UCSB, Physics 129AL, Computational Physics: Section Worksheet, Week 7B

Zihang Wang (UCSB), zihangwang@ucsb.edu

February 20, 2025

Section Participation and Submission Guidelines

Section attendance is required, but you do not need to complete all the work during the section. At each section, the TA will answer any questions that you might have, and you are encouraged to work with others and look for online resources during the section and outside of sections. Unless otherwise stated, the work will be due one week from the time of assignment. The TA will give you 1 point for each task completed. You can see your grades on Canvas.

We will use GitHub for section worksheet submissions. By the due date, you should have a single public repository on GitHub containing all the work you have done for the section work. Finally, upload a screenshot or a .txt file to Canvas with your GitHub username and repository name so the TA knows who you are and which repository you are using for the section.

Remember: talk to your fellow students, work together, and use GPTs. You will find it much easier than working alone. Good luck! All work should be done in the Docker container, and don't forget to commit it to Git!

Task 1: Sobol sequence

Using dynamic programming, generate a 2D Sobol sequence with the following primitive polynomial,

$$p_2(x) = x^3 + x^2 + x + 1. \tag{1}$$

Write a Python program that calculates the first 50 elements of the Sobol sequence, and plot them in 2D. Use the following initial condition,

$$m_1 = 1, \quad m_2 = 3, \quad m_3 = 5,$$

$$v_{2,1} = \frac{1}{2}, \quad v_{2,2} = \frac{3}{4}, \quad v_{2,3} = \frac{5}{8}.$$

Task 2: Monte Carlo Integration vs. Deterministic Quadrature

In this problem, you will be looking at the difference between various deterministic and non-deterministic methods. Let's consider an ellipsoid parametrized by,

$$\frac{x^2+y^2}{\beta^2}+\frac{z^2}{c^2}=1$$

where $-1 \leq x \leq 1$ and $-1 \leq y \leq 1$. Let's consider the surface element,

$$A=\int_{\partial V} 1\, dA$$

a)

Write down the explicit the formula in LaTeX for surface area. Hint: You should be able to express it using a single variable.

b)

Use the above formula to approximate the area by using **two** deterministic quadrature techniques you like, 1) the midpoint rule and 2) Gaussian quadrature.

Verify the calculated values with the formulas below:

$$A=2\pi\beta^2\left(1+\frac{c}{ae}\sin^{-1}(e)\right),\quad e=1-\frac{\beta^2}{c^2}.$$

What is a in this formula? You may realize that I did not provide the value of a and c. Plot the error as a heatmap with various β, c values ranging from $[0.001, 1000]$. Do you expect the value of β and c have any influence on the error?

c)

Let's first consider the surface with non-deterministic quadrature techniques (Monte Carlo, MC). As discussed in class, the following simple Monte Carlo simulation can be used to approximate a 2d integral,

$$\int_a^b f(x)dx=\lim_{N\to\infty}\frac{b-a}{N}\sum_{i=0}^{N} f(x_i),\quad X\sim U(a,b)$$

and we set $2\beta=c=1$. For each sampling size, N=[10,100,1000,10000,100000], calculate the error from the exact solution, and visualize them. Use a uniform distribution in this case.

d)

For each sampling size, N=[10,100,1000,10000,100000] , use the importance sampling and inverse transformation sampling, perform the MC in Python via the following proposal functions,

$$q_1(x) = \exp(-3x), \tag{2}$$

and,

$$q_2(x) = \sin^2(5x). \tag{3}$$

Visualize the difference from the uniform sampling in part **c)**.

e) Box–Muller Transform

Let's take a closer look at a joint probability,

$$p(x, y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}$$

and changing from x, y to θ, R, we have,

$$p(r, \theta) = p(x, y)\frac{\partial|x, y|}{\partial|\theta, R|} = \frac{1}{2\pi} Re^{-r^2/2},$$

where $\frac{\partial|x,y|}{\partial|\theta,r|} = \det|J| = r$. This is called the Rayleigh distribution, which coincides with the χ-square distribution with two degrees of freedom.

Let's consider two random variables that follow a standard normal distribution, with X and Y denoted as $X, Y \sim \mathcal{N}(0, 1)$, representing two components of a vector. The cumulative distribution function (CDF) of the joint probability is given by:

$$P(\sqrt{X^2 + Y^2} \leq R) = \int_0^{2\pi} \int_0^R r \frac{1}{2\pi} e^{-r^2/2}\, dr\, d\theta.$$

It gives the same PDF as the probability density above.

Using the Box–Muller transform, write a python function that generates Gaussian distributed samples (return a numpy array) with mean μ and standard deviation σ.

Plot the histogram with sampling size, N=[10,100,1000,10000,100000].

f)

Do the Monte Carlo integration, this time using different Gaussian-distributed samples (Gaussian proposal function),

$$\int_a^b f(x)dx = \lim_{N\to\infty} \frac{b-a}{N} \sum_{i=0}^{N} f(x_i), \quad x_i \sim N(\mu, \sigma)$$

and we set $2a = c = 1$. Let's first assume $\mu = 0$ and $\sigma = 1$. For each sampling size, N=[10,100,1000,10000,100000], visualize the difference from those previous proposal functions.

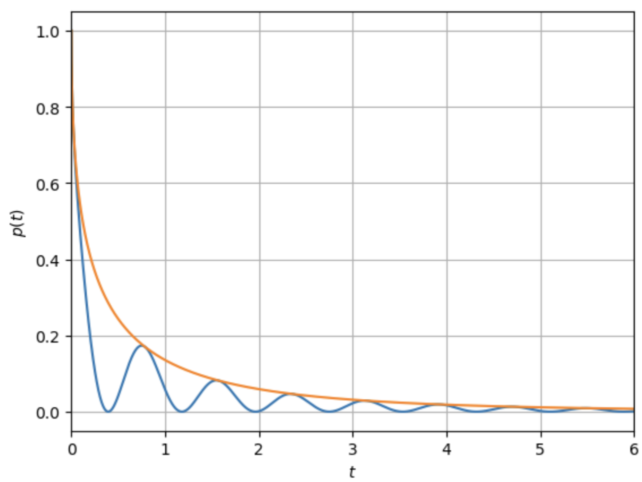
Next, test various μ and σ with a fixed $N = 10000$. How does it different from above?

Task 3: Rejection Sampling

Rejection sampling is used to generate data points that follow a particular complicated distribution. Let's say that the probability of observing a particle decay event at time t follows the probability density function (PDF),

$$p(x) = e^{-bt} \cos^2(at), \quad t \geq 0.$$

See below for an example when $a = 4$ and $b = 4$. (If you are interested, you can refer to the lecture notes on the Lorentzian function for energy dissipation.)



a)

Write a rejection sampling function by using a uniform proposal function,

$$q(t) \sim U(0, t_f)$$

to sample the PDF discussed above, with $a = 4b = 4$.

It should return a N sample numpy array. How do you select the t_f? Plot the resulting sample histogram with N=[100,1000,10000]. What is the rejection ratio, number of accepts divided by number of rejects?

b)

Write a rejection sampling function by using an exponential proposal function,

$$q(t) \sim Exp(1) = e^{-2t}$$

to sample the PDF discussed above, with the same $a = 4b = 4$.

It should return a N sample numpy array. Plot the resulting sample histogram with N=[100,1000,10000]. Make a comparison between the uniform proposal function and exponential proposal function for various sample sizes. What is the rejection ratio in this case?