{E}_P[4 f(2Y, 2Z_1)]$

a)



$B \cap A$ it's the area where $f \neq 0$.

To see that I_N are P-unbiased estimators of $\int_0^2 \int_0^2 f(x,y)\,dx\,dy$ we need to check that

$$\mathbb{E}[I_N] = \int_0^2 \int_0^2 f(x,y)\,dx\,dy.$$

First and second condition of the "non-bias" definition already hold because we have a probability space, and, hence, a probability measure. Also, $I_n \in \mathcal{L}^1(P, |\cdot|_{\mathbb{R}})$.

Now,

all Z_i and Y_i are independent

$$\mathbb{E}\left[\frac{4}{N}\sum_{n=1}^{N} f(2Y_n, 2Z_n)\right] = \frac{4}{N}\,\mathbb{E}\left[\sum_{n=1}^{N} f(2Y_n, 2Z_n)\right] =$$

$$4\,\mathbb{E}[f(2Y_n, 2Z_n)] = 4 \cdot \int f(2y, 2z) \cdot \frac{1}{4}\, g_{Y,Z}(y,z) = \boxed{\int_0^2 \int_0^2 f(2y, 2z)\,dy\,dz}$$

$g_{Y,Z}(2y, 2z) = \frac{1}{4}\, g_{Y,Z}(y,z)\,dy\,dz$

joint densities

② In order to get a realization of $N(0, Q(t))$, let's construct the following:

- Given $t=(t_1,\dots,t_n) \in A$, we reorder t with a new array called t' which t_i are ordered ($t_i' < t_j'$ if $i<j$)

$$t' = (t_1', \dots, t_n')$$

- Now, as we are asked to use at most $n = \text{len}(t)$ realizations of a $N(0,1)$ random variable, we construct the following:

Given $Z_i \sim N(0,1)$, $X_i = Z_i \cdot \sqrt{(t_i' - t_{i-1}')}$

Where $X_i \sim N(0, t_i' - t_{i-1}')$

↳ This X_i realizations are going to be the increments of the standard brownian motion. Hence, if we take the values of the brownian motion:

$$S_{i+1} = S_{i-1} + X_i$$
$$S_0 = X_0$$

Now, we see how $Var(S_j - S_i) = Var(S_i + X_{i+1} + \dots + X_j - S_i)$

$$= \sum_{k=i+1}^{j} Var(X_k) = t_{i+1}' - t_i' + \dots + t_j' - t_{j-1}' =$$

$$Var(S_j - S_i) = t_j - t_i \checkmark$$

Let's proof, also, how $Cov(S_i, S_j) = \min\{t_i, t_j\}$,
By construction, $Cov(S_i, S_j) = Cov(S_i, S_i + \sum_{k=i+1}^{j} X_k)$

$$= \underbrace{\cancel{Cov(\sum X_j, S_i)}}_{0 \text{ (independence of increments)}} + Cov(S_i, S_i) = t_i = \min\{t_i, t_j\} \checkmark$$

So, the stochastic process constructed $\{S_i\}$ it is distributed $N(0, Q(t'))$, that if we reconstruct to de initial order we get a sample of $N(0, Q(t))$ with $Q = \min\{t_i, t_j\}$ for t unordered.

Note: The mean $=0$ of each S_i is straightforward by construction $E[S_i] = \sum_{j=1}^{i} E[X_j] = 0$.

See code for the implementation //

(3) (Brownian Motion and Montecarlo simulations)

b) $(X_t)_{t\in[0,1]}$ be the stochastic process defined by $X_t = 1+t+3W_t$.

$$\mathbb{E}[e^{\sigma W_t}] \underset{\uparrow}{=} \int e^{\sigma x} \cdot \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}} dx = \int \frac{1}{\sqrt{2\pi t}} e^{-\frac{(x^2 - 2\sigma x t + \sigma^2 t^2)}{2t}} \cdot e^{\frac{\sigma^2 t^2}{2t}} dx$$

$W_t \sim N(0,t)$

$$= e^{\frac{\sigma^2 t}{2}} \underbrace{\int \frac{1}{\sqrt{2\pi t}} e^{-\frac{(x-\sigma t)^2}{2t}} dx}_{=1} = \boxed{e^{\frac{\sigma^2 t}{2}}}$$

(ii) $\mu = \mathbb{E}[e^{1+1+3W_1}] \; e^2 \cdot \mathbb{E}[e^{3W_t}]\big|_{t=1} = e^2 \cdot e^{\frac{9}{2}} = \boxed{e^{\frac{13}{2}}}$

In the code, we only 2 time steps $t_0=0$ and $t_1=1$, as we know that the distribution of W_t only depends on t.

Based on CLT, we can estimate a 95% confidence interval.