HACETTEPE UNIVERSITY

Department of Nuclear Engineering

NEM393 - ENGINEERING PROJECT II
Project Assignment II

Student Name : Enes Mert Ulu

Delivery Date : 18 November 2023
Due Date : 19 November 2023

1. INTRODUCTION

In this study, the fission fragments produced in the fuel region by fission are investigated. The fuel region is spherical with a radius of $R_f = 0.16\,mm$ and with an outer radius of $R_{shell} = 0.18\,mm$.

The white region is the fuel region which is the centre of the sphere. The blue-shaded region is the shell region. In every fission, two fission fragments are produced in opposite directions at random points in the fuel region as shown in Figure 1.

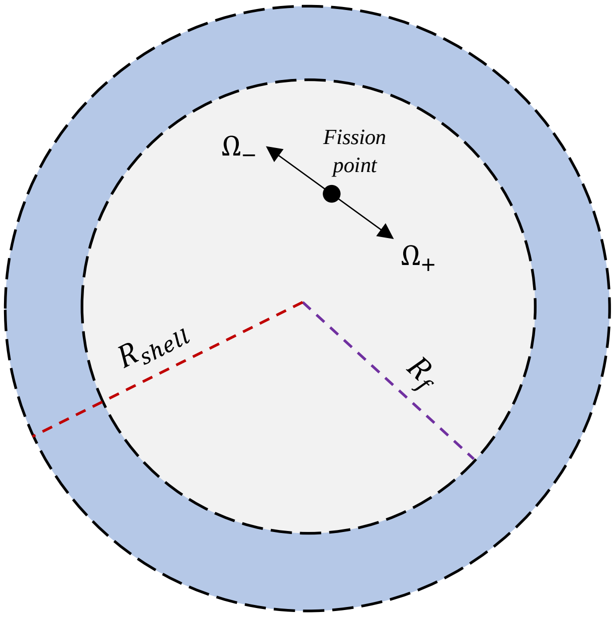


Figure 1 – The Sphere Region and the Fission Event with Fission Products

The distance taken by the fission fragments is denoted with S. While $S \geq 0$ probability density function is given as:

$$P(S) = \frac{0.7}{l} \exp\left(-\frac{S}{l}\right) + 0.3\delta(S - d) \qquad eq.1$$

where $l = 0.1\,mm$ and $d = 0.05\,mm$

In this study, the main purpose is to virtualise the distribution of fission fragments for $10^4, 10^6$ and 10^8 fissions by using Monte Carlo Algorithm.

2. METHODS AND CALCULATIONS

In this study, the Monte Carlo Algorithm is used to find the fractions of the fission fragments that escaped from the fuel region and absorbed fragments in the shell region.

The calculations are done by using the Monte Carlo Algorithm and Python 3.11 with libraries *NumPy, math* and *random*.

The principle of the Monte Carlo Algorithm is to choose random values and to generate random results, at the end of the calculations the results converge to a value.

The Monte Carlo Algorithm can be given as:

$$\frac{\int_0^r 4\pi r'^2 dr'}{\int_0^R 4\pi r'^2 dr'} = \xi \qquad eq.2$$

$where\ \xi\ is\ a\ uniform\ random\ variable\ with\ a\ domain\ of\ \xi\ \epsilon$ [0,1].

ξ values are generated by Python randomly. With these random ξ values, the location of r is calculated as:

$$r = z = R_f \xi^{\frac{1}{3}} \qquad eq.3$$

To simplify it, the symmetry condition can be used as:

$$z = z_0$$
$$x = 0$$
$$y = 0$$

(x, y, z_0) can be taken as centre of the event.

After that to determine the direction of fission fragments, following algorithm is used:

$$\mu = cos\theta = 2\xi - 1 \qquad eq.4$$

$$\varphi = 2\pi\xi \qquad eq.5$$

$$\Omega_x = (1 - \mu^2)^{\frac{1}{2}} cos\varphi \qquad eq.6$$

$$\Omega_y = (1 - \mu^2)^{\frac{1}{2}} sin\varphi \qquad eq.7$$

$$\Omega_z = \mu \qquad eq. 8$$

Then for the first fission fragment μ is taken as eq.4. However, for the second fission fragment μ is taken as $-\mu$. Because the fission fragments go to the opposite directions.

