

Monte-Carlo modelization of a
thermal neutron detector based on a
 ^6LiF -deposited Silicon semiconductor
using Labview

Autor: Oriol Sans Planell

Director: Dr. Antonio Garijo Real

UOC-URV 2016-17

ABSTRACT

The detection of neutrons has never been an easy task, and the simulation of such detectors isn't either. Current state-of-the-art codes can't appropriately define the behaviour of low-energy particles, and often their results prove imprecise when faced with the experimental confront. At the LEMRAP group, part of the INFN in Rome, we've developed a new kind of low-energy neutron detector, but there's a need to have a precise knowledge of the setup under which such detector must be assembled. This creates the need to program a model able to accurately describe the behaviour of thermal neutrons inside the instrument and hence allow us to optimise its characteristics. Simulations regarding similar setups can be found in the literature, but none of them is sufficiently coincidental with the theory. With the use of Labview I attempt to create a software that will allow us to further study this detector and improve its features in the future.

INDEX

ABSTRACT	1
INTRODUCTION	3
A BRIEF THEORETICAL INTRODUCTION.....	6
MODEL DESCRIPTION.....	8
A COMPUTATIONAL STUDY	15
RESULTS.....	19
CONCLUSIONS AND FURTHER WORK	26
BIBLIOGRAPHY	27

INTRODUCTION

In nuclear physics, one of the hardest tasks is the detection of neutrons, as they're particles without interaction with the electromagnetic fields, and thus cannot be measured directly. Neutrons are present in a wide variety of fields, amongst which medicine (neutron therapy [1][2], secondary particles created from the human body while irradiated by X-Rays [3] or from the use of LINAC [4]), nuclear energy production [5][6] or industry [7] with the neutron radiography. To measure such particles, the system of choice is making use of a nuclear reaction, which is only done under certain well-known conditions, which will be explained further on in detail – In the model description -. To summarize it, a low energy neutron impacting the nucleus creates two particles which are easy to measure, despite the process being at low probability. When creating a detector, three steps must be followed: theoretical, model and application. First, theoretically the instrument must be capable of working, basing its work on a solid physical principle. Once the basis is created, computational models ensure its functionality and optimal range of work, as well as its properties. Finally, with all the information the prototype can be created and put into test. In the case of this work, a detector has already been tested several times, but it's still uncertain which are the optimal properties under which it can operate. Typically, this study would be performed with MCNP, which is a code specifically designed for Monte-Carlo simulation in Nuclear Physics [8][9], but the program still has some limitations, among which the low energies. The statistical models that the software uses to compute the behaviour of low-energy particles are not well-defined, and thus the results are not trustworthy. This generates the need of creating a simulation from scratch to overcome this limitation, which brings sense to the making of this work.

The main objective of this study is to simulate the energy spectrum of a thermal neutron flux measured with a Silicon detector with an area of 1cm^2 having a deposited layer of ${}^6\text{LiF}$ on it. Our group focuses on the development of radiation detectors for medical and research purposes. On this project, the main scope will be a study regarding the optimal ${}^6\text{LiF}$ layer height.

The detector to simulate is a silicon-based photodetector with an effective area of 1cm^2 from the Hamamatsu brand. Normally this kind of instruments are unable to detect neutrons at any energy, therefore a compound must be placed on top of it that, in a way, makes those particles visible to the detector. The technique used is a vapor deposition of Lithium Fluoride. When a low energy (thermal) neutron interacts with a Lithium nucleus, a reaction takes place that emits two products, one alpha particle (Helium nucleus) and a tritium (hydrogen-3) in opposite direction one another. Typically, the particle that doesn't move towards the photodetector shall be considered lost, as it won't produce any measurable signal, whilst the other, upon interaction with the Silicon environment – which is modeled as pure Silicon, despite having a measured presence of impurity rounding a particle per million – generates ionization on its path while releasing kinetic energy.

The silicon detector is a diode, meaning it has two electronic poles, positive and negative, which generate an electric field that can be enlarged externally by inducing a bigger potential between those poles (by connecting it to a power supply, for example). Making the electric field stronger generates a widening on what comes called “depletion zone”. The depletion zone is the effective volume on the detector upon which the signal is created. Such space varies with the electric field of the diode, reaching a maximum, in the laboratory case upon which this simulation is based, while applying a voltage of 60V in inverse current. The total height of the depletion zone while the current is applied is rounding the 300 micrometres, which is the value that will be taken as reference on the simulation.

