Institute for Cross-Disciplinary Physics and Complex

Simulations Methods

Task 1

Author: Daniel Montesinos Capacete

October 2, 2023

1 Random Number Generator

All computations in this study will be conducted using the Julia programming language, employing its default random number generator, the Xoshiro256++ algorithm [1], unless specific cases necessitate the use of an alternative generator.

2 Exercise 15

Firstly, we will compute the integral analytically:

$$I = \int_0^1 \sqrt{1-x^2}dx = \int_0^{\pi/2} \cos(\theta)\sqrt{1-\text{sen}^2(\theta)}d\theta = \int_0^{\pi/2} \cos^2(\theta)d\theta =$$
$$= \int_0^{\pi/2} \frac{1+\cos(2\theta)}{2}\, d\theta = \left[\frac{\theta}{2} + \frac{1}{4}\sin(2\theta)\right]_0^{\pi/2} = \frac{\pi}{4} \approx 0.7853981643$$

where we have use the following trigonometric relation

$$\cos^2(\theta) = \frac{1+\cos(2\theta)}{2}$$

and change of variables

$$x = \text{sen}(\theta) \quad \longrightarrow \quad dx = \cos(\theta)d\theta \;\; ; \;\; x \in [0,1] \quad \longrightarrow \quad \theta \in [0, \pi/2].$$

In our numerical computations using the hit and miss Monte Carlo method, we will use the notation from the lecture's slides, setting a = 0, b = 1, and c = 1. While c could theoretically be greater than 1 (since the maximum value of $\sqrt{1-x^2}$ if 1 when $x \in [0,1]$), we choose this value for its superior performance in yielding accurate results. Following the lecture's slides, the root mean square *sigma* can be computed as

$$\sigma[\hat{I}] = c(b-a)\sqrt{\frac{p(1-p)}{M}} = \sqrt{\frac{\pi}{4}\left(1-\frac{\pi}{4}\right)\frac{1}{M}} \approx \frac{0.4105458419}{\sqrt{M}} \qquad (1)$$

where $p = \frac{\pi}{4}$ since it is the result of the integral and $c(b-a) = 1$. We can verify this relationship by visualizing the experimental data points for sigma obtained from the simulation, as illustrated in Figure 1. In every experiment, the outcomes exhibit variability. However, we are presenting the results from a specific execution as an example. On the left side of the figure, you can observe a remarkable alignment between the experimental points and the theoretical curve. Additionally, a linear regression analysis has been performed, as demonstrated on the right side of Figure 1. The coefficients resulting from this regression are documented in Equation (2). We have confirmed that the experimental data points adhere to the theoretical relationship described in Equation (1).

$$\sigma = \frac{\alpha}{\sqrt{M}} + \beta \;\; ; \quad \alpha = 0.3990 \pm 0.0041 \quad \beta = -0.00028 \pm 0.00056 \tag{2}$$

