MA3227 Numerical Analysis II

Assignment 3

Deadline: 23 April 2021, 12 noon
Total marks: 20

1 Uniformly distributed points on the disk [10 marks]

The goal of this exercise is to use inverse transform and rejection sampling for generating uniformly distributed points on the unit disk

$$D \;=\; \big\{x \in \mathbb{R}^2 \mid \|x\|_2 \leq 1\big\}.$$

1. Complete `randdisk_rejection()` such that it samples $X \sim \operatorname{Uniform} D$ using rejection sampling with proposal distribution $\operatorname{Uniform}[-1,1]^2$.

   Hint. You can check your answer using `plot_samples()`.

2. How many samples of $\operatorname{Uniform}[-1,1]^2$ does `randdisk_rejection()` require on average to produce a single sample of $\operatorname{Uniform} D$?

3. Show that if $R \sim \operatorname{PDF}(r \mapsto 2r \text{ on } [0,1])$ and $\Phi \sim \operatorname{Uniform}[0, 2\pi]$ independently, then

   $$X(R,\Phi) = \begin{pmatrix} R\,\cos(\Phi) \\ R\,\sin(\Phi) \end{pmatrix} \;\sim\; \operatorname{Uniform} D.$$

   Hint. You may want to use that

   $$\int_D f(x)\,dx = \int_0^1 \int_0^{2\pi} f\big(X(r,\phi)\big)\,r\,d\phi\,dr.$$

4. Complete `randdisk_transform()` such that it samples $X \sim \operatorname{Uniform} D$ using the result from Task 3.

   Hint. You can check your answer using `plot_samples()`.

5. [unmarked] Which of `randdisk_rejection()` and `randdisk_transform()` do you think is faster? Check your answer using `performance_shootout()`.

2 Importance sampling for highly concentrated integrals [10 marks]

The goal of this task is to use Monte Carlo sampling to estimate the integral

$$\mathbb{E}[f(X)] \;=\; \int_0^1 f(x)\,dx \qquad \text{where} \qquad X \sim \operatorname{Uniform}[0,1].$$

1. Complete `uniform_sampling(f,N)` such that it estimates both

   $$\mathbb{E}[f(X)] \qquad \text{and} \qquad \operatorname{Var}[f(X)] \;=\; \mathbb{E}\Big[\big(f(X) - \mathbb{E}[f(X)]\big)^2\Big]$$

   using N Monte Carlo samples.

2. [Unmarked] The provided function `sin_integral()` uses `uniform_sampling()` to estimate

$$\int_0^1 \tfrac{\pi}{2}\,\sin(\pi\,x)\,dx \;=\; 1, \tag{1}$$

and `plot_histogram()` to visualise the accuracy of the estimates. Use this function to check your answer to Task 1. If your answers are correct, you will observe that $\tilde{\mathbb{E}}_{100}[f(X)]$ lies within a distance 0.1 of the exact expectation with high probability.

3. [Unmarked] The provided function `concentrated_integral()` uses `uniform_sampling()` to estimate

$$\int_0^1 \tfrac{1}{0.0906401}\,\exp\Big(-\big(20\,(x-0.5)\big)^4\Big)\,dx \;\approx\; 1. \tag{2}$$

Run this function and observe how the error in $\tilde{\mathbb{E}}_{100}[f(X)]$ increases to roughly 0.5 when applied to (2) instead of (1).

The problem in `concentrated_integral()` is that the integrand in (2) is highly concentrated (see `plot_integrand()`); hence it is far from constant and hence it has a high variance. We can improve the situation by replacing direct sampling with importance sampling with a sampling distribution $\mathrm{PDF}(q)$ such that new random variable

$$f(Y)\,\tfrac{p(Y)}{q(Y)}, \qquad Y \;\sim\; \mathrm{PDF}(q),$$

is as close to constant as possible. Put differently, this means that we want to estimate (2) through importance sampling with a sampling distribution which is as close to $f(x)\,p(x)$ as possible but which is also easily samplable. Looking at `plot_integrand()` once more, we conclude that choosing $q(x)$ to be an appropriately shifted and scaled Gaussian should achieve this goal.

4. Complete `importance_sampling(f,N,m,s)` such that it estimates both

$$\mathbb{E}\Big[f(Y)\,\tfrac{p(Y)}{q(Y)}\Big] \qquad \text{and} \qquad \mathrm{Var}\Big[f(Y)\,\tfrac{p(Y)}{q(Y)}\Big]$$

using N Monte Carlo samples, where $Y \sim \mathrm{Normal}(m, s^2)$ and

$$p(x) \;=\; \begin{cases} 1 & \text{if } 0 \le x \le 1, \\ 0 & \text{otherwise}, \end{cases} \qquad\qquad q(x) \;=\; \tfrac{1}{\sqrt{2\pi}\,s}\,\exp\Big(-\tfrac{(x-m)^2}{2s^2}\Big)$$

denote the PDFs of $\mathrm{Uniform}[0,1]$ and $\mathrm{Normal}(m, s^2)$, respectively.

Hints. You can sample $Y \sim \mathrm{Normal}(m, s^2)$ using `y = m + s*randn()`. The normal PDF $q(x)$ is implemented in `normal_pdf(m,s,x)`.

5. (Unmarked) Rerun `concentrated_integral()`, but this time change `importance = false` to `importance = true`. Observe how importance sampling achieves a much better accuracy than uniform sampling.