The code used to program the software is NI Labview 2015 [10][11][12]. Labview is a systems engineering software for applications that require test, measurement and control with rapid access to hardware and data insights. It also has the capability to generate high-end simulations, and it can be used with a wide variety of fronts, from the graphic interpretation of an image to the control of an Arduino, passing through an extensive range of mathematical calculations. At the LEMRAP group [13] this code has been widely used for a variety of system, such as: data acquisition software for impulse based detectors, which analyses every electrical pulse sent by a digitalizer and creates an energy spectrum out of the compound; current-mode acquisition software, which calculates the mean signal from a continuous electric pulse coming from a detector; the full program that controls an X-Ray tube, with multifunction to also acquire data from experiments connected to it. It is also used to elaborate big amounts of data, easily creating routines that would be more difficult to program in another language – such as C, java or python -.

Following the lines of the MSc, it has also been done a computational study of the final software, to find out what the response is, from a time point of view, while increasing the size of the execution, and what is the behaviour of the program at a big regime. Such information can help in the future to further improve the simulation, as it won't only be necessary to increase the physical significance of the model, but also render it more efficient computationally speaking, as a more complete algorithm will also take a bigger toll on the performance, and thus will need some fine tuning to optimize the computer-time required for the big execution cases.

Such a simulation is not menial in terms of size, and it is also complete in terms of subject, since it requires a deep understanding of both the physics involved and the knowledge regarding this kind of stochastic computational routines. For such, many assignments from the MSc have proven useful: For the calculations, the experience obtained at the Differential Equations course has been valuable, as it worked the mathematical approach to real-life scenarios; regarding the design of the algorithm, the Simulation course on the first semester has given the insight and resources to create a routine to solve the problem, as well as provide the tools to render it practical; From the course in High Performance Computing has come all the knowledge on how to optimize the algorithm, what variables and issues to look at and how to face them. In short, the

overall of the Master has proven, each course on a different degree, vastly precious to the making of this simulation.

During the programming process of the simulation there have been a wide diversity of difficulties to face, from the making of the libraries, to the normalization of all the physical units involved in the model, passing by the incoherencies that the first results showed in comparison with what was expected.

Essentially the program needed the full libraries of the cross-sections of the neutrons in ${}^6\text{LiF}$ at the various energies involved – considering that the most probable energy is 0.025eV but it follows a Maxwellian distribution, it should be considered up to 1eV -, therefore the algorithm was in need of a file with the discrete values from the minimum to the maximum expected, as well as a routine to infer the exact cross-section value as a function of the given energy. A similar procedure was needed for the range of the particles in the material.

In this work the reader will find a brief theoretical introduction, which will grant the necessary knowledge to understand the model and give a little insight on the physics involved in the process for those who are outsiders to the field. The model description is a step-by-step explanation of the algorithm, with graphics that detail the origin of the data that is used to validate the outcome of the program. Before the results, where the most relevant spectra are shown there is a computational study made which analyses the routine from a computational point of view, exploring the limitations in terms of performance, and theorising a way to improve it in the future. Finally, the conclusions and future work reflects on the work that is going to be done in the following months to improve this project even further.

A BRIEF THEORETICAL INTRODUCTION

Neutrons are subatomic particles with a neutral electric charge and a mass of 939MeV. They are baryons constituted by two down quarks and one up quark. They are considered fermions, as they have a half spin, and are susceptible to all four fundamental forces.

As neutrons don't interact with the electric field, the method to detect them must be indirect. The process upon it is based the detection in the case studied is the nuclear reaction that takes place when the neutron is absorbed by a nucleus of Lithium, which releases a couple of particles, an alpha and a tritium. To interact, the neutron must directly impact the nuclei, and such is a stochastic process, which is based on the cross section, expressed as the probability of the neutron to collide.

The simulation uses the Monte-Carlo method in order to emulate the random behaviour of the particles. Monte-Carlo experiments are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. This method was firstly ideated by Fermi back in the 1930s, although officially created, developed and properly theorized by John Mauchly and Presper Eckert in the University of Pennsylvania in 1945¹, with its first applications being the calculations of the distance travelled by neutrons inside a material.

Essentially this kind of algorithms follow three routes: optimization, numerical integration and generating draws from probability distributions. This method is widely used in physics-related problems and mathematics, as well as economics or medicine. Essentially, Monte-Carlo methods are used to solve problems with many degrees of freedom that are based on probability distributions. As much as the algorithm may vary from one application to another, the procedure follows a similar pattern: define the domain of possible inputs, randomly generate the inputs, perform deterministic procedures on those inputs and finally aggregate the results. One classic example of the Monte-Carlo procedure can be the calculation of the area inside a circle. With Labview a simple problem like this can be solved the following way: first of the routine generates the incircle of a square which has a known side. Further on, the program starts randomly putting points inside this square. It will look like this:

¹ Information extracted from the book "The beginning of Monte-Carlo method" by N. Metropolis [14].