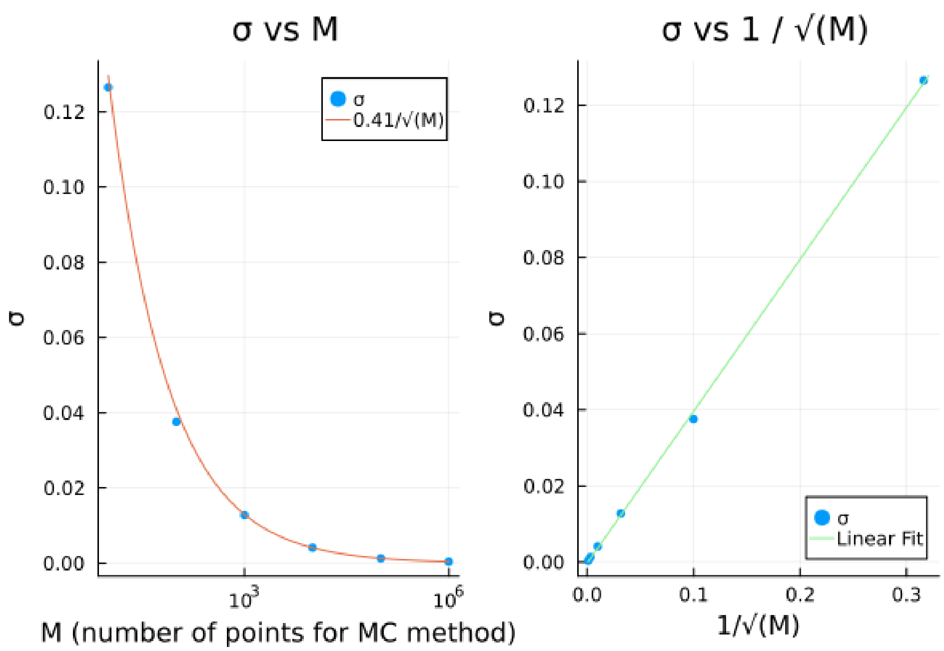

Figure 1: Left Panel: σ vs. M on a logarithmic scale. Right Panel: Linear fit for σ

We have additionally computed the real error by determining the absolute difference between the numerical integral and the analytical value $\frac{\pi}{4}$. The findings from this analysis are presented in Figure 2. On the left side of the figure, it is shown that the real error and sigma exhibit similar behavior. To characterize the dependence of the real error, we conducted another linear regression, and the resulting coefficients are documented in Equation (3). The trend is evident: as $M \to \infty$, the real error approaches zero. If we continue to increase the value of M, the slope would diminish because the error would approach zero at a slower rate.

$$\log_{10}(\text{Real Error}) = \alpha' \log_{10}(M) + \beta' \ ; \ \alpha' = -0.381 \pm 0.086 \quad \beta' = -1.05 \pm 0.33 \tag{3}$$

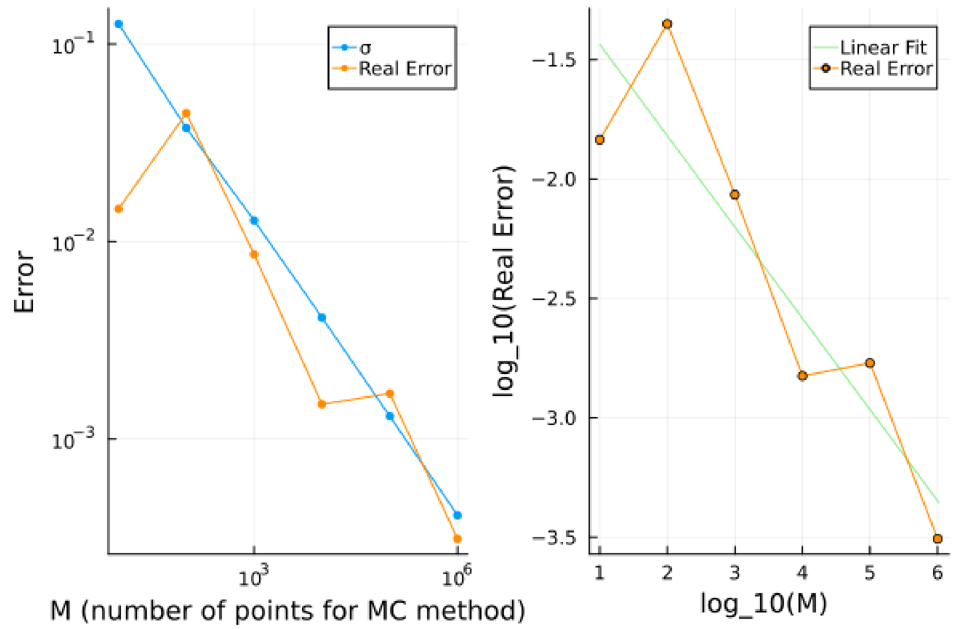

Figure 2: Left Panel: σ and Real Error vs. M on a logarithmic scale for both axes. Right Panel: Linear fit for Real Error

3 Exercise 17

We applied the Hit and Miss method with a sample size of M = 100 and conducted each experiment 10^4 times. In this case, we hae used 3 different random number generators available in Julia (Xoshiro256++, MersenneTwister, RandomDevice) [1]. The results can be seen in table 1. The percentages achieved for each random number generator exhibit remarkable similarity.

