NEUTRON DIFFUSION

Theoretical and Numerical Aspects of Nuclear Physics Presentation

Riccardo Milioli

We want to study the diffusion of neutrons in fissile material (in particular we assume to work with U-235) with different underlying geometries, where collisions between free neutrons and nuclei result in the release of secondary neutrons. The result is a chain reaction that can lead to an intense explosion.
Our goal is to find the size at which criticality occurs with calculations based on Dirichlet boundary conditions (where the neutron density is exactly zero at the boundary, so neutrons don't escape). This is a simplified model with respect to the case of the more physical Neumann boundary conditions, which allow for neutron escape.
For the 3D case in spherical coordinates, we will perform the analysis with both boundary conditions.

Diffusion Equation

We start from the diffusion equation with a source term since the fissile material acts a source of neutrons:

$$\frac{\partial n}{\partial t} = \mu \nabla^2 n + \eta n.$$

Where n, μ, η are the neutron density, the diffusion constant and the neutron rate of fromation respectively.
We solve this equation by variable separation method, that is, if we are just in $1D$, assuming the solution is of the form:

$$n = T(t)X(x).$$

This allows us to separate the PDE into 2 ODEs which are much easier to solve:

$$\frac{1}{T}\frac{\partial T}{\partial t} - \eta = \frac{\mu}{X}\frac{\partial^2 X}{\partial x^2} = -k.$$

Where k is the separation constant which can be determined by imposing the boundary conditions.
The solutions to these equations are:

$$T(t) = A_1 e^{(\eta - k)t} \quad \text{and} \quad X(x) = A_2 e^{i\sqrt{\frac{k}{\mu}}x}.$$

This method can be easily generalized to a higher number of dimensions.

1D Cartesian Coordinates

For the $1D$ case in cartesian coordinates, the diffusion equation is:

$$\frac{\partial n}{\partial t} = \mu \frac{\partial^2 n}{\partial x^2} + \eta n.$$

With the method described previously and making use of the superposition principle, since the diffusion equation is linear, one finds the following neutron density:

$$n(t,x) = \sum_{p=1}^{\infty} a_p \, e^{(\eta - \mu(\frac{p\pi}{L})^2)t} \sin\left(\frac{p\pi x}{L}\right).$$

The boundary conditions have already been imposed to obtain this form.
L is the length of our system.

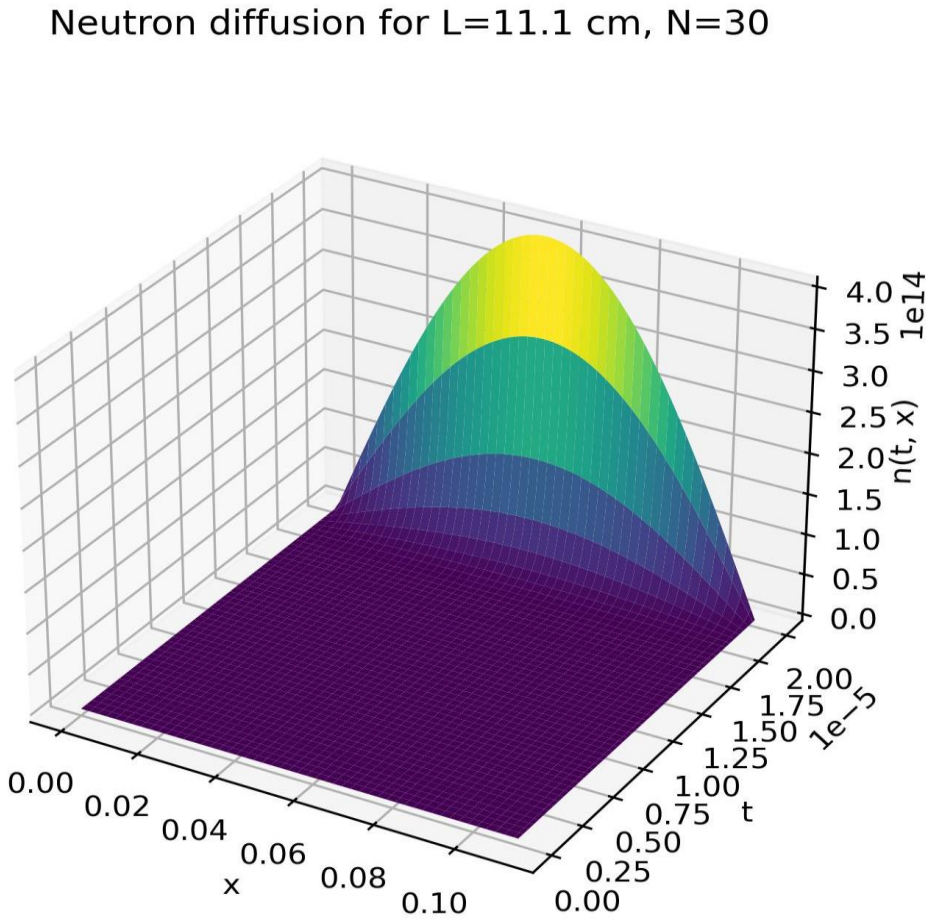
It's clear that the density will increase unbounded only if the argument of the exponential is positive. This leads us to the value of the critical length, that is the smallest length of the fissile material needed for a sustained chain reaction, which is obtained when $p = 1$. Using the data for U-235 one finds: $L_{crit} = 11.05\ cm$. We will perform the simulation with $L = 11.1\ cm$ (supercritical state: the fission proceeds at increasing rate).
To simulate the evolution of the neutron density we only need the coefficients a_p, which can be found exploiting the orthogonality of the sine function:

$$\int_0^L f(x) \sin\left(\frac{l\pi x}{L}\right) dx = \int_0^L \sum_{p=1}^{\infty} a_p \ \sin\left(\frac{p\pi x}{L}\right) \sin\left(\frac{l\pi x}{L}\right) dx$$
$$= \frac{L}{2} a_l.$$