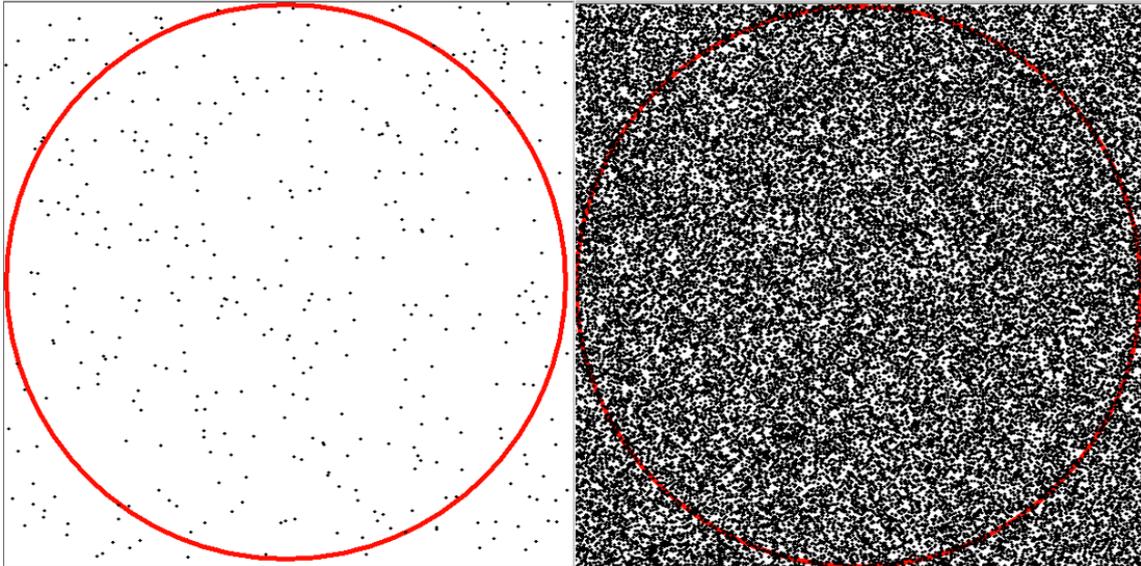


Figure 1: monte-Carlo method to calculate PI through the proportion of target-miss points in a circle

On the first image the algorithm has only generated 300 points, from those, the proportion of which that are inside the circle enable the calculation of PI, although the result is 2.974, which is pretty imprecise. Letting the program run some more, up to 35.000 iterations, the output (PI) is 3.1443, which is way more accurate. If the routine was let to calculate for a large amount of time, the result would be more and more precise. This creates the need to define a point upon which it makes no sense to continue with the simulation, as no additional relevant information will be provided. In this case, if the person calculating PI was in need of – for example – 3 significant digits, around 100k iterations would be more than enough. In the case regarding this Master’s thesis, the uncertainty becomes sufficiently small after 5 million iterations (particles), that’s why this amount has been chosen for the main calculations. Longer simulations could have been done, but would be quite demanding from a temporal point of view, and would not yield significant variations on the results.

MODEL DESCRIPTION

The detector is a 1 square centimetre photodiode with a deposit of ${}^6\text{LiF}$ on it. In order to model it, some assumptions have been made:

- The detector and the layer on it are geometrically regular.
- The materials composing both the layer and the detector are perfectly pure.
- The neutrons that cross to any point outside the deposited layer are considered lost.
- The space outside the detector and layer is absolute void with no effects on any particle that enters it.
- Particles not directed downwards are considered lost.

Considering the approach, the detector is modelled as a regular prism of pure silicon, with another regular prism on top made of ${}^6\text{LiF}$. Both layers are perfectly in contact, and the whole system is centred at the origin. A 2D cut on the plane $y = 0$ would make the detector look like this:



Figure 2: Detector schematic

With the red dot in the middle being the centre of the geometric space $(0,0,0)$. The upper region represents the ${}^6\text{LiF}$ deposit, while the brown region corresponds to the silicon detector.

