Problem Set 3

Dario Loprete

September 24, 2024

GitHub link

`https://github.com/dariolop76/phys-ga2000`

1 Problem 1

For Problem 1, we deal with the computation time needed to evaluate the matrix product

$$C = A \text{ x } B, \tag{1}$$

where A, B and C are N x N matrices. In particular, we are asked to compare two methods: one uses three nested for loops to implement

$$C_{ij} = \sum_k A_{ik} B_{kj}, \tag{2}$$

while the other makes use of the function `numpy.dot`. In the code, we use the package `timeit` in order to compute the time needed to perform these operations. Additionally, we are also asked to check if the computation time in the case of a for loop goes as N^3. Figure 1 shows the obtained result, namely the computation time versus the matrix size N, for values of N in the range $[10, 120]$.

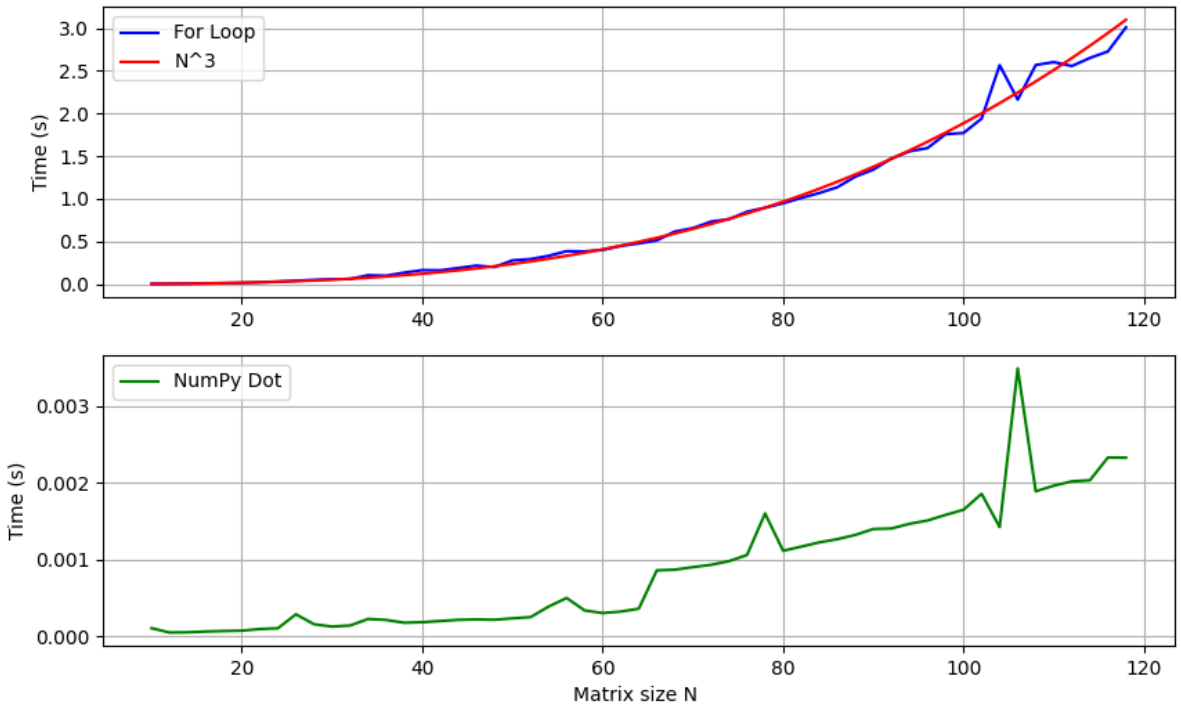

Figure 1: Computation time vs matrix size. Top: case of for loops, and expected behaviour (N^3); bottom: case of `numpy.dot`.

As we can see from the plots, the execution time using for loops rises as N^3 and it is less efficient to follow this method than to use `numpy.dot`.

2 Problem 2

Given the instructions and theory of Exercise 10.2, we have simulated the decay of ^{213}Bi to ^{209}Bi. The main for loop in the code takes into consideration the probabilities of the decays $^{213}Bi \rightarrow {}^{209}Tl$ and $^{213}Bi \rightarrow {}^{209}Pb$ by applying the same methodology, namely by simulating the decay when a certain random number is less than the given probability. The obtained results are plotted in Figure 2.