	Xoshiro256++	MersenneTwister	RandomDevice
1σ	66.89%	67.42%	66.86%
2σ	94.31%	94.25%	94.48%
3σ	99.45%	99.5%	99.54%

Table 1: Percentage of cases for which the numerical result differs from the exact one in less that 1σ, 2σ and 3σ

4 Exercise 19

When working with multidimensional Monte Carlo methods, we need to take into account the dimensionality of the integral. In the context of the hist and miss method, this consideration leads us to

$$\langle \hat{I} \rangle = c(b-a)^n p \quad ; \quad \sigma[\hat{I}] = c(b-a)^n \sqrt{\frac{p(1-p)}{M}} \tag{4}$$

and for the uniform sampling

$$G(x) = (b-a)^n g(x) \tag{5}$$

where n is the dimension of the integral. For the simpson's rule, we have implementd the simpson's 1/3 rule where the interval is divided in 2 subintervals and the 1D integral can be calculated as:

$$\int_a^b f(x)dx \approx \frac{h}{3}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) \quad ; \quad h = \frac{b` - a}{2} \tag{6}$$

In higher dimensions, it is possible to derive a general expression using recursion with $a = -1$ and $b = 1$, leading to the following result: (I have worked out this expression with other classmates)

$$I = \left(\frac{h}{3}\right)^n \sum_{\vec{x}\in\{-1,0,1\}^n} 4^{\left(n-\sum_{i=1}^n x_i^2\right)} f(\vec{x}) \tag{7}$$

Table 2 presents the results obtained for all the methods across various dimensions. Notably, the Simpson's 1/3 rule exhibits a disparity when compared to the Monte Carlo methods. This discrepancy arises from the limitation of employing only three points for integration, which proves insufficient to yield an accurate outcome. The Simpson method becomes impractical for higher dimensions because increasing the number of subintervals leads to complexity and significantly higher computational requirements compared to Monte Carlo methods.

	Hit and Miss		Uniform Sampling		Simpson's 1/3 rule	
Dimension	I	σ	I	σ	I	σ
1	1.32606	0.00094	1.32434	0.00057	1.4049	-
2	1.8254	0.0019	1.8295	0.0011	2.4422	-
3	2.8585	0.0038	2.865	0.0018	3.7554	-
5	5.871	0.012	5.8678	0.0051	9.1374	-
10	37.18	0.19	36.901	0.051	83.9559	-

Table 2: Results obtained through various numerical methods for the integral.

5 Exercise 20

We will compute the integral

$$I = \int_0^\infty \sqrt{x}\cos(x)e^{-x}dx$$

using the sampling method. In order to do that, will use $f_{\hat{x}}(x) = e^{-x}$ for our probability density function. Thus, the inverse of the cumulative distribution can be calculated following

$$F_{\hat{x}}(x) = \int_0^x e^{-x'}dx' = -e^{-x} + 1 \Longrightarrow x = -\ln(1 - F_{\hat{x}}(x)).$$

Then, we will use $x = -\ln(u)$ where u is a $U(0,1)$ random number. These are the results we get for different number of points:

$$I = 0.236 \pm 0.019 \quad M = 10^3$$

$$I = 0.2016 \pm 0.0063 \quad M = 10^4$$

$$I = 0.2036 \pm 0.0019 \quad M = 10^5$$

$$I = 0.20145 \pm 0.00063 \quad M = 10^6$$

References

[1] Julia:Random Numbers