After that, the latest location of the first fission fragment is given as:

$$\underline{\Omega_1} = \Omega_x \underline{i} + \Omega_y \underline{j} + \Omega_z \underline{k} \qquad eq. 9$$

and for the second fission fragment, it can be written as:

$$\underline{\Omega_1} = \underline{-\Omega_2} \qquad eq. 10$$

Then, to calculate the distance of fission fragments travelled, S, following algorithm is used:

$$\boldsymbol{\xi < 0.3 \qquad S = d \qquad eq. 11}$$

$$\xi > 0.3 \qquad S = -l \log \xi \qquad eq. 12$$

In every step, different ξ values are used.

If S is known, the location of the fission fragment might be determined as shown in eq.13, eq.14, and eq.15.

$$r_0 = \left(x_0 = 0, \qquad y_0 = 0, \qquad z_0 = R_f \xi^{\frac{1}{3}} \right) \qquad eq. 13$$

$$\underline{r_0} + \underline{\Omega} S = \underline{r_{new}} \qquad eq. 14$$

$$\underline{r_{new}} = \{ \Omega_x S \underline{i}, \qquad \Omega_y S \underline{j}, \qquad R_f \xi^{\frac{1}{3}} + \Omega_z S \underline{k} \} \qquad eq. 15$$

If $\underline{r_{new}}$ is outside of the sphere, fission fragment reached to the outer boundary of fuel, else it stayed at the fuel region.

After that, we must find out the closest distance to reach to the fuel boundary. It can be found as:

$$\boldsymbol{\left(x_b^2 + y_b^2 + z_b^2 \right) = R_f^2 \qquad eq. 16}$$

$$x_b^2 = (x_0 + \Omega_x d_s)^2$$
$$y_b^2 = (y_0 + \Omega_y d_s)^2$$
$$z_b^2 = (z_0 + \Omega_z d_s)^2$$

By substituting these to eq.16, eq.17 can be obtained.

$$\Omega_x^2 d_s^2 + \Omega_y^2 d_s^2 + z_0^2 + 2\Omega_z d_s z_0 + \Omega_z^2 d_s^2 = R_f^2 \qquad eq. 17$$

then,

$$d_s^2 + 2\mu z_0 d_s + z_0^2 - R_f^2 = 0 \qquad eq. 18$$

Since this is a quadratic equation,

$$\boldsymbol{a = 1}$$

$$\boldsymbol{b = 2\mu z_0}$$

$$\boldsymbol{c = z_0^2 - R_f^2}$$

and

$$\boldsymbol{d_{s1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}} \qquad \boldsymbol{eq. 19}$$

Positive root of eq.19 is the shortest distance to the fuel surface, and it is denoted by d_s^*.

According to the results, if $S < d_s^*(R_f)$ fission fragments lost all of its energy in the fuel region, if $S > d_s^*(R_f)$ fission fragment escape from the fuel region.

For the shell, we need to determine the shortest distance to the shell surface, and to obtain it following equation must be solved:

$$\Omega_x^2 d_s^2 + \Omega_y^2 d_s^2 + z_0^2 + 2\Omega_z d_s z_0 + \Omega_z^2 d_s^2 = R_{shell}^2 \qquad eq. 20$$

$$\boldsymbol{a = 1}$$

$$\boldsymbol{b = 2\mu z_0}$$

$$\boldsymbol{c = z_0^2 - R_{shell}^2}$$

$$d_{s1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad eq.21$$

It is also d_s^* is the shortest distance to the shell.

According to the results, if $S < d_s^*(R_{shell})$ fission fragment lost all of its energy in the fuel region and the shell region if $S > d_s^*(R_{shell})$ fission fragment escape from the fuel and the shell region.

3. RESULTS

The Monte Carlo Algorithm is run 4 times by Python and the results are given in the Table 1 below.

The code is run 4 times because it is wanted to see the value that the value converges.

Table 1 – Escape Probability from the Fuel Region

Number of Fission	First Output	Second Output	Third Output	Fourth Output
10	0.00%	20.00%	50.00%	50.00%
10^2	19.00%	19.00%	34.00%	28.00%
10^3	21.10%	21.30%	23.30%	23.70%
10^4	22.98%	21.78%	23.26%	23.49%
10^5	23.23%	22.93%	23.20%	22.95%
10^6	22.95%	23.00%	23.03%	22.93%
10^7	22.96%	22.97%	22.96%	22.95%
10^8	22.96%	22.96%	22.96%	22.96%

As seen in Table 1, the values are converging a value.

Table 2 – Average Escape Probability from the Fuel Region per Fission

Number of Fission	The Average Values
10	30.00%
10^2	25.00%
10^3	22.35%
10^4	22.88%
10^5	23.08%
10^6	22.98%
10^7	22.96%
10^8	22.96%

As shown, the escape probability values are converging some values.