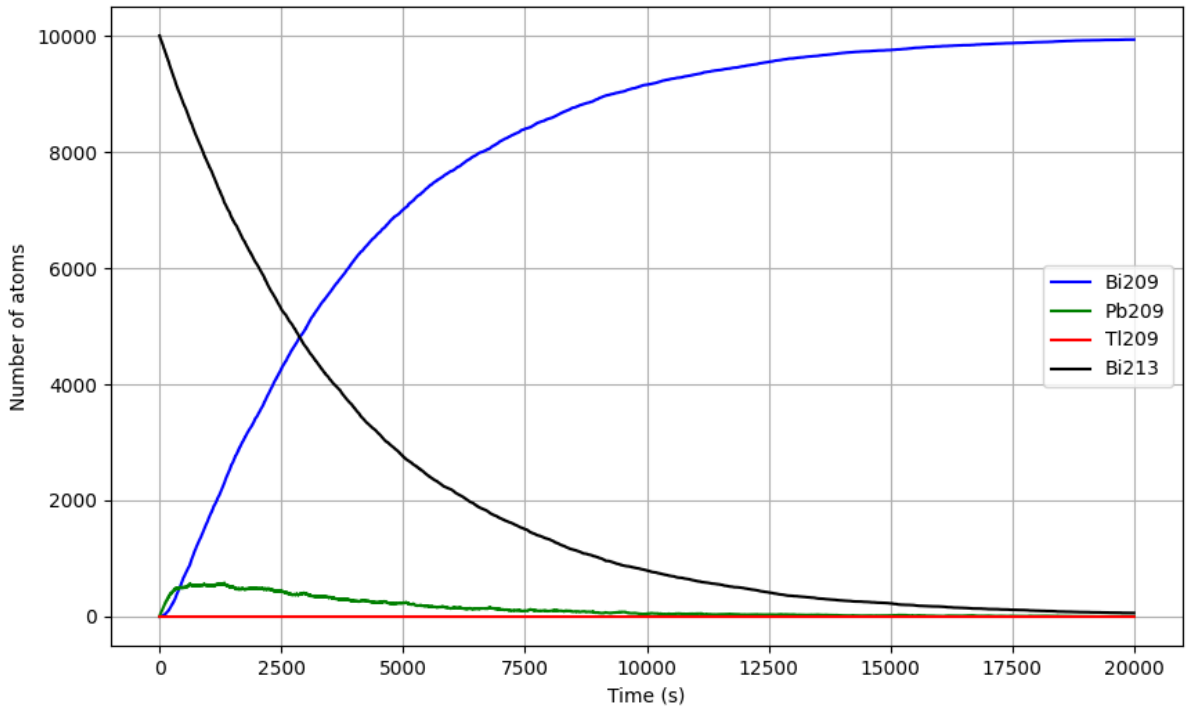

Figure 2: Decay of a sample of ^{213}Bi. The curves show the number of atoms involved in the decay chain as a function of time.

As we would expect from looking at the values of the probability for the decays $^{213}Bi \rightarrow {}^{209}Tl$ and $^{213}Bi \rightarrow {}^{209}Pb$, the amount of thallium produced is very small compared to the other elements. Indeed, only 2.09% of the initial ^{213}Bi decays to ^{209}Tl, which, in turn, decays to lead quite fastly. In regards to the amount of lead produced, this has a peak at the beginning. As time increases, the number of ^{213}Bi atoms decreases, and so does that of ^{209}Pb, while the amount of ^{209}Bi increases, until all there is left are ^{209}Bi atoms.

3 Problem 3

Exercise 10.4 asks to replicate the results of Example 10.1, but this time with a different method. Instead of simulating the decay by comparing a uniformly distributed random number with a certain threshold and keeping track of the number of decayed atoms at each step, we can use a more efficient approach. We can associate to each decaying ^{208}Tl atom a random number that follows a non uniform distribution. This number represents a random time at which the corresponding atom decays. Then, given the ensemble of decay times, we can keep track of the number of atoms which

have decayed by comparing their (random) decay time t to the atom's half life τ. If $t > \tau$, it means that that atom has not yet decayed. This approach makes use of the transformation method, which allows to go from uniform random numbers to non uniform random numbers. In particular, the relation used for this exercise is the following:

$$x = -\frac{1}{\mu} \log(1 - z), \tag{3}$$

where z is a uniformly distributed random variable, while x follows the distribution

$$p(x) = \mu e^{-\mu x}. \tag{4}$$

In our case, namely a radioactive decay problem, we set $\mu = \log(2)/\tau$. Figure 3 shows the result obtained following this approach.