The neutrons are all generated on the plane $z = h_{6\text{LiF}}$, being that height defined as the thickness of the deposition layer on the detector, with a speed distribution depending on the energy which follows a Maxwellian distribution:

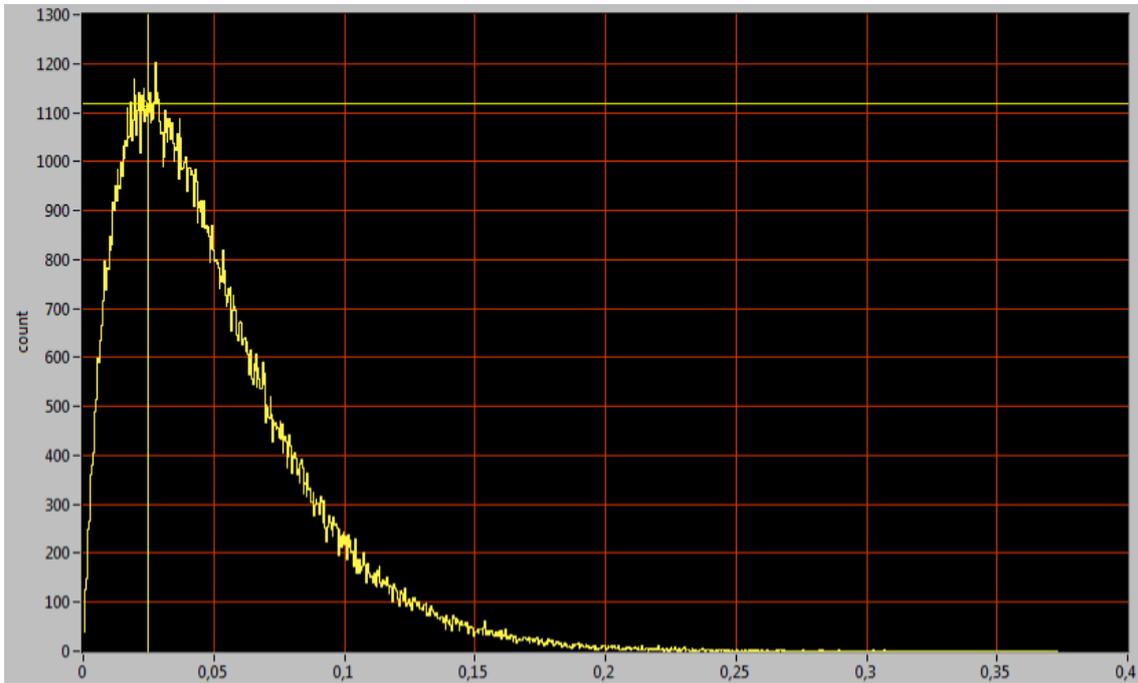


Figure 3: Energy distribution for 25meV neutrons

On Figure 2 it can be recognized the energy distribution followed by the thermal neutrons. This graph has been generated by the simulation, showing the initial energy of 200k neutrons distributed over 1000 energy bins. The most probable energy value is 0.025 eV, which is the energy at which the neutrons have a thermal behaviour, meaning that their kinetic energy is too low to bounce elastically when they collide with another structure (typically a nucleus) and thus they get absorbed on their first impact. Higher energy neutrons (above 0.2eV) would drop energy on the environment via elastic bouncing at least once before getting absorbed. That fact would imply a change on the shielding used to contain the source, or a variation on the expected response of the detector measuring them.

The thermal neutrons generated are expected to be absorbed on the 6LiF layer, following the reaction:



That means, a neutron reacting with a Lithium-6 nucleus that leaves, as a product, an alpha particle plus a tritium, along with 4.78MeV of kinetic energy that is shared between both products.

Following the work from P. Meyer et al. in 1995 [1] and L. Makovicka in 1987 [2], the software calculates the distance between the starting point and the first interaction point (absorption in our case) at each particle as:

$$Dist = \frac{-1}{\Sigma_{TOT}} \ln(1 - rnd)$$

Where Σ_{TOT} is the total macroscopic cross-section of the neutrons, and rnd is a random number between 0 and 1.