By using the Monte Carlo Algorithm, we virtualise the same event for several times and our values are started to converge a significant value as can be seen in Table 2 above.

The same calculations are done for the absorption probability from the shell region. These values are given in the Table 3 below.

Table 3 – Absorption Probability from the Shell Region

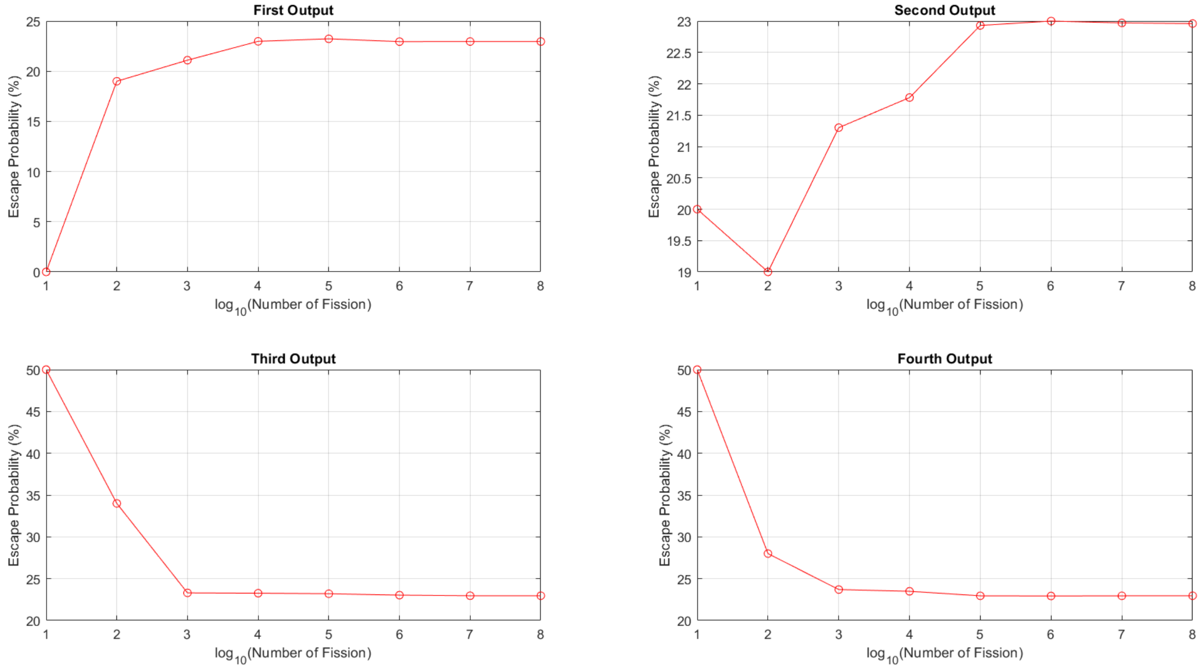
Number of Fission	First Output	Second Output	Third Output	Fourth Output
10	0.00%	20.00%	10.00%	10.00%
10^2	7.00%	11.00%	14.00%	9.00%
10^3	9.30%	9.70%	9.70%	10.20%
10^4	10.29%	10.09%	10.89%	10.87%
10^5	10.83%	10.71%	10.72%	10.68%
10^6	10.65%	10.69%	10.67%	10.61%
10^7	10.65%	10.63%	10.64%	10.66%
10^8	10.64%	10.64%	10.64%	10.64%