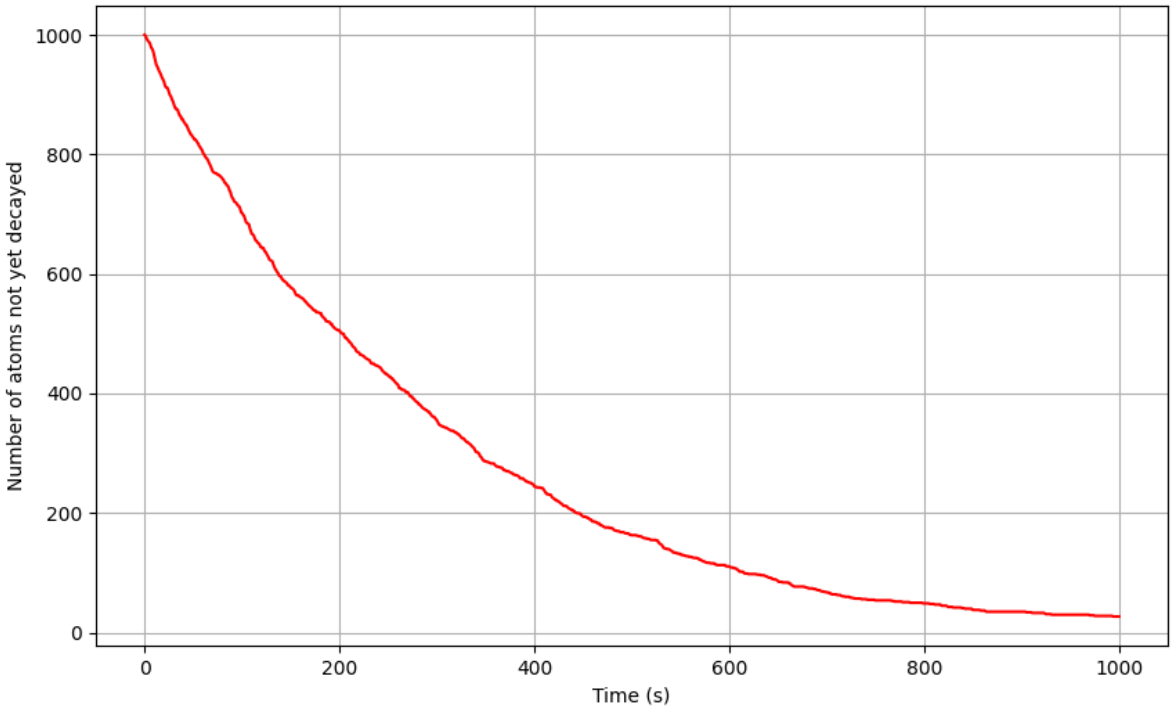

Figure 3: ^{213}Bi atoms which have not yet decayed to ^{208}Pb as a function of time.

The obtained plot is in agreement with that of Example 10.1.

4 Problem 4

Problem 4 aims at a visual representation of the Central Limit Theorem (CLT). For this specific case, we need to generate a set of N i.i.d. random variables x_i, whose distribution is given by

$$p(x) = e^{-x}, \tag{5}$$

and show that for large N the distribution of the variable

$$z = \sqrt{N}(y - \mu) \tag{6}$$

approaches a Gaussian distribution with mean = 0 and variance σ^2, where σ^2 and μ are, respectively, the variance and the mean of the x_i, while y is the sample average, namely $y = \frac{1}{N}\sum_{i=1}^{N} x_i$.

4.1 Computation of μ and σ

$$\mu = E[x] = \int_0^{+\infty} dx\ xe^{-x} = \tag{7}$$

$$= (-xe^{-x})\Big|_0^{+\infty} + \int_0^{+\infty} dx\ e^{-x} = \tag{8}$$

$$= -e^{-x}\Big|_0^{+\infty} = 1 \tag{9}$$

$$E[x^2] = \int_0^{+\infty} dx\ x^2 e^{-x} = \tag{10}$$

$$= (-x^2 e^{-x})\Big|_0^{+\infty} + 2\int_0^{+\infty} dx\ xe^{-x} = 2 \tag{11}$$

$$Var(x) = E[x^2] - E[x]^2 = 2 - 1 = 1 \tag{12}$$

$$\sigma = \sqrt{Var(x)} = 1 \tag{13}$$

4.2 Visual representation of CLT

In the code `prob_4_1.py`, we generate $M = 10000$ samples of N random variables x_i and compute the corresponding values of y and z. We do this for 4 different values of $N \in \{5, 10, 50, 200\}$. Figure 4 shows the distribution of the variable z for each case.

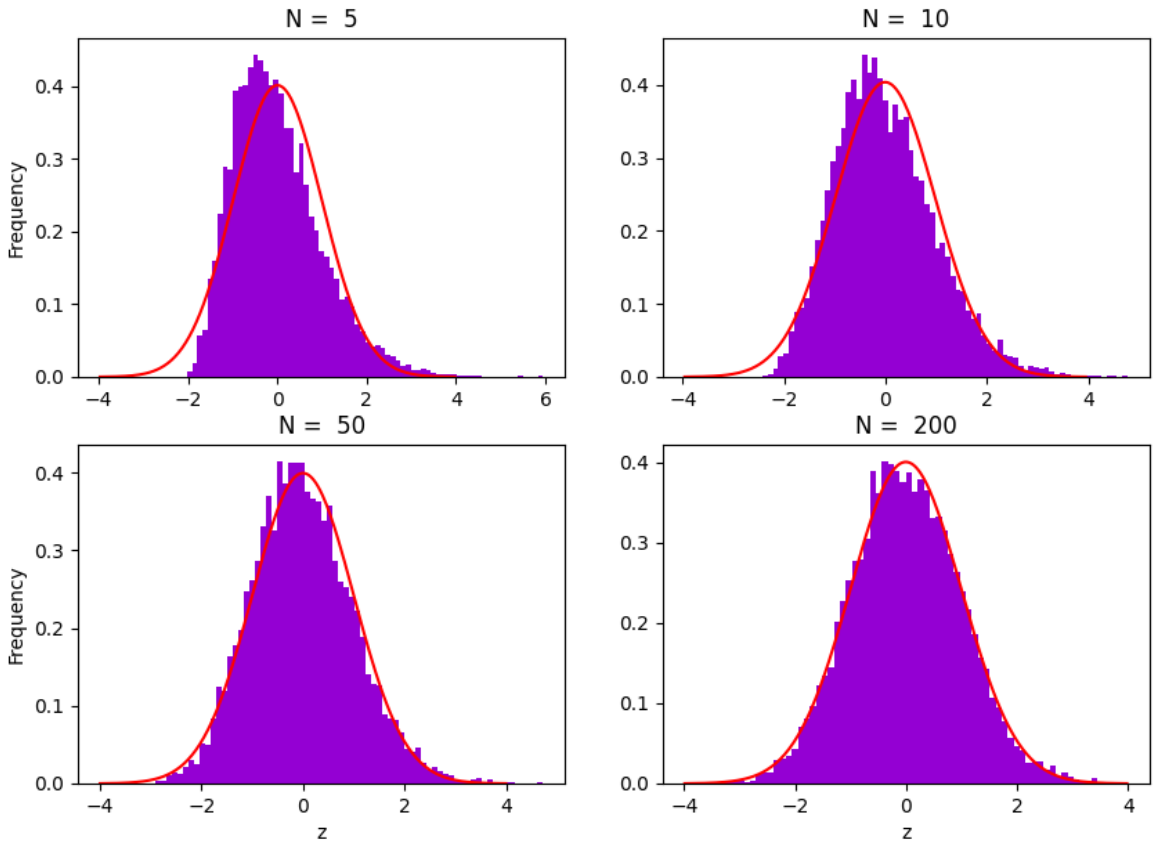

Figure 4: Distribution of the variable z for different values of N. In red, the Gaussian distribution centered at 0 with $\sigma = 1$ is plotted.

We can see that, as N increases, the distribution tends to a Gaussian centered at 0 with $\sigma = 1$. Therefore, Figure 4 provides a visual representation of the CLT.

4.3 Mean and variance of y

$$E[y] = E\Big[\frac{1}{N}\sum_{i=1}^{N} x_i\Big] = \tag{14}$$

$$= \frac{1}{N}\sum_{i=1}^{N} E[x_i] = \frac{1}{N}\sum_{i=1}^{N} 1 = \tag{15}$$

$$= \frac{N}{N} = 1 \tag{16}$$

$$Var[y] = Var\Big[\frac{1}{N}\sum_{i=1}^{N} x_i\Big] = \tag{17}$$

$$= \frac{1}{N^2}\sum_{i=1}^{N} Var[x_i] = \frac{1}{N^2}\sum_{i=1}^{N} 1 \tag{18}$$

