Exercise – Understanding Monte Carlo integration using a profit function

1. Aim of the exercise

Monte Carlo (MC) integration is a technique for estimating integrals. in this exercise we use this technique to estimate a profit function.

2. Theory

A thorough treatment of MC theory is covered in the dedicated exercise on MC integration theory. Here, we illustrate the method through an example.

Consider an energy supplier that provides both electricity and gas. The supplier's goal is to approximate total profit over a representative one-year period using MC integration, treating time and daily sales quantities as stochastic inputs. Gas demand is assumed to vary seasonally, while electricity demand remains constant throughout the year. For simplicity, we assume an initial price of 5 euros per day for both products, with 100 units of each available daily.

Let x and y denote the daily quantities sold of gas and electricity, respectively. The profit function $\pi : \mathbb{R}^3 \to \mathbb{R}$, evaluated at a given time t, is defined as

$$\pi(x, y, t) = [P(x, y, t) - C(x, y, t)] \cdot Q(x, y, t),$$

where P, C, and Q represent the price, cost, and demand functions, respectively.

The price function $P : \mathbb{R}^3 \to \mathbb{R}$ is defined as

$$\begin{aligned} P(x, y, t) &= \text{Base price} \cdot \text{Inflation factor} - \text{Price sensitivity} \\ &= 5 \cdot \left(1 + \frac{0.02}{365}\right) - \frac{1}{200}x - \frac{1}{300}y, \end{aligned}$$

where the term $\left(1 + \frac{0.02}{365}\right)$ represents a daily inflation adjustment, referred to as the inflation factor.

The cost function $C : \mathbb{R}^3 \to \mathbb{R}$ is defined as

$$\begin{aligned} C(x, y, t) &= [\text{Fixed costs} + \text{Variable costs}] \cdot \text{Time adjustment factor} \\ &= [2 + 0.015x + 0.01y] \cdot \left(1 + \frac{t}{1000}\right). \end{aligned}$$

The demand function $Q : \mathbb{R}^3 \to \mathbb{R}$ is defined as

$$\begin{aligned} Q(x, y, t) &= [\text{Initial demand} - \text{Projected demand reduction}] \cdot \text{Demand growth factor} \\ &= [80 - 0.05 \cdot x \cdot \rho - 0.08 \cdot y] \cdot e^{\frac{t}{1000}}, \end{aligned}$$

where

$$\rho := \frac{1}{3} \cos\left(\frac{2\pi t}{365} - \frac{\pi}{6}\right) + 1$$

is the seasonal adjustment term. This formulation captures higher demand during winter months relative to summer. Thus, $Q(x, y, t)$ represents the total effective demand across both products, modulated by seasonal and temporal dynamics.

The supplier's goal is to approximate the total annual profit, defined as a deterministic integral over time and daily sales quantities. We define total profit over the year as

$$\Pi := \int_0^{365} \int_0^{100} \int_0^{100} \pi(x, y, t)\, dx\, dy\, dt.$$

We estimate this quantity using random sampling over the domain of time and sales levels.

We begin by defining the random variables $T \sim \text{Unif}(0, 365)$, with PDF

$$f_T(t) = \frac{1}{365} \cdot \mathbb{I}\{t \in [0, 365]\},$$

and $X, Y \sim \text{Unif}(0, 100)$, with PDFs

$$f_X(x) = f_Y(y) = \frac{1}{100} \cdot \mathbb{I}\{x, y \in [0, 100]\}.$$

This formulation allows us to express the total profit Π as the expected value of $\pi(X, Y, T)$, where X, Y, T are random variables uniformly distributed over their respective domains.

Since the PDFs of the uniform distributions over time and sales quantities are constant, meaning each point in the domain is equally likely to be sampled, the integral of the profit function can be expressed as the average value of the function over the domain, scaled by the total volume of that domain. In the representation below, we use lowercase x, y, t to denote integration variables corresponding to realizations of X, Y, T:

$$\Pi = \underbrace{365 \cdot 100 \cdot 100}_{\text{Volume V}} \cdot \int_0^{365} \int_0^{100} \int_0^{100} \pi(x, y, t) \cdot f_X(x) \cdot f_Y(y) \cdot f_T(t)\, dx\, dy\, dt.$$

Note that to compute the expected value of $\pi(X, Y, T)$, we use the law of the unconscious statistician: 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.

Next, generate N i.i.d. samples

$$\theta_i := (X_i, Y_i, T_i) \sim \text{Unif}([0, 100]^2 \times [0, 365]), \quad \text{for } i = 1, \ldots, N.$$

Each sample θ_i represents a random configuration of daily sales quantities and time. The total profit Π can then be approximated via Monte Carlo integration:

$$\Pi \approx \frac{V}{N} \sum_{i=1}^{N} \pi(\theta_i),$$

where $\pi(\theta_i) := \pi(X_i, Y_i, T_i)$ denotes the profit evaluated at sample θ_i. This construction yields an unbiased MC estimator of the total annual profit. Accuracy may be further improved through variance reduction techniques. This approach provides a simulation-based estimate of annual profit under modelled seasonal and pricing dynamics.

