Crude Monte Carlo method

Student: Giacomo Calabria

Introduction

Using a crude "hit or miss" Monte Carlo method we wanna calculate the volume inside a sphere in different dimensions. The volume delimited by a sphere of radius R in D dimension is

$$V_{sphere} = \int \cdots \int \theta(R^2 - x_1^2 - \cdots - x_N^2) dx_1 \dots dx_N \tag{1}$$

where we defined the theta function as

$$\theta(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x > 0. \end{cases} \tag{2}$$

1 Area of a circle in 2D

Consider a circle of unit radius, $R = 1$, in two dimensions. Generate N_{iter} random numbers using uniform random distribution. Then calculate the probability that point (x, y) lies inside of the circle and use it to approximate the area of the circle S and compare it to the exact results: $S_{extact} = \pi R^2$.
Report the statistical error $\sigma/\sqrt{N_{iter}}$ and compare it with the actual error, then make a log-log plot showing the statistical error and the actual error $|S - S_{exact}|$ as a function of the number of iterations.

We use the following code

```
Niter = 1000
x = 2*np.random.rand(Niter, 2)*R-R
Nhint = 0
for i in range(Niter):
    if x[i,0]**2 + x[i,1]**2 < R**2:
        Nhint += 1
S = 4*R**2*Nhint/Niter
```