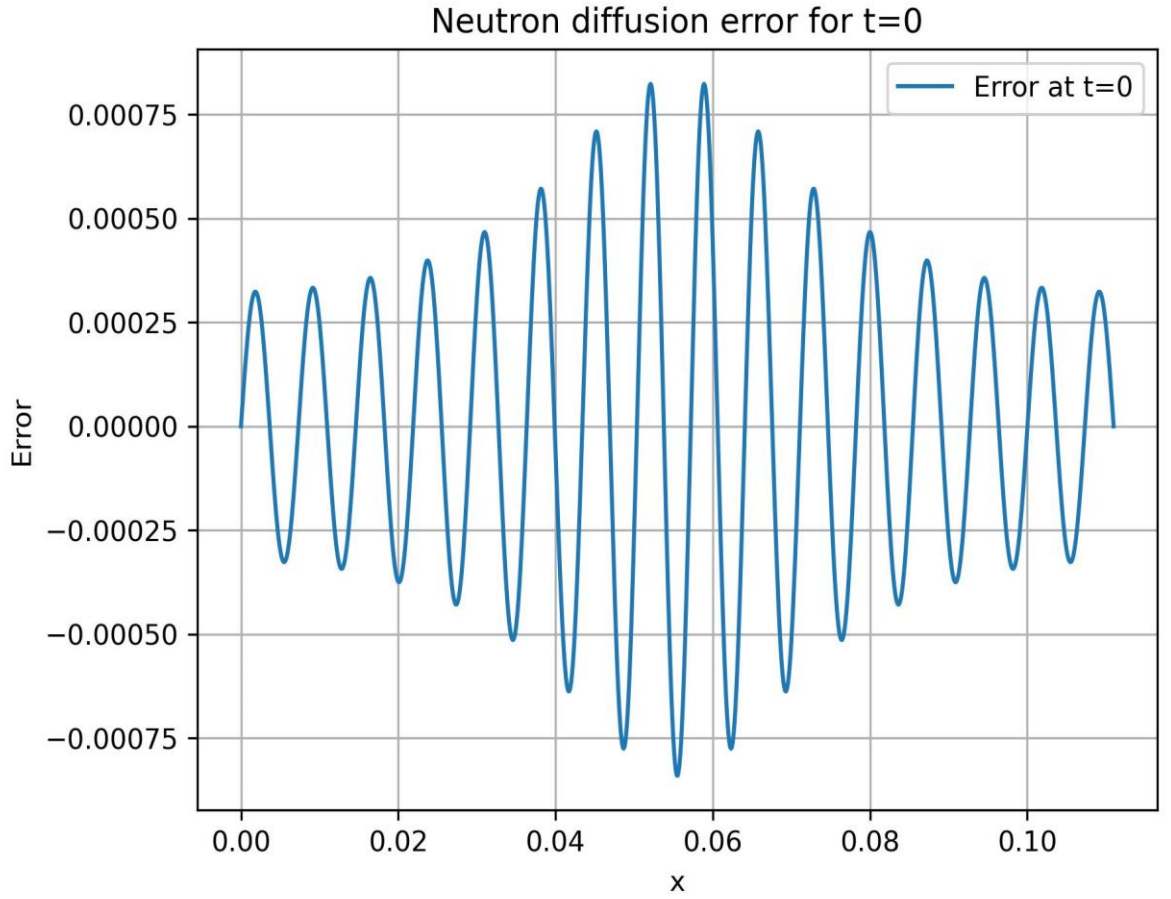
Where $f(x) = n(t = 0, x)$ is the initial condition. In particular we choose:

$$f(x) = A \exp\left(\frac{-\lambda(x - \frac{L}{2})^2}{L^2}\right)$$

With these choices the Python simulation gives the following results (for $N = 30$):



Plot of the neutron denisty for $0 < t < 2 \cdot 10^{-5}$



Plot of the error $n(0, x) - f(x)$. Our choice for $f(x)$ is good since the error is small.

2D Cartesian Coordinates

For the $2D$ case in cartesian coordinates, the diffusion equation is:

$$\frac{\partial n}{\partial t} = \mu \frac{\partial^2 n}{\partial x^2} + \mu \frac{\partial^2 n}{\partial y^2} + \eta n.$$

The variable separation method now leads us to 3 ODEs: the one for $T(t)$ is analogous to the one we find in the $1D$ case. Those for $\mathrm{X}(x)$ and $\mathrm{Y}(y)$ are analogous to the one for $\mathrm{X}(x)$ we find in the $1D$ case. This gives:

$$n(t,x,y) = \sum_{p,q=1}^{\infty} a_{pq}\, e^{\left(\eta - \mu\pi^2\left(\frac{p^2}{L_x^2}+\frac{q^2}{L_y^2}\right)\right)t} \sin\left(\frac{p\pi x}{L_x}\right) \sin\left(\frac{q\pi y}{L_y}\right).$$

Once again the boundary conditions have already been implemented.
Where $L = L_x = L_y$ is the side of our system (we assume it's a square).

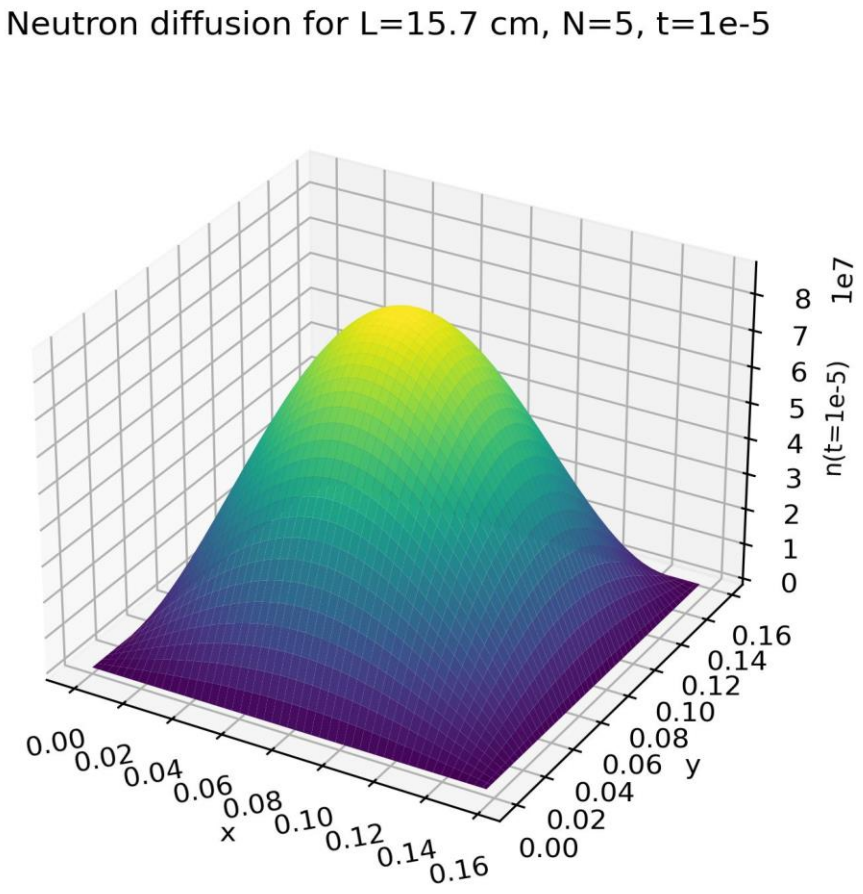
Once again the density will increase unbounded only if the argument of the exponential is positive and for $p = q = 1$ we get the critical length. For U-235 such length is $L_{crit} = 15.62\ cm$. We will perform the simulation with $L = 15.7\ cm$.
We can compute the coefficients a_{pq} as we did for the a_p:

$$a_{pq} = \frac{4}{L^2} \int_0^L \int_0^L f(x,y) \sin\left(\frac{p\pi x}{L}\right) \sin\left(\frac{q\pi y}{L}\right) dxdy.$$