Table 4 – Average Absorption Probability from the Shell Region per Fission

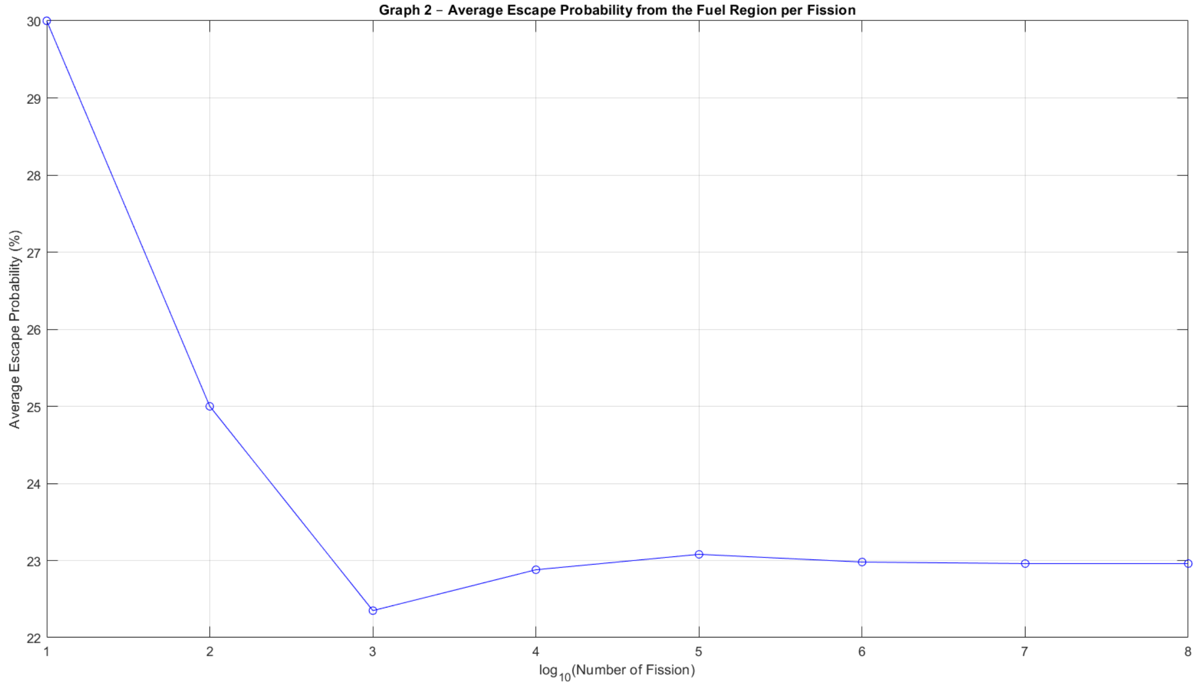
Number of Fission	The Average Values
10	10.00%
10^2	10.25%
10^3	9.72%
10^4	10.54%
10^5	10.74%
10^6	10.66%
10^7	10.65%
10^8	10.64%

In addition to these tables, corresponding graph are plotted as shown below.

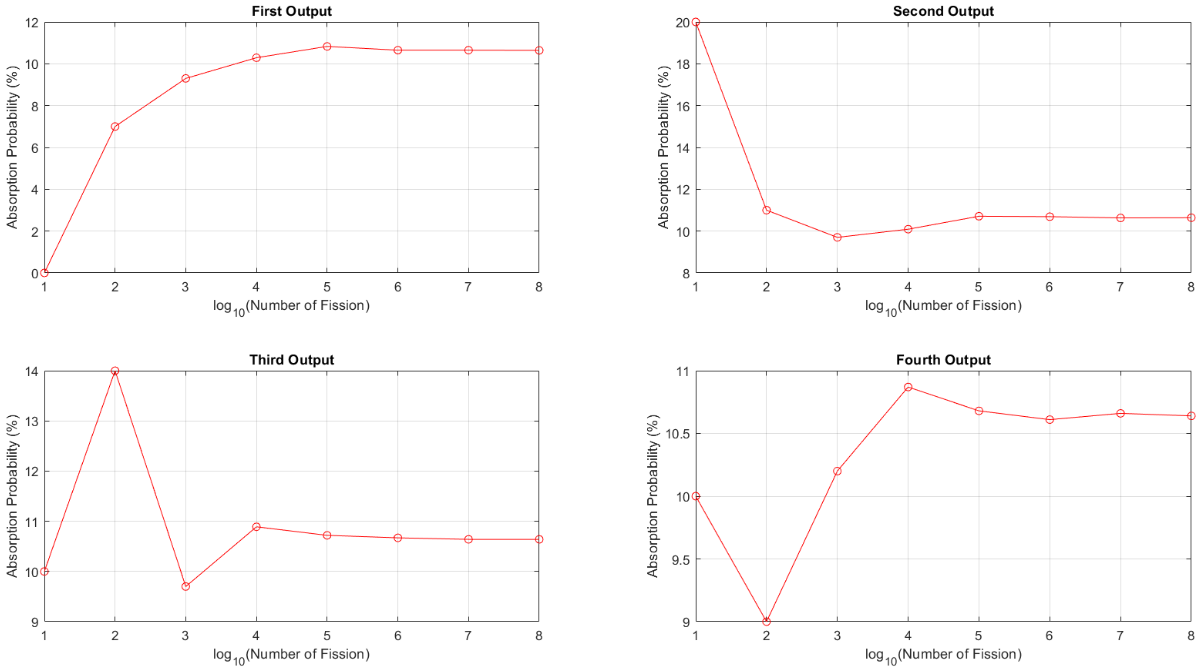
Graph 1 – Escape Probability from the Fuel Region