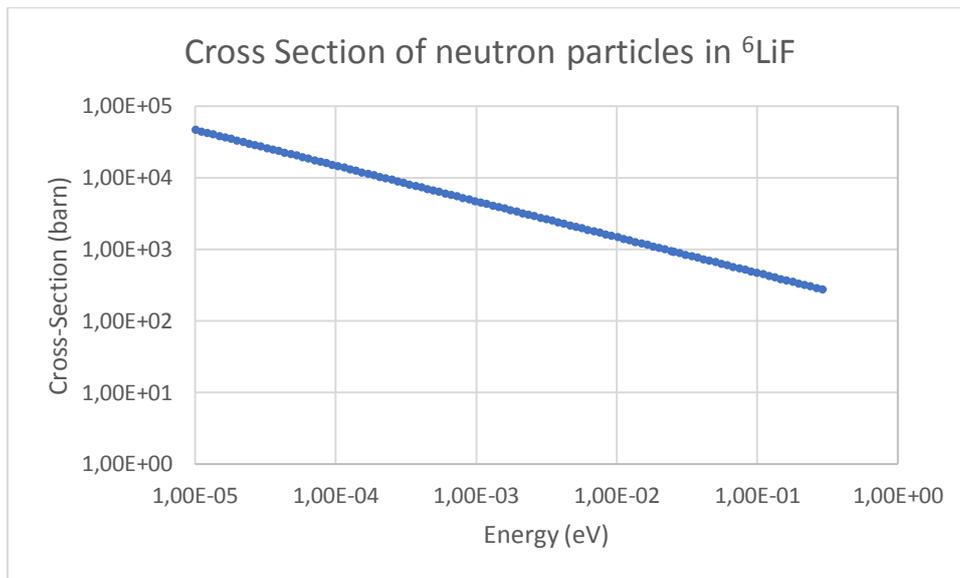


Figure 4: Cross Section of neutrons in 6LiF at low energies.

Once determined the interaction point, the two resulting particles are created, with an energy of 2.73 MeV in the case of the tritium and 2.05 MeV for the alpha, heading towards opposed directions randomly chosen. The algorithm proceeds to calculate the interaction point between the line projected by the course of both particles and the box that is the 6LiF layer. If one of the interaction points is the lower face, coinciding with the top limit of the Silicon layer, the range of the given particle is calculated using data taken from the SRIM code database, and interpolated using the linear method.

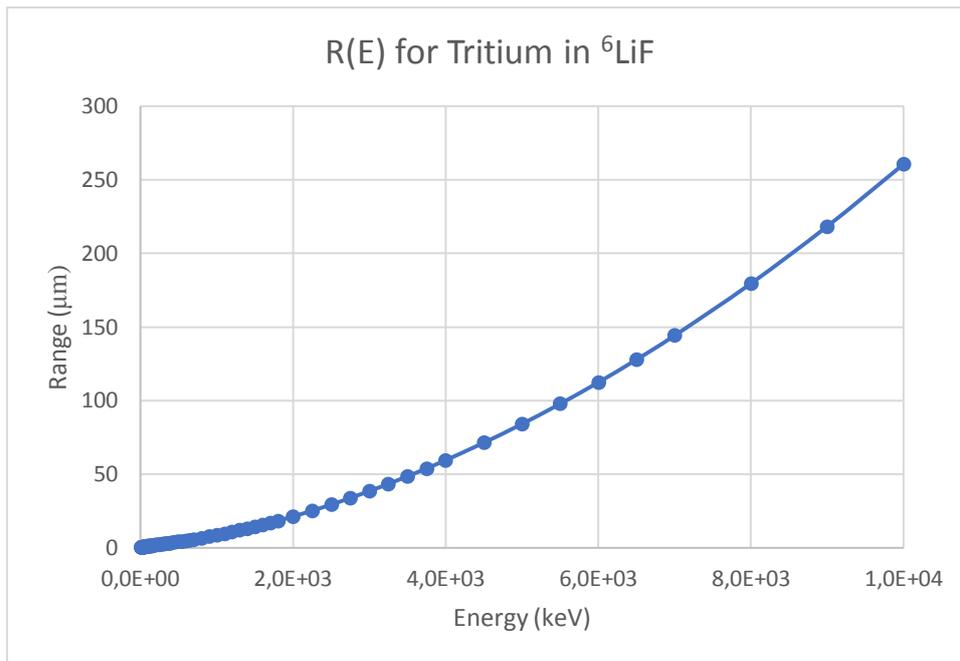


Figure 5: Range of tritium particles in ${}^6\text{LiF}$ as a function of energy

If the range of the particle is smaller than the distance between the generation point and the interaction with the top Si layer it means that it gets stopped before crossing, and therefore it won't leave any energy for the detector to measure, and thus the program ignores this particle and moves on to the next one.

Considering the case where the particle successfully arrives to the Silicon detector, it will have lost some energy on the way, which is calculated by the difference between the range and the distance travelled. On image 1 it is shown an example of a tritium particle in a ${}^6\text{LiF}$ environment, with an energy of approximately $7\text{E}+3\text{keV}$, which means it has a range of 150 micrometres, travelling a distance of $50\mu\text{m}$. Being the range of the particle $100\mu\text{m}$ at the end, it will mean that the particle has lost the difference in energy to the surrounding environment, which would be, in this case, around $1.5\text{E}+3\text{keV}$. Hadn't it been the ${}^6\text{LiF}$, but the Si, the particle would have ionized the surrounding, and the difference in voltage between the top and bottom layer of the detector would have collected the generated charge, producing a signal that would later be elaborated as a pulse for the spectrum.