$$= \frac{N}{N^2} = \frac{1}{N} \tag{19}$$

4.4 Moments of the distribution of y

In the code `prob_4_2.py`, we compute the mean, the variance, the skewness and the kurtosis of the distribution of the variable y for values of N in the range $[1, 1000]$. The mean ($\bar{y}$) and the variance are computed using the NumPy functions `numpy.mean` and `numpy.var`, while for the skewness and the kurtosis we use the following estimators:

$$s = \frac{\frac{1}{M}\sum_{i=1}^{M}(y_i - \bar{y})^3}{(\frac{1}{M}\sum_{i=1}^{M}(y_i - \bar{y})^2)^{3/2}} \tag{20}$$

$$k = \frac{\frac{1}{M}\sum_{i=1}^{M}(y_i - \bar{y})^4}{(\frac{1}{M}\sum_{i=1}^{M}(y_i - \bar{y})^2)^2} - 3 \tag{21}$$

Figure 5 shows the obtained values as a function of N.

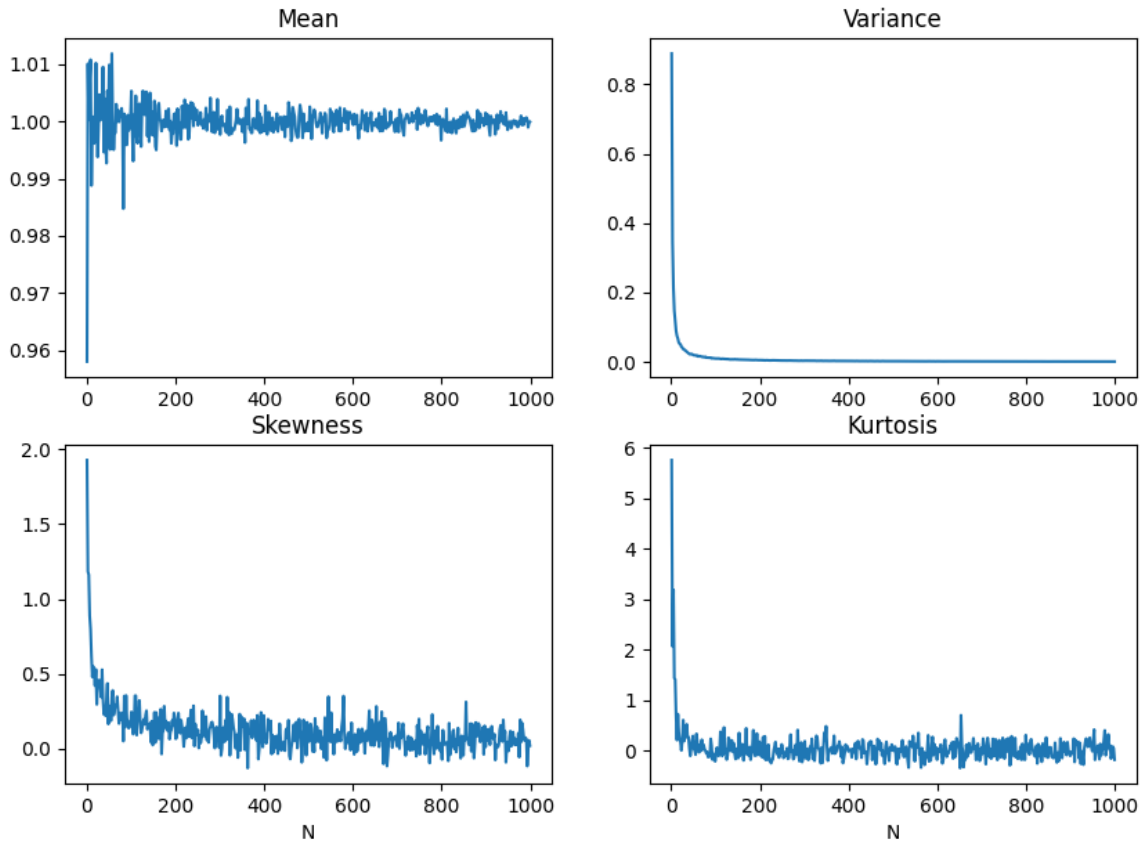

Figure 5: Mean, variance, skewness and kurtosis for the variable y as a function of N.

The behaviour is the expected one: the mean remains equal to 1 (up to fluctuations), while the other three moments go to zero as N increases. Additionally, the values of skewness and kurtosis for $N = 1$ correspond to those of the exponential distribution, namely 2 and 6.

Finally, the code also provides an estimate of the value of N such that the skewness (N_s) and the kurtosis (N_k) reach about 1% of their value at $N = 1$:

$N_s = 109$
$N_k = 23$