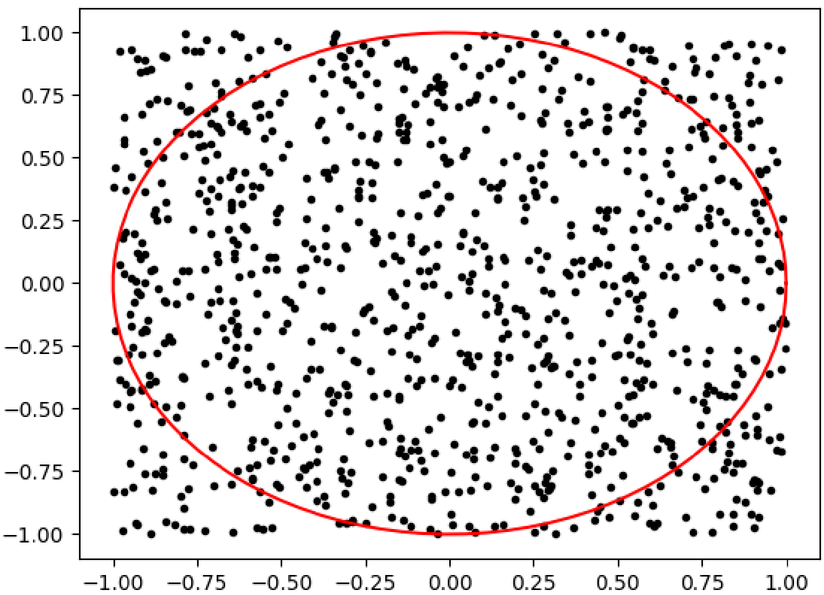

Figure 1

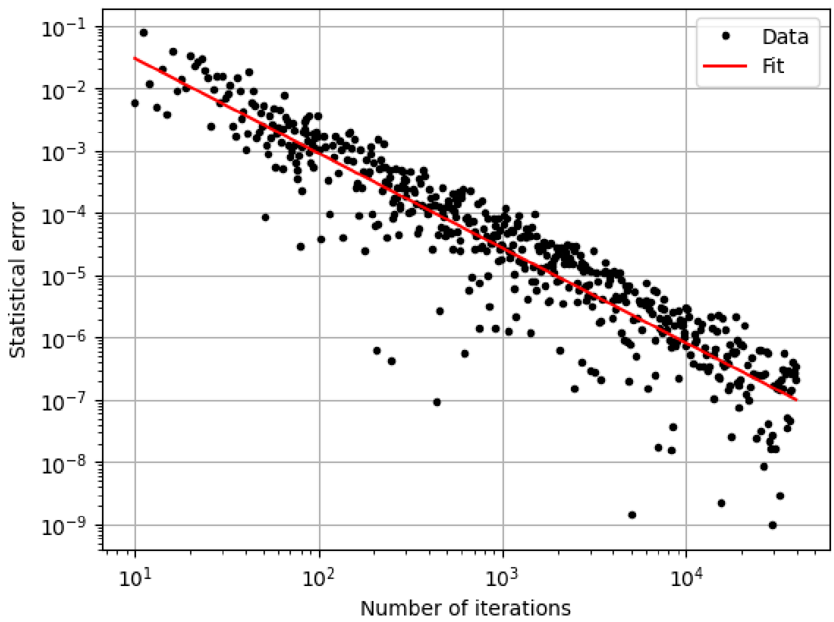

Figure 2: log-log plot of the statistical error

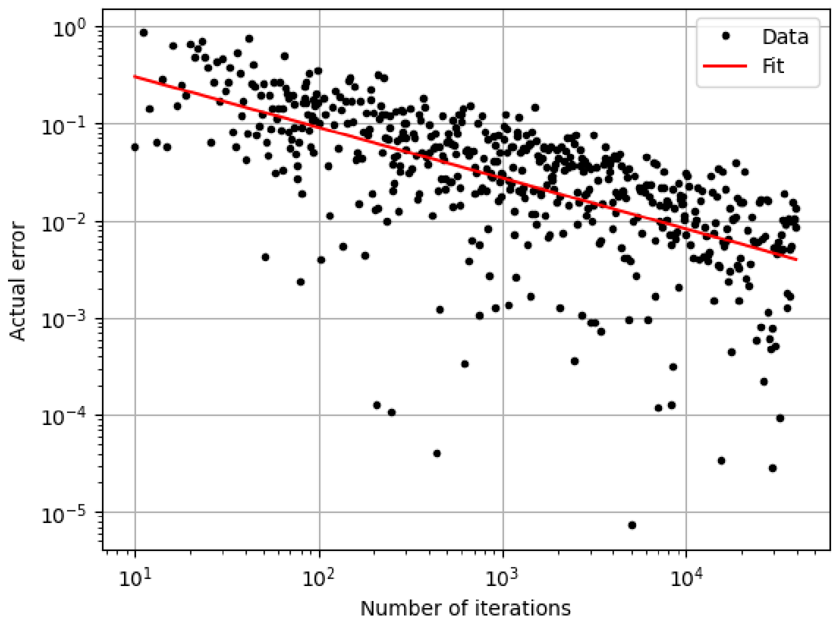

Figure 3: log-log plot of the actual error

2 Volume of a spere in 3D

Consider a sphere of unit radius, $R = 1$, in two dimensions. Generate N_{iter} random numbers using uniform random distribution. Then calculate the probability that point (x, y, z) lies inside of the sphere and use it to approximate the volume of the sphere V and compare it to the exact results: $V_{extact} = \frac{4}{3}\pi R^3$. As before report the statistical error $\sigma/\sqrt{N_{iter}}$ and compare it with the actual error, then make a log-log plot showing the statistical error and the actual error $|S - S_{exact}|$ as a function of the number of iterations.

```
Niter = 100000
for i in range(Niter):
    a = random.randint(1, 6)
    b = random.randint(1, 6)
    dice_outcome[i] = (a + b) / 2
bin_edges = np.arange(1, 7, 0.5)
counts, _ = np.histogram(dice_outcome, bins=bin_edges)
pdf = counts / Niter
plt.bar(bin_edges[:-1], pdf, width=0.5)
```

images/Figure2.png

Figure 4: Distribution of the average value of two dice roll

3 Volume of a sphere in D dimensions

We now consider the follow random event: throwing a dice N_{iter} times and calculating the average value.

$$x = \frac{\sum_{i=1}^{N_{iter}} \mathrm{rand}(6)}{N_{iter}} \tag{3}$$

With that random event x we calculate the probability distribution $p(x)$. As we use large N we can consider the random value x as a continuous variable. And then we normalise the PDF as $\int p(x)dx = 1$. With this code we generate calculate the probability distribution:

```
Ndice = 100000
Niter = 10000
def random_event():
    return np.sum([random.randint(1, 6) for _ in range(Niter)])/Niter

random_outcome = [random_event() for _ in range(Ndice)]
hist, bins = np.histogram(random_outcome, bins=50, density=True)
bin_centers = (bins[:-1] + bins[1:]) / 2
plt.plot(bin_centers, hist, label='Computed distribution')

x = np.linspace(min(random_outcome)-0.1, max(random_outcome)+0.1, 1000)
gaussian = norm.pdf(x, loc=3.5, scale=np.std(random_outcome))
plt.plot(x, gaussian, label='Gaussian distribution',color='red', linestyle='--')
```

images/Figure3.png

Figure 5: Distribution of the average value of multiple dice roll

As we can see as we use a large number of dice the resulting distribution is pretty similar to the Gaussian distribution expected by the Central Limit Theorem

4 Error estimation

We wanna now calculate the average value and estimate the statistical error, associated with such estimation. Assume that single-die throwing is used to estimate the mean value and the variance

$$\mu = \langle x \rangle \approx \frac{\sum_{i=1}^{N_{iter}} x_i}{N_{iter}} \tag{4}$$

$$\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2 \approx \frac{\sum_{i=1}^{N_{iter}} x_i^2}{N_{iter}} - \left(\frac{\sum_{i=1}^{N_{iter}} x_i}{N_{iter}} \right)^2 \tag{5}$$

Calculate the mean value by throwing the dice $N_{iter} = 10$ and $N_{iter} = 100$ times and compare the estimation of the mean value and the variance with the exact values, given by

$$\mu = \langle x \rangle = \frac{\sum_{\ell=1}^{6} \ell}{6} = 3.5 \quad \sigma^2 = \frac{\sum_{\ell=1}^{6} \ell^2}{6} - \left(\frac{\sum_{\ell=1}^{6} \ell}{6} \right)^2 = \frac{35}{12} \approx 2.92$$

Using the following code

```
Niter = 10
dice_rolls = np.array([random.randint(1, 6) for _ in range(Niter)])

u = np.sum(dice_rolls)/Niter
var = np.sum(dice_rolls**2)/Niter-np.mean(dice_rolls)**2
```

we obatin the results, which are reported in Table 1

N_{iter}	μ	σ^2	ε
10			
100			
1000			

Table 1: Results of error estimation