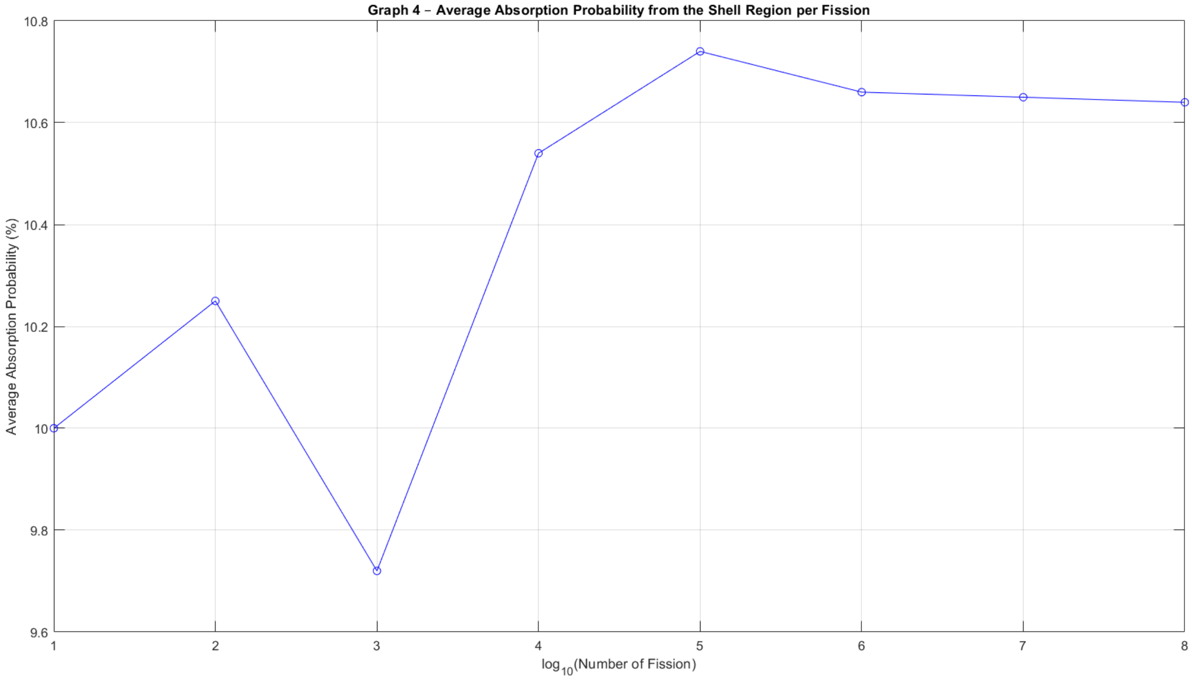
Graph 2 – Average Escape Probability from the Fuel Region per Fission



Graph 3 – Absorption Probability from the Shell Region



Graph 4 – Average Absorption Probability from the Shell Region per Fission



In addition to these, detailed particle amounts are given in appendix.

4. CONCLUSION

To sum up, the Monte Carlo Algorithm is used to analyse the distribution of fission fragments. It is clearly seen that even if the numbers may change in order the fission number at the end of the process, they converge to a significant number step by step. This shows the balance in a nuclear power reactor. It is also clear to see that fission fragments go to the opposite direction and some of them may stuck on the fuel region while the other pair escape to the shell region.

However, this complexness creates a harmony and at the end of the process this sophistication comes through an array.

5. REFERENCES

[1] (n.d.). *Python Random Module*. W3schools. Retrieved November 13, 2023, from https://www.w3schools.com/python/module_random.asp

[2] (n.d.). *Math - Mathematical functions*. Docs.Python.org. Retrieved November 14, 2023, from https://docs.python.org/3/library/math.html

[3] (n.d.). *NumPy Introduction*. W3schools. Retrieved November 14, 2023, from https://www.w3schools.com/python/numpy/numpy_intro.asp

6. APPENDIX

[1] Python Code: NEM393-Project-II- Appendix [1] Python Code.py
[2] First Output: NEM393-Project-II- Appendix [2] Output 1.txt
[3] Second Output: NEM393-Project-II- Appendix [3] Output 2.txt
[4] Third Output: NEM393-Project-II- Appendix [4] Output 3.txt
[5] Fourth Output: NEM393-Project-II- Appendix [5] Output 4.txt