Exercise – Understanding Monte Carlo integration using an exponential function

1. Aim of the exercise

Monte Carlo (MC) integration is a powerful technique for estimating definite integrals using random sampling. In this exercise, we apply MC integration to approximate the value of the Gaussian integral, a classic example in probability and statistics. The procedures we follow here closely mirror those used in the earlier exercise on MC integration with the identity function. It is recommended to review that exercise first, as the foundational explanations provided there will not be repeated here.

2. Theory

We do not provide a detailed treatment of MC integration theory here; this is covered in the dedicated theory exercise.

An important integral in probability theory is the Gaussian integral, which has a well-known closed-form solution:

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi} \approx 1.772.$$

This result underlies the normalization constant of the standard normal PDF. Specifically, since the standard normal PDF is defined as

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

the integral

$$\int_{-\infty}^{\infty} f(x)\,dx = 1$$

follows from the fact that

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi},$$

which ensures that the total area under the curve equals one when scaled appropriately.

In this exercise, we will approximate the Gaussian integral using MC integration. To apply MC integration, we must choose a sampling distribution over which to draw random inputs. We choose to sample from the standard normal distribution $X \sim \mathcal{N}(0,1)$ because its support matches the domain of e^{-x^2}, its shape aligns with the integrand, and its density is analytically tractable. The PDF of X is given by:

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

for all $x \in \mathbb{R}$.

We define the target function

$$\phi(x) = e^{-x^2},$$

and rewrite it as

$$\phi(x) = f_X(x) \cdot g(x),$$

where

$$g(x) = \frac{\phi(x)}{f_X(x)} = \frac{e^{-x^2}}{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}} = \sqrt{2\pi} \cdot e^{-\frac{1}{2}x^2}.$$

This allows us to express the integral as an expectation under the standard normal distribution:

$$\begin{aligned}\int_{-\infty}^{\infty} e^{-x^2}\,dx &= \int_{-\infty}^{\infty} f_X(x)\cdot g(x)\,dx \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \cdot \sqrt{2\pi} e^{-\frac{1}{2}x^2}\,dx \\ &= \mathbb{E}\left[\sqrt{2\pi} e^{-\frac{1}{2}X^2}\right] \\ &= \mathbb{E}\left[g(X)\right].\end{aligned}$$

Note that to compute the expected value of $g(X)$, we use a rule from probability theory: if we have a function of a random variable, we can compute its expectation by integrating that function times the PDF of the original variable. This is called the law of the unconscious statistician. This law allows us to avoid computing the distribution of $g(X)$ directly.

To estimate this expectation, we draw N independent samples $x_1, \ldots, x_N \sim \mathcal{N}(0,1)$, and compute the MC estimator:

$$\frac{1}{N}\sum_{i=1}^{N} g(x_i) = \frac{1}{N}\sum_{i=1}^{N} \sqrt{2\pi} e^{-\frac{1}{2}x_i^2}.$$

This yields an approximation of the Gaussian integral using standard Monte Carlo integration.

3. Define the number of samples

Clear the memory. Set the sample size.

```
%% 3. Define the number of samples

% 3.1. Clear workspace and memory
clear;

% 3.2. Number of random samples to estimate the integral
N_samples = 1000;
```

4. Monte Carlo integration: Estimating integrals via sampling

We estimate the value of the Gaussian integral using MC integration by sampling from the standard normal distribution. First, we generate random draws from $\mathcal{N}(0,1)$. At each sampled point, we evaluate the unnormalized Gaussian function $e^{-\frac{1}{2}x^2}$, which computes the height of the Gaussian curve at that location. Averaging these values and multiplying by $\sqrt{2\pi}$ yields an estimate of the full integral.

This approach works because the integrand $e^{-\frac{1}{2}x^2}$ is proportional to the PDF of the standard normal distribution, which is used for sampling. Since the samples are drawn from $\mathcal{N}(0,1)$, the regions where the integrand has significant mass are visited frequently, reducing variance in the estimate. Moreover, the symmetry of both the integrand and the sampling distribution around zero ensures balanced coverage of the domain, and the known normalization constant $\sqrt{2\pi}$ allows us to scale the average function values to recover the full integral.