3. Simulation setup

Clear the memory. Set the sample size.

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

% 3.1. Clear workspace and memory
clear;

% 3.2. Set number of random samples
N_samples = 1000;
```

4. Monte Carlo sampling of input variables

To estimate annual profit via MC integration, we generate random input configurations over the domain of interest. The integration volume spans 365 days and 100 units each of gas and electricity. We sample daily gas quantities, electricity quantities, and time (day of the year) from continuous uniform distributions over their respective ranges. This preserves the smoothness of the profit function and improves convergence behaviour. Each sampled triplet (x, y, t) serves as an input to the profit function, enabling a probabilistic approximation of the total annual profit.

```
%% 4. Monte Carlo sampling of input variables

% 4.1. Define total volume over time and space
volume_v = 365*100*100;

% 4.2. Generate random x values
x_sample = 100*random('Uniform',0,1,[N_samples 1]);

% 4.3. Generate continuous y values
y_sample = 100*random('Uniform',0,1,[N_samples 1]);

% 4.4. Generate continuous time values (days)
t_sample = 365*random('Uniform',0,1,[N_samples 1]);
```

5. Compute profit components

We begin by defining a compound inflation adjustment to reflect daily price growth. We then calculate the price function by adjusting the base price for inflation and subtracting input-based sensitivities. We continue by defining the cost function, which increases with both input quantities and time. Next, we model seasonal variation in gas demand using a cosine function, which gives us a periodic adjustment over time. Then we compute a growth factor to capture exponential demand increase throughout the year. Using these, we define the quantity function, which depends on both the sampled inputs and seasonal effects. Finally, we determine profit per sample by subtracting cost from price and scaling the result by the effective quantity.

```
%% 5. Compute profit components

% 5.1. Inflation adjustment: compound interest model
inflation = (1+0.02/365).^t_sample;

```

```
% 5.2. Price function: adjusted for inflation and input costs
p_function = 5.*inflation-(1/200).*x_sample-(1/300).*y_sample;

% 5.3. Cost function: increases with inputs and time
c_function = (2+0.015*x_sample+0.01*y_sample).*(1+t_sample./1000);

% 5.4. Seasonal effect: models periodic variation over time
seasonal_effect = (1/3)*cos((2*pi/365).*t_sample-pi/6)+1;

% 5.5. Growth factor: exponential increase over time
growth = exp(t_sample./1000);

% 5.6. Quantity function: depends on inputs and seasonal effects
q_function = (80-0.05.*x_sample.*seasonal_effect-0.08.*y_sample) ...
    .*growth;

% 5.7. Profit per sample: revenue minus cost, scaled by quantity
pi_function = (p_function-c_function).*q_function;
```

6. Estimate total profit

We estimate the total annual profit using Monte Carlo integration by averaging the profit values computed across all sampled configurations. Specifically, we take the mean of the profit values stored in `pi_function` and scale it by the total volume of the integration domain `volumve_v`. This yields an unbiased approximation of the integral representing total profit over the year.

```
%% 6. Estimate total profit

% Monte Carlo estimate of total profit
profit_estimate = volume_v * mean(pi_function);
```

7. Construct confidence interval for integral estimate

Here we construct a 95% confidence interval for our Monte Carlo estimate of total profit over a domain of volume `volume_v`. We begin by computing the sample variance of the profit evaluations, which quantifies variability in the integrand across sampled points. We then derive the standard error of the integral estimate by scaling the square root of the variance over sample size, capturing uncertainty due to finite sampling. Using the normal approximation, we construct the confidence interval, corresponding to 95% coverage under the central limit theorem. These figures summarise both central tendency and uncertainty.

```
%% 7. Construct confidence interval for integral estimate

% 7.1. Compute sample variance of profit evaluations
sample_variance = var(pi_function);

% 7.2. Compute SE of the profit estimate
profit_estimate_SE = volume_v*sqrt(sample_variance/N_samples);

```

```
% 7.3. Construct 95% CI for the profit estimate
CI_lower = profit_estimate-1.96*profit_estimate_SE;
CI_upper = profit_estimate+1.96*profit_estimate_SE;
```

8. Diagnostic: Theoretical MSE of the Monte Carlo integral

We estimate the theoretical mean squared error (MSE) of our MC integral to assess how much uncertainty remains in our approximation. First, we compute the scaled MSE by multiplying the square of the domain volume with the sample variance and dividing by the number of samples. This gives us the expected squared error over the full domain. Next, we calculate the normalized MSE by dividing the sample variance by the sample size, which lets us interpret the error per unit volume. These figures summarise the reliability of our estimate and guide any further refinement or sampling decisions.

```
%% 8. Diagnostic: Theoretical MSE of the Monte Carlo integral

% 8.1. Estimate theoretical MSE of the Monte Carlo integral (scaled)
MSE_scaled = volume_v^2*sample_variance/N_samples;

% 8.2. Estimate normalized MSE (per unit volume)
MSE_normalized = sample_variance/N_samples;
```

9. Final notes

This file is prepared and copyrighted by Jelmer Wieringa and Tunga Kantarcı. This file and the accompanying MATLAB file are available on GitHub and can be accessed using this link.