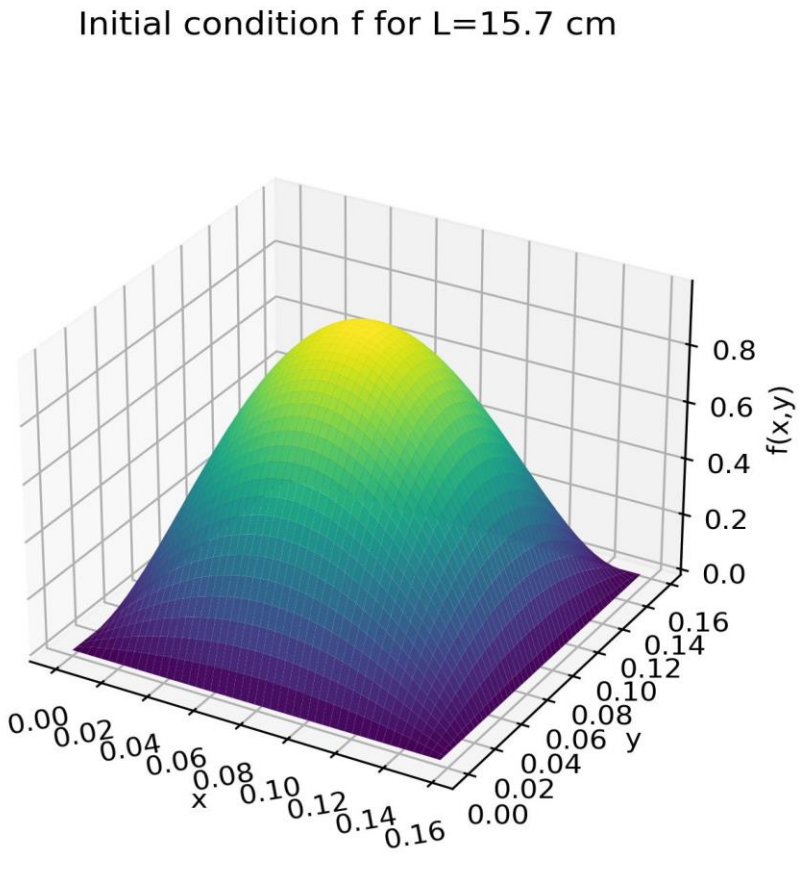
For the initial condition $f(x,y)$ we choose:

$$f(x,y) = \frac{16xy}{L^2}\left(1 - \frac{x}{L}\right)\left(1 - \frac{y}{L}\right).$$

With these choices the Python simulation gives the following results (for $N = 5$):



Plot of the neutron denisty for $t = 1 \cdot 10^{-5}$



Plot of the initial condition. A direct comparison shows that the neutron density blows up very fast.

3D Cartesian Coordinates

For the $3D$ case in cartesian coordinates, the diffusion equation is:

$$\frac{\partial n}{\partial t} = \mu\frac{\partial^2 n}{\partial x^2} + \mu\frac{\partial^2 n}{\partial y^2} + \mu\frac{\partial^2 n}{\partial z^2} + \eta n.$$

Things still play out like before, but now we get 4 ODEs and the resulting neutron density is:

$$n(t,x,y,z) = \sum_{p,q,r=1}^{\infty} a_{pqr}\, e^{\left(\eta - \mu\pi^2\left(\frac{p^2}{L_x^2}+\frac{q^2}{L_y^2}+\frac{r^2}{L_z^2}\right)\right)t} \sin\left(\frac{p\pi x}{L_x}\right)\sin\left(\frac{q\pi y}{L_y}\right)\sin\left(\frac{r\pi y}{L_z}\right).$$

Once again the boundary conditions have already been implemented and once again, setting $p = q = r = 1$, we find $L_{crit} = 19.14\ cm$ (assuming to be in a cube $L_z = L_x = L_y$). Now we can also compute the corresponding volume $V_{crit} = 7007 cm^3$ and mass $M_{crit} = 131\ kg$.

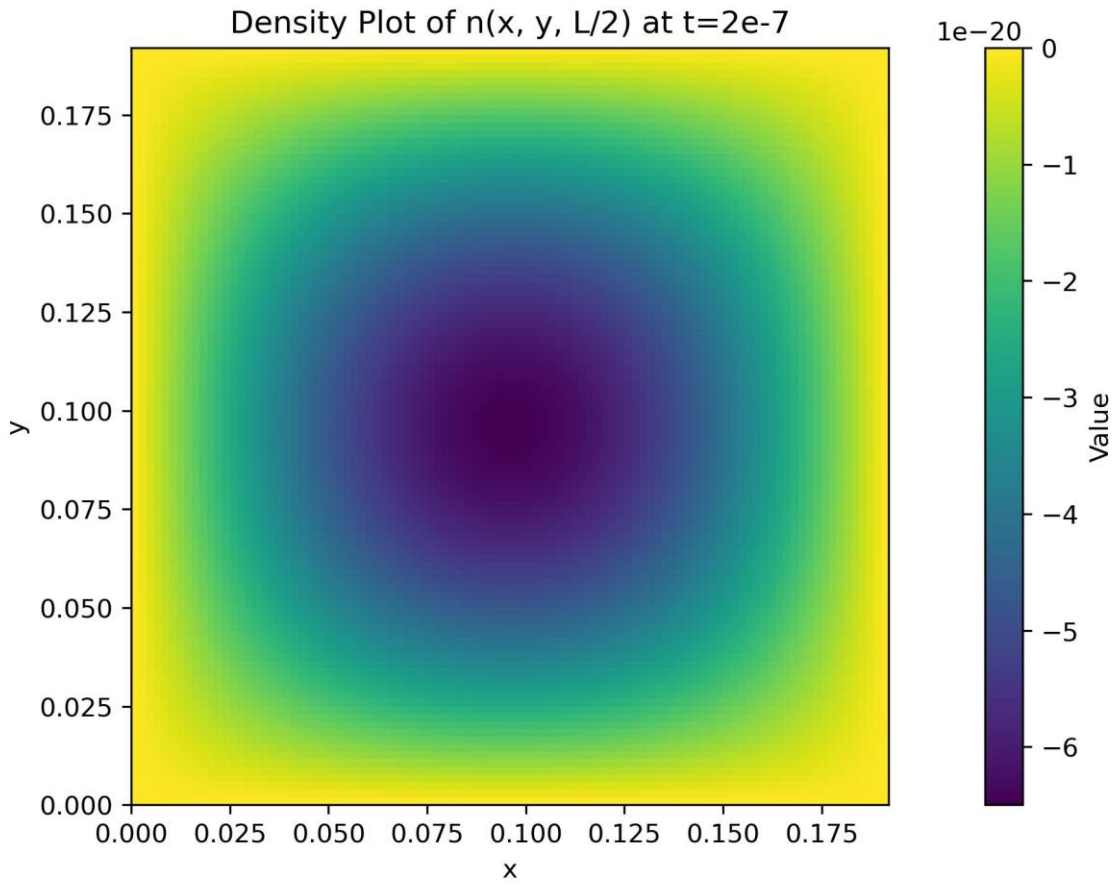
The coeffiecients a_{pqr} are now given by:

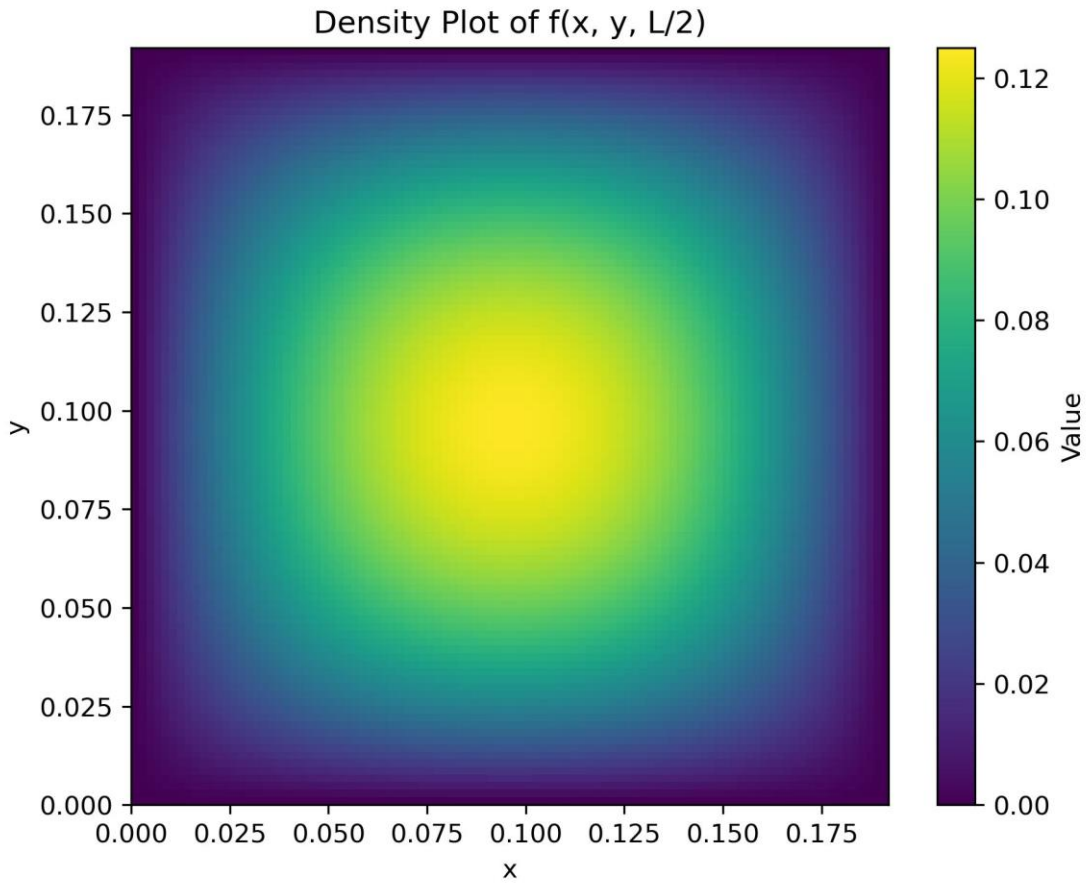
$$a_{pqr} = \frac{8}{L^3} \int_0^L \int_0^L \int_0^L f(x,y,z) \sin\left(\frac{p\pi x}{L}\right) \sin\left(\frac{q\pi y}{L}\right) \sin\left(\frac{r\pi z}{L}\right) dxdydz.$$

For the initial condition $f(x,y,z)$ we choose:

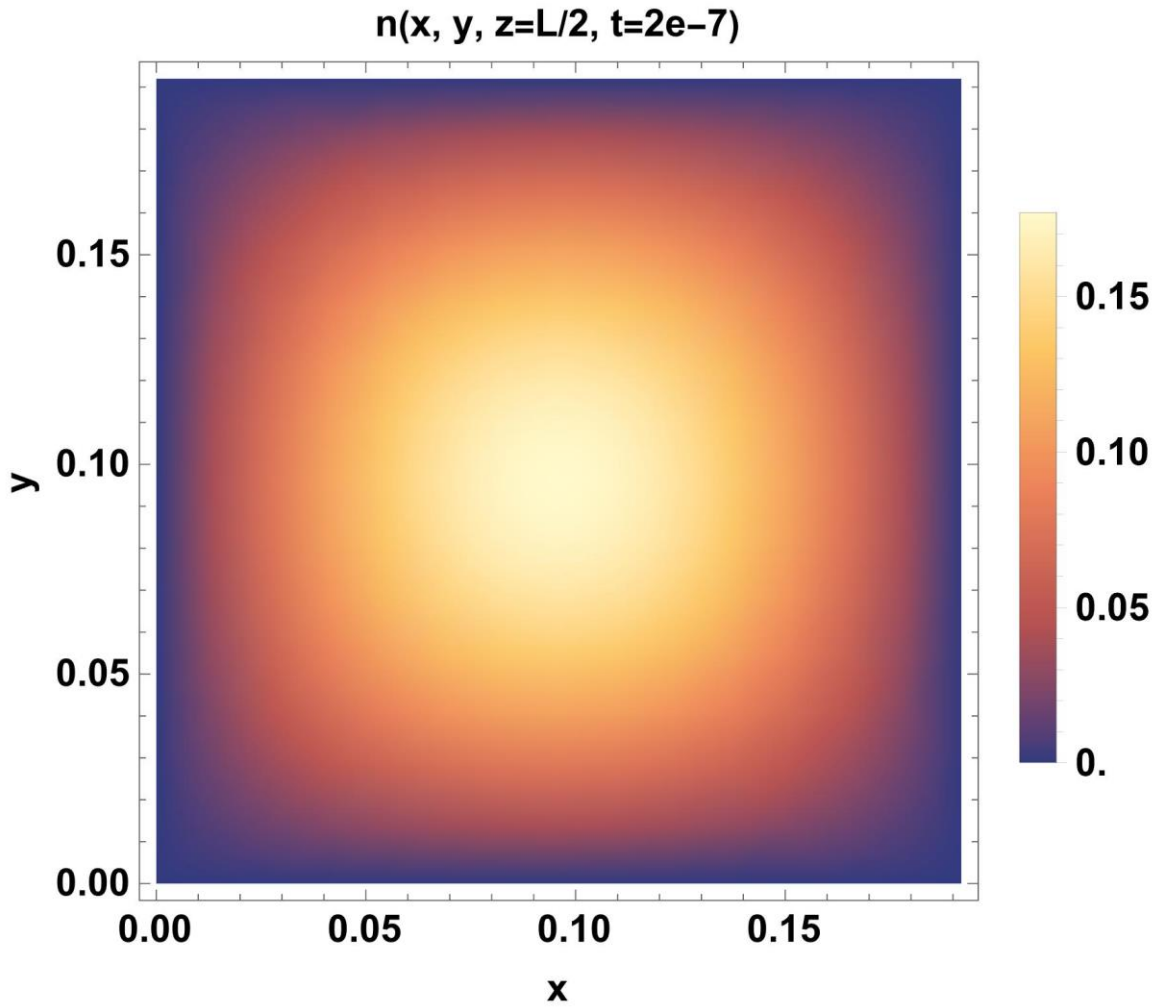
$$f(x,y,z) = \frac{8xyz}{L^3}\left(1-\frac{x}{L}\right)\left(1-\frac{y}{L}\right)\left(1-\frac{z}{L}\right).$$

If we set the side of our cube to be $L = 19.2\ cm$ the Python simulation gives the following results (for $N = 4$):





The pictures above are the density plot of the $z = \frac{L}{2}$ slice of the neutron density and of the initial conidition.
It's clear that the plot of $n(t, x, y, z)$ is not physical since we get a negative density, despite the initial condition is. Therefore there must be something going on when we move from one to the other.
A possibile source of error is represented by the factors a_{pqr}, for this reason I replicated this simulation in Mathematica, which gives a physical result.



This is the plot given by Mathematica for the same geometry.

3D Cylindrical Coordinates

Thanks to radial symmetry, the diffusion equation for $3D$ case in cylindrical coordinates is only a $2D$ problem (for the spatial part):

$$\frac{\partial n}{\partial t} = \mu\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r}\right)n + \mu\frac{\partial^2 n}{\partial z^2} + \eta n.$$

However, if on one hand the problem has less degrees of freedom, on the other hand we have to deal with more complicated equations:

$$\frac{1}{T}\frac{\partial T}{\partial t} - \eta = \frac{\mu}{R}\frac{\partial^2 R}{\partial r^2} + \frac{\mu}{rR}\frac{\partial R}{\partial r} + \frac{\mu}{Z}\frac{\partial^2 Z}{\partial z^2} = -l.$$

Its solutions are given by:

$$R(r) = CJ_0\left(r\sqrt{\frac{l}{\mu} - \left(\frac{p\pi}{L}\right)^2}\right).$$

and

$$T(t) = Ae^{(\eta - l)t}$$

$$Z(z) = B\sin\left(\frac{p\pi z}{L}\right).$$

To find those solutions we implemented only the boundary conditions on z but if we do the same also for r we find:

$$n(t,r,z) = \sum_{q=1}^{\infty} a_{1q} \exp\left(\frac{\eta r_1^2 L^2 - \mu(\alpha_q^2 L^2 + p^2\pi^2 r_1^2)}{r_1^2 L^2} t\right) \sin\left(\frac{\pi z}{L}\right) J_0\left(\frac{\alpha_q}{r_1} r\right).$$

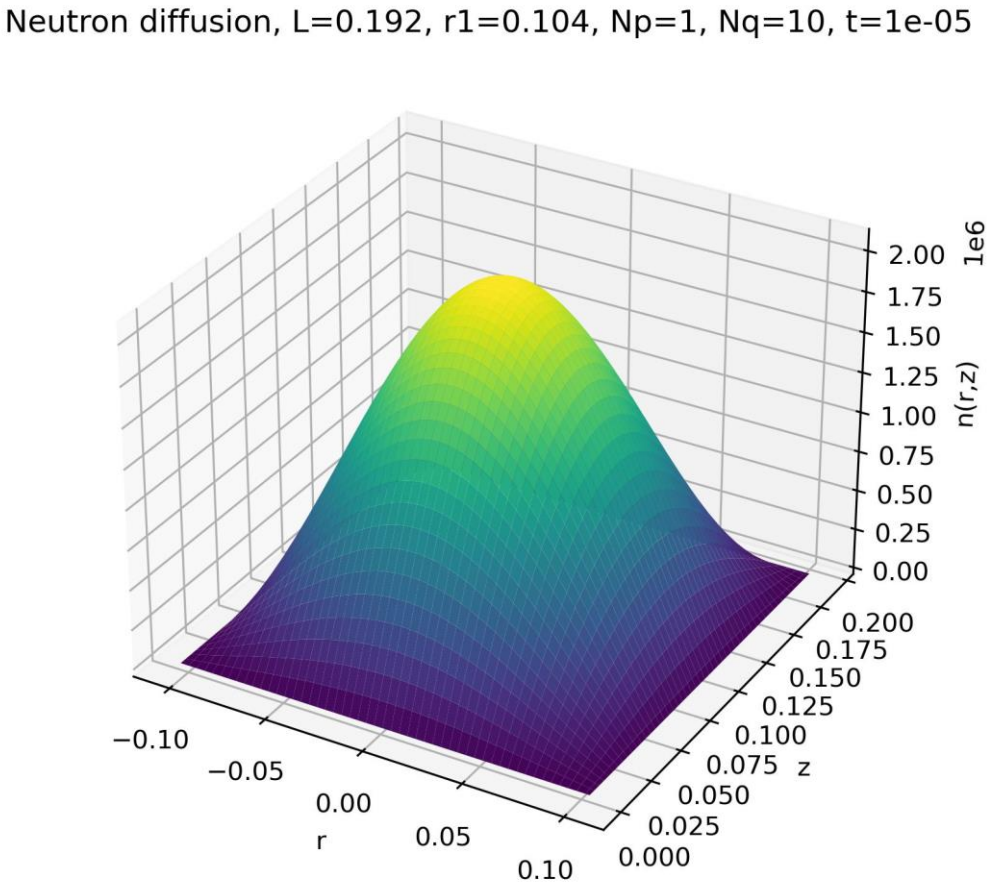
The sum would actually be a double sum over p and q, but an appropriate choice of the initial condition leads to this simplification.
From n we find the critical parameters requiring unboundedness and fixing $q = 1$: $L_{crit} = 19.14\ cm$, $r_{crit} = 10.36\ cm$, $V_{crit} = 6450\ cm^3$ and $M_{crit} = 121\ kg$. We will perform the simulation with $L = 19.2\ cm$ (length of the cylinder) and $r_1 = 10.4\ cm$ (radius of the cylinder).

