3. N-dimensional integration using mean value Monte Carlo integration

Use the Monte Carlo mean value method to calculate integrals in 1 to 10 dimensions. Consider the following integral

$$I = \int_{-3}^{3} e^{-x_1^2} \mathrm{d}x_1 \, , \int_{-3}^{3}\int_{-3}^{3} e^{-x_1^2-x_2^2} \mathrm{d}x_1 \mathrm{d}x_2 \, , \int_{-3}^{3}\int_{-3}^{3}\int_{-3}^{3} e^{-x_1^2-x_2^2-x_3^2} \mathrm{d}x_1 \mathrm{d}x_2 \mathrm{d}x_3 \, , \ldots$$

The Monte Carlo mean value method approximates the exact value of the n-dimensional integral as follows.

$$I = \int_{\Omega} f(\bar{x}) \, \mathrm{d}\bar{x} \approx \frac{V(\Omega)}{N} \sum_{i=1}^{N} f(\bar{x_\iota})$$

where N is the number of n-dimensional vectors $\bar{x_\iota}$ that are generated in the space $\Omega \in R^n$ and $V(\Omega)$ is the volume of the space given by the function $f(\bar{x})$.

Discuss your GPU implementation as follows

- compare the outputs and time consumption up to dimension 3 with the built-in functions of Numpy and Scipy packages
- show the time complexity of the implementation up to dimension 10