

Computational Physics Assignment 4: Ising Model

R. Abele^a and N. Matera^a

^a*Eberhard Karls University of Tuebingen*

Abstract

The following exercises showcase several uses of Monte Carlo methods, specifically highlighting their use in simulating the Ising model.

Contents

1. Background	2
2. Program	2
3. Results	2
3.1. Task 1	2
3.1.1. Part A	2
3.1.2. Part B	4
3.2. Task 2	7
3.3. Task 3	7
3.4. Task 4	7

1. Background

2. Program

The programs for this series of problems were written using Rust. A `nix flake` is used to manage the programming environment and necessary dependencies; the programming environment can be entered by running `nix develop` (assuming that `nix` is installed). From there, one can run `cargo run` in any directory containing a Rust project to compile and run the program.

Results can be plotted by running `python3 ./<plotting_script_name>` in the directory containing the corresponding python script.

3. Results

3.1. Task 1

3.1.1. Part A

The first part of task 1 asks us to calculate π numerically through a Monte Carlo simulation involving the random distribution of points (x, y) in the unit square.

The area of a circle is of course

$$a = \pi r^2 \tag{1}$$

and that of the unit square is 1. If we center a circle with radius 1 on the origin, then the fraction of the unit square covered by the circle f is given by

$$f = \frac{\pi r^2}{4} \cdot \frac{1}{1} = \pi \frac{r^2}{4} \tag{2a}$$

$$= \frac{\pi}{4} \quad (2b)$$

$$\implies \pi = 4f. \quad (2c)$$

Furthermore, we can easily determine if a point $(x, y,)$ is in a unit circle by checking if its radius r is less than 1.

$$x^2 + y^2 < r^2 \quad (3a)$$

$$\implies x^2 + y^2 < 1 \quad (3b)$$

Ergo, we can create a program for this task by generating q random points (x, y) with $x, y \in [0, 1)$, and calculate the number of points within the unit circle p , meaning that

$$\pi = \frac{4p}{q}. \quad (4)$$

To show convergence for this task with consistent results, a seed-based random number generator was used and the logarithm of percent error was plotted as a function of points used.

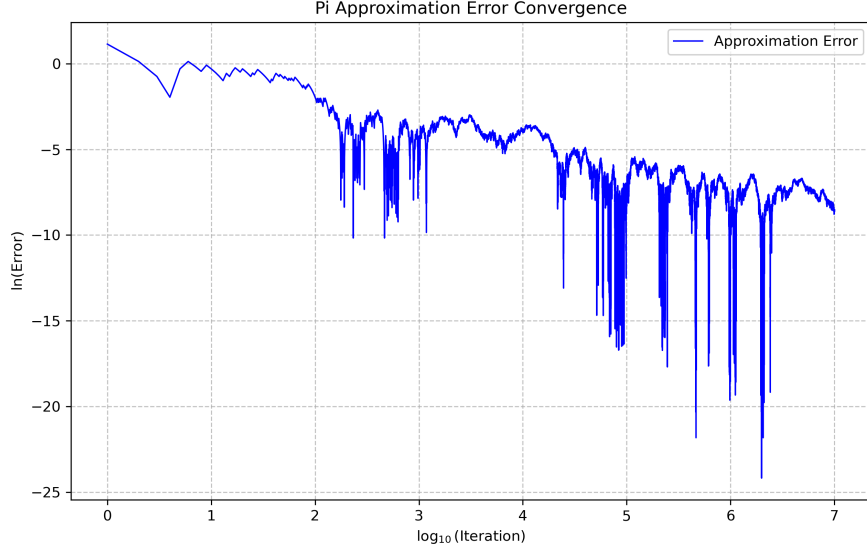


Figure 1: Plot of Pi Approximation as a function of Points used

As seen in 1, there is a steady decrease in the natural log of the error as a function of $\log_{10}(\text{iterations})$, indicating the expected increase in precision following an increase in data points.

3.1.2. Part B

This task asks us to first create a random number generator following a Gaussian distribution with an average μ and a width σ . This function should then be used to calculate the following integral through MC Integration:

$$I = (2\pi)^{-2} \int_{-\infty}^{\infty} dt \cos(t) \exp\left(-\frac{t^2}{2}\right). \quad (5)$$

We are given the analytical value $I = e^{-\frac{1}{2}} \approx 0.606531$ to check our answer.

When checked through numerical and analytical integration methods though, it is found that the value of the integral in the above expression is as follows:

$$\int_{-\infty}^{\infty} dt \cos(t) e^{-\frac{t^2}{2}} = \frac{\sqrt{2\pi}}{\sqrt{e}}. \quad (6)$$

This of course would not cancel out with the term $(2\pi)^{-2}$, leading us to conclude that the constant in front contains a mistake and should instead be $(2\pi)^{-\frac{1}{2}}$. When this change is implemented, the integral $I = e^{-\frac{1}{2}}$ as desired.

We know that the standard normal distribution $f(x)$ is given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2}\sigma^2}. \quad (7)$$

In our case $\mu = 0$ and $\sigma = 1$, yielding

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \quad (8)$$

Knowing this, we can refactor our given integral as follows:

$$I = \frac{1}{(2\pi)^{\frac{1}{2}}} \int_{-\infty}^{\infty} dt \cos(t) e^{-\frac{t^2}{2}} \quad (9a)$$

$$= \frac{\sqrt{2\pi}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt \cos(t) \left[\frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \right] \quad (9b)$$

$$= \int_{-\infty}^{\infty} dt \cos(t) f(t). \quad (9c)$$

$f(x)$ is of course normalized, meaning that

$$\int_{-\infty}^{\infty} dx f(x) = 1. \quad (10)$$

This in turn means that we replace $f(t)$ in the integral by selecting points according to this normal distribution:

$$I = \frac{1}{N} \sum_{i=0}^N \cos(x_i) \quad (11)$$

with N being the number of data points used and x being a normally distributed random value.

The required normal distribution was realized by using two uniformly generated numbers and a Box-Muller transform. The two random numbers $x, y \in (0, 1]$ were processed as follows:

$$R = \sqrt{-2 \cdot \ln x} \quad (12a)$$

$$\theta = 2\pi y, \quad (12b)$$

which can then be used to calculate the two normally distributed values Z_0, Z_1

$$Z_0 = R \cos(\theta) \quad (13a)$$

$$Z_1 = R \sin(\theta). \quad (13b)$$

We account for the modification of σ and μ by adding them as follows:

$$R = \sigma \cdot \sqrt{-2 \ln x}, \quad (14a)$$

$$\theta = 2\pi y, \quad (14b)$$

$$Z_0 = R \cos(\theta) + \mu, \quad (15a)$$

$$Z_1 = R \sin(\theta) + \mu. \quad (15b)$$

3.2. Task 2

3.3. Task 3

3.4. Task 4