In this particular run, the integral estimate produces a value of approximately 1.788.

```
%% 4. Monte Carlo integration: Estimating the integral via sampling

% 4.1. Generate random samples from a standard normal distribution
standard_normal_samples = random('Normal',0,1,[N_samples 1]);

% 4.2. Evaluate the unnormalized Gaussian function at each sample
unnormalized_gaussian = exp(-0.5*standard_normal_samples.^2);

% 4.3. MC estimate of the integral
integral_estimate = sqrt(2*pi)*mean(unnormalized_gaussian);
```

5. Set the true value of the integral of the identity function

To examine how the MC estimate converges toward the true value, we will visualize both the estimate and its mean squared error (MSE) across increasing sample sizes. This helps us understand the stability and precision of the approximation as more samples are used. In the code snippet below, we set the true value of the Gaussian integral as a reference point. This value is used to compute the MSE of the Monte Carlo estimates.

```
%% 5. Set the true value of the integral of the identity function

% Set the true value of the integral
integral_true_value = sqrt(pi);
```

6. Monte Carlo estimation error: Mean Squared Error (MSE)

This section reports the accuracy of the MC integration by quantifying its estimation error. Specifically, it computes the MSE, which measures how far the MC estimate deviates from the true value of the integral. The formula used is simply the square of the difference between the estimated and true integral values. The resulting MSE indicates a small error, suggesting that the MC approximation is quite close to the actual value.

```
%% 6. Monte Carlo estimation error: Mean Squared Error (MSE)

% Compute the squared error of the Monte Carlo estimate
MSE = (integral_estimate-integral_true_value)^2;
```

7. Convergence behavior of the MC integral estimate

To analyze how the MC estimate behaves as more samples are added, we compute and visualize both the running estimate and its MSE. The first line computes a running average of the scaled Gaussian evaluations $\sqrt{2\pi} \cdot e^{-\frac{1}{2}x^2}$, where each average serves as a MC estimate of the integral using all samples up to that point. For each sample size n, we take the mean of the first n evaluations, treating them as independent draws from the standard normal distribution, and use this mean to approximate the expected value of the integrand. By the law of large numbers, this cumulative average converges to the true value of the integral $\sqrt{\pi}$ as n increases, since the sample mean becomes increasingly accurate with more data. This sequence of estimates thus reveals the convergence behavior of the MC method in practice.

The second line tracks the MSE by accumulating the squared differences between each scaled estimate and the true value, then normalizing by the sample count.

Together, these metrics reveal how quickly the approximation stabilizes and how the error diminishes with increasing sample size, offering a clear view of the estimator's convergence properties.

```
%% 7. Convergence behavior of the MC integral estimate

% 7.1. Track how the estimate evolves with more samples
convergence_integral_estimate = ...
    cumsum(sqrt(2*pi).*unnormalized_gaussian)./(1:N_samples)';

% 7.2. Track how the mean squared error (MSE) evolves with more samples
convergence_MSE = ...
    cumsum((sqrt(2*pi).*unnormalized_gaussian- ...
    integral_true_value).^2)./(1:N_samples)';
```

8. Convergence behavior of the Monte Carlo integral estimate

MC error typically decreases at a rate proportional to $\mathcal{O}(1/n^{\frac{1}{2}})$ as the number of samples n increases. Here we prepare a reference curve that illustrates this expected decay in error with sample size.

```
%% 8. Convergence behavior of the Monte Carlo integral estimate

% 8.1. Theoretical benchmark
theoretical_error_decay = 1./sqrt(1:N_samples);
```

9. Visualize convergence of the MC estimate and its error

Figure 1 illustrates how the Monte Carlo estimate converges toward the true value $\sqrt{\pi}$ as the sample size increases. Each point on the curve represents the cumulative average of scaled Gaussian evaluations up to that sample count. Notably, the estimate stabilizes quickly and is already reasonably accurate with just 250 samples, demonstrating the efficiency of the method.

Figure 2 shows the behavior of the MSE on a log-log scale as the sample size increases. The MSE curve closely follows the theoretical decay rate of $\mathcal{O}(1/\sqrt{n})$, represented by the reference line $1/\sqrt{n}$. While the MSE may fluctuate above or below this line depending on sampling variability, its overall trend confirms the expected convergence rate of the MC estimator.

```
%% 9. Visualize convergence of the MC estimate and its error

% 9.1. Plot how the Monte Carlo estimate converges to the true value
figure
hold on
plot(1:N_samples,convergence_integral_estimate, ...
    'DisplayName','MC integral estimate');
yline(integral_true_value, ...
    'DisplayName','True integral value');
title('Fig. 1. Convergence of Monte Carlo estimate');
xlabel('Number of samples (draws)');
```

```
ylabel('Integral estimate');
legend('show');
hold off

% 9.2. Plot how the MSE decreases with more samples (log-log scale)
figure
hold on
loglog(1:N_samples,convergence_MSE, ...
    'DisplayName','MSE of MC estimate');
loglog(1:N_samples,theoretical_error_decay.^2, ...
    'DisplayName','Theoretical MSE decay');
title('Fig. 2. Log-log convergence of MC estimation error');
xlabel('Number of samples (log scale)');
ylabel('Mean Squared Error (log scale)');
legend('show');
hold off
```

Fig. 1. Convergence of Monte Carlo estimate

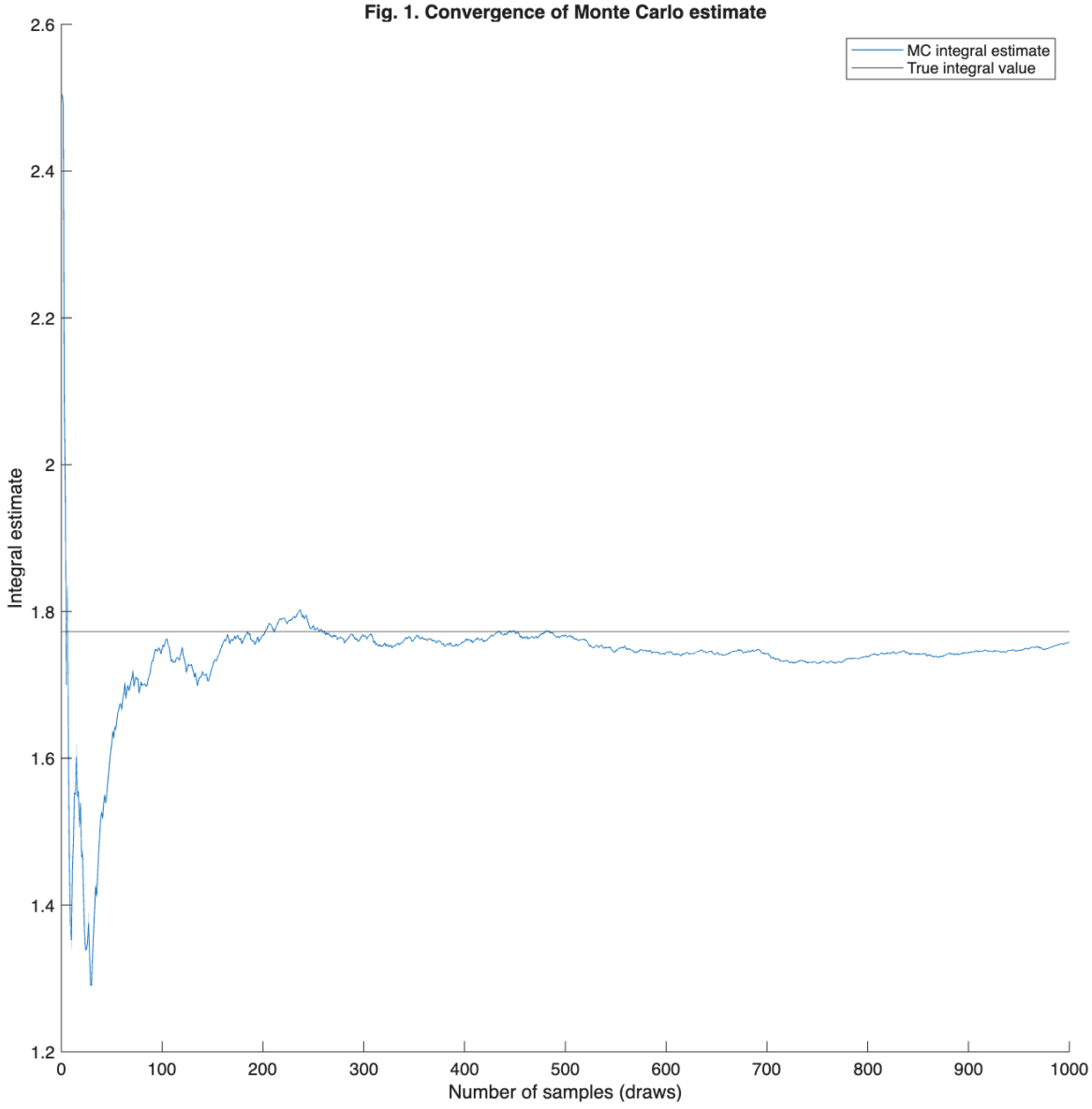
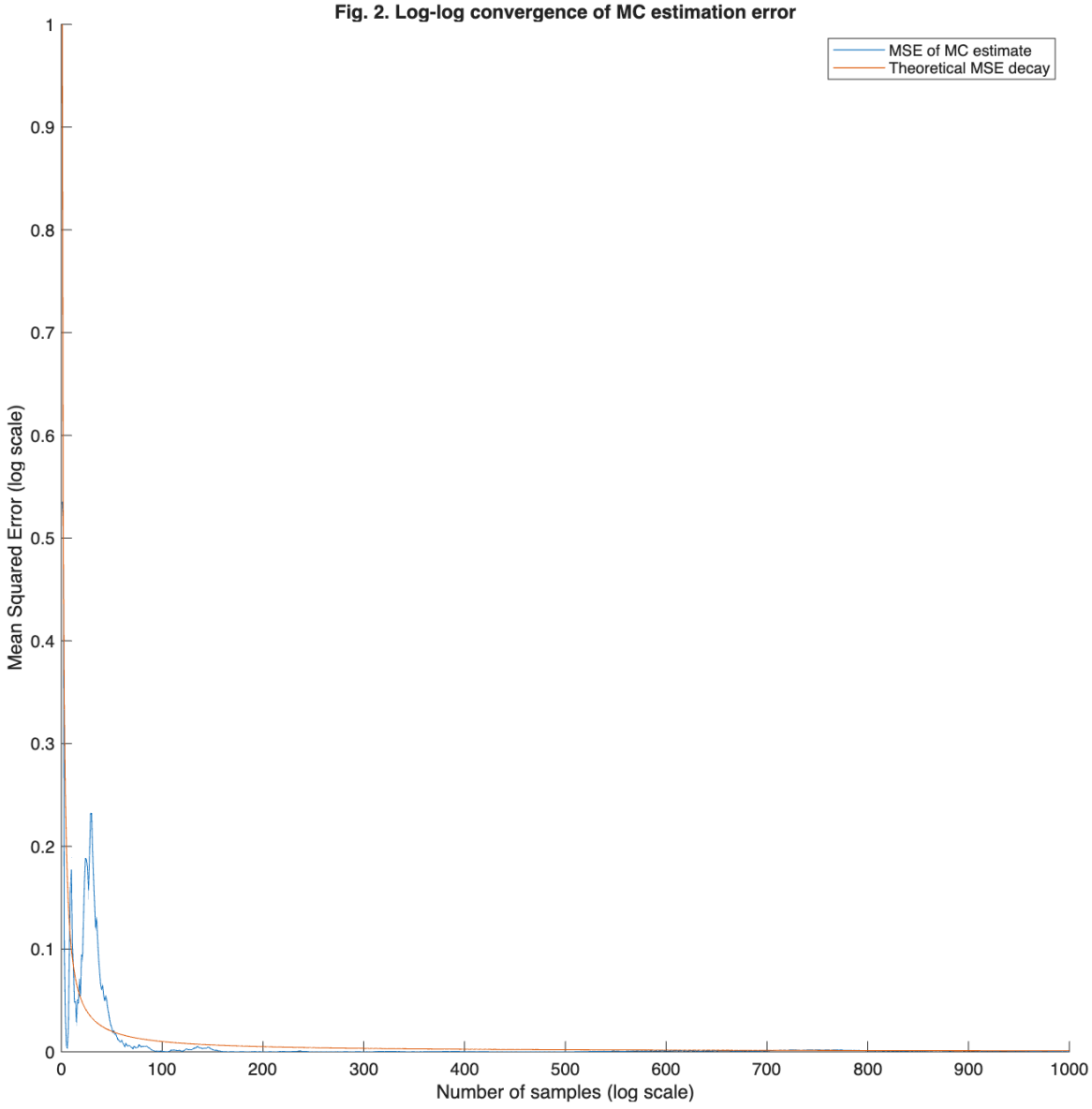


Fig. 2. Log-log convergence of MC estimation error



10. Final notes