The coefficients a_{1q} are given by:

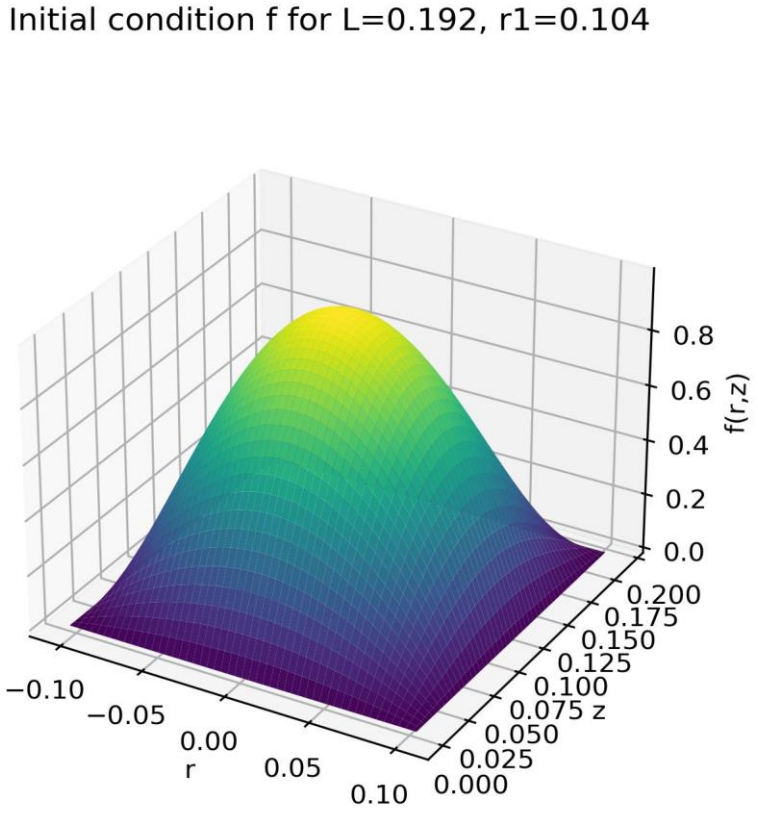
$$a_{1q} = \frac{4}{L r_1^2 J_1^2(\alpha_q)} \int_0^{r_1} \int_0^{L} J_0\left(\frac{\alpha_q}{r_1} r\right) r \left(1 - \left(\frac{r}{r_1}\right)^2\right) \sin^2\left(\frac{\pi z}{L}\right) dr dz$$

They follow from choosing $f(r,z) = \left(1 - \left(\frac{r}{r_1}\right)^2\right) \sin(\frac{\pi z}{L})$.

With these choices the Python simulation gives the following results (for $N_q = 10$):



Plot of the neutron denisty for $t = 1 \cdot 10^{-5}$



Plot of the initial condition. A direct comparison shows that the neutron density blows up very fast.

3D Spherical Coordinates

As for the cylindrical case, also here the radial symmetry simplifies the problem. In fact we have only 1 spatial degree of freedom since the diffusion equation is:

$$\frac{\partial n}{\partial t} = \mu \left(\frac{\partial^2}{\partial r^2} + \frac{2}{r} \frac{\partial}{\partial r} \right) n + \eta n.$$

The solution for $T(t)$ is the usual exponential and the one for $R(r)$ is:

$$R(r) = \frac{B}{r} \sin\left(\frac{p\pi r}{r_1} \right).$$

Hence the neutron density:

$$n(t,r) = \sum_{p=1}^{\infty} \frac{a_p}{r} e^{(\eta - \mu(\frac{p\pi}{r_1})^2)t} \sin\left(\frac{p\pi r}{r_1} \right).$$

The boundary conditions have already been imposed to obtain this form.
r_1 is the radius of our sphere.

From there we can find the critical parameters setting $p = 1$: $r_{crit} = 11.04\ cm$, $V_{crit} = 5649\ cm^3$ and $M_{crit} = 106\ kg$.

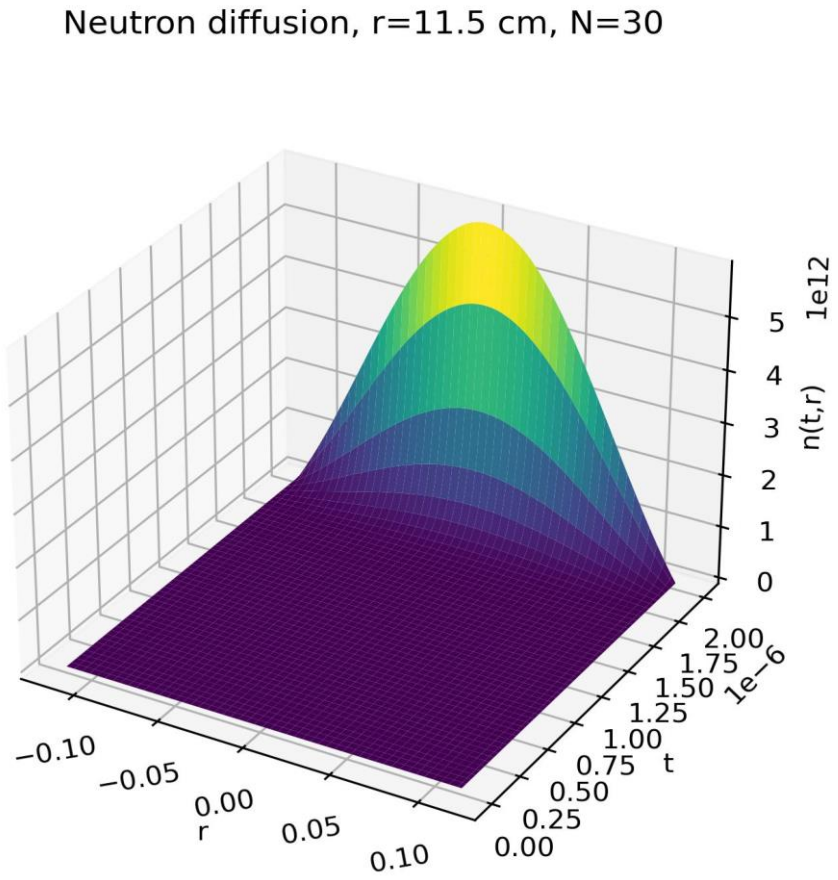
The coefficients a_p are given by:

$$\int_0^{r_1} r f(r) \sin\left(\frac{l\pi r}{r_1}\right) dr = \int_0^{r_1} \sum_{p=1}^{\infty} a_p \sin\left(\frac{p\pi r}{r_1}\right) \sin\left(\frac{l\pi r}{r_1}\right) dr$$
$$= \frac{r_1}{2} a_l$$

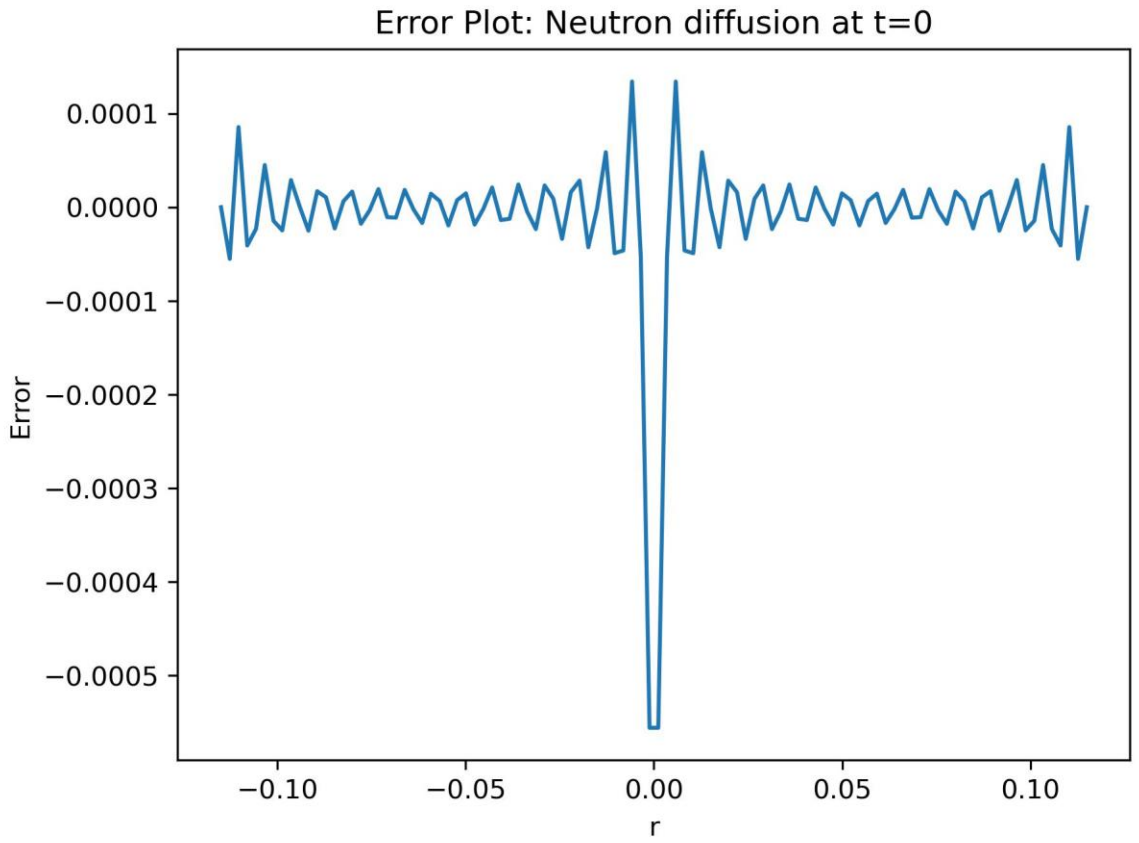
Where for the initial condition $f(r)$ we choose:

$$f(r) = 1 - \left(\frac{r}{r_1}\right)^2$$

If we work with $r_1 = 11.5\ cm$, the Python simulation gives the following results:



Plot of the neutron denisty for $0 < t < 2 \cdot 10^{-6}$



Plot of the error $n(0, r) - f(r)$. Our choice for $f(x)$ is good since the error is small.

3D Spherical Coordinates (Neumann BC)

We want to repeat the analysis we just made but using Neumann boundary conditions instead of Dirichelt ones. Therefore the diffusion equation does not change:

$$\frac{\partial n}{\partial t} = \mu \left(\frac{\partial^2}{\partial r^2} + \frac{2}{r}\frac{\partial}{\partial r} \right) n + \eta n.$$

But instead of asking $n(t, r = r_1) = 0$, we ask that:

$$\left.\frac{dn}{dt}\right|_{r=r_1} = \left. -\frac{3n}{2\lambda_t} \right|_{r=r_1}$$

Where λ_t is the transport free path parameter.
The solutions do not change, however we collect constants differently from before, so we get:

$$T(t) = Ae^{(-\alpha t)},$$

$$R(r) = \frac{B}{r}\sin(kr).$$

Where α is the separation constant and $k = \sqrt{\frac{\eta+\alpha}{\mu}}$.

The implementation the boundary condition yields:

$$-1 + kr\cot(kr) + \frac{3r}{2\lambda_t} = 0.$$

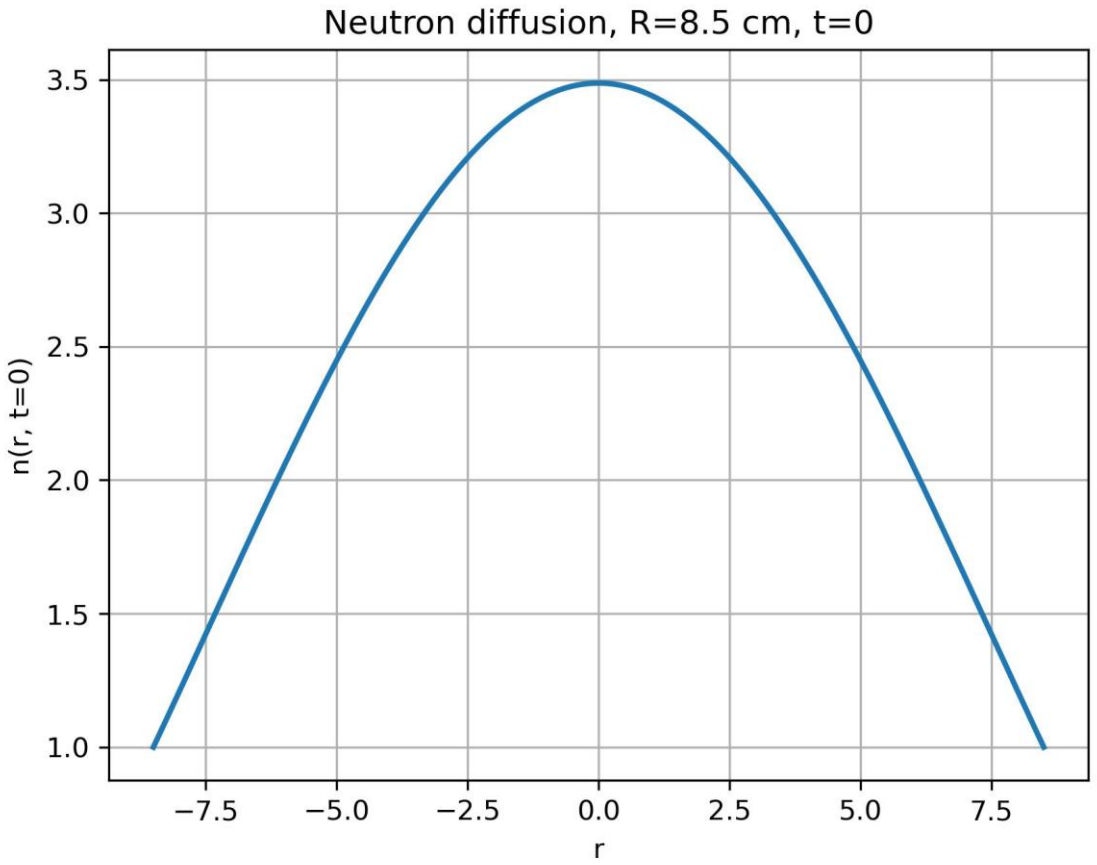
From this equation we can find r_{crit} by setting $\alpha = 0$ and the Python simulation gives: $r_{crit} = 8.363\ cm$, $V_{crit} = 2450\ cm^3$ and $M_{crit} = 45.8\ kg$. The more physical Neumann boundary conditions give a much smaller critical mass.

The neutron denisty is given by:

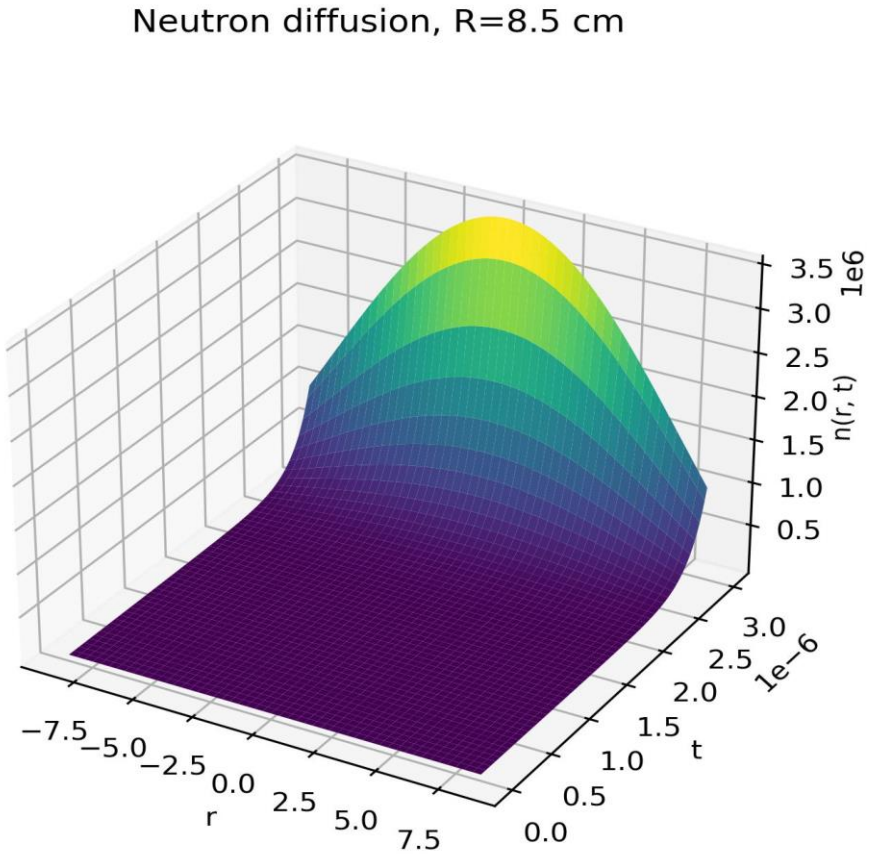
$$n(t,r) = \frac{A}{r} e^{(-\alpha t)} \sin(kr)\,.$$

The constant A is determined via the initial condition, that we choose to be $n(t,R) = 1$, where $R = 8.5\ cm$ is the radius of the sphere we want to work with. Fixing $r = R$ in the equation above we can find $\alpha = -4.06 \cdot 10^{-6}\ s$, $k = 0.2808\ cm^{-1}$ and $A = 12.415\ cm^{-2}$.

With these choices the Python simulation gives the following results:



Plot of the neutron density at $t = 0$ for $r = R = 8.5\,cm$.



Plot of neutron density for $0 < t < 3 \cdot 10^{-6}$ for $r = R = 8.5\,cm$.

Neutron Mean Free Path

We conclude with the computation of the mean free path of a neutron of U-235, that is the average distance travelled by a neutron without any interaction.
Consider to have a block of thickness x which is bombarded with N_0 neutrons. Each slab has a number density on nuclei n and their cross sectional area to the incoming neutrons is σ. Then the number of neutrons that react is:

$$N_{reac} = N_0 \left(1 - e^{-\sigma n x}\right) .$$

Dividing N_{reac} by N_0 we find the proability of reaction, whose derivative is a probability density (of reaction). The mean free path is then given by the expectation value of the position with such probability density:

$$\begin{aligned} \langle x \rangle &= \sigma n \int_0^\infty x e^{-\sigma n x} dx \\ &= \frac{1}{\sigma n}. \end{aligned}$$

Plugging the numerical values for σ and n one finds $\langle x \rangle = 16.89\ cm$.
From the Python simulation we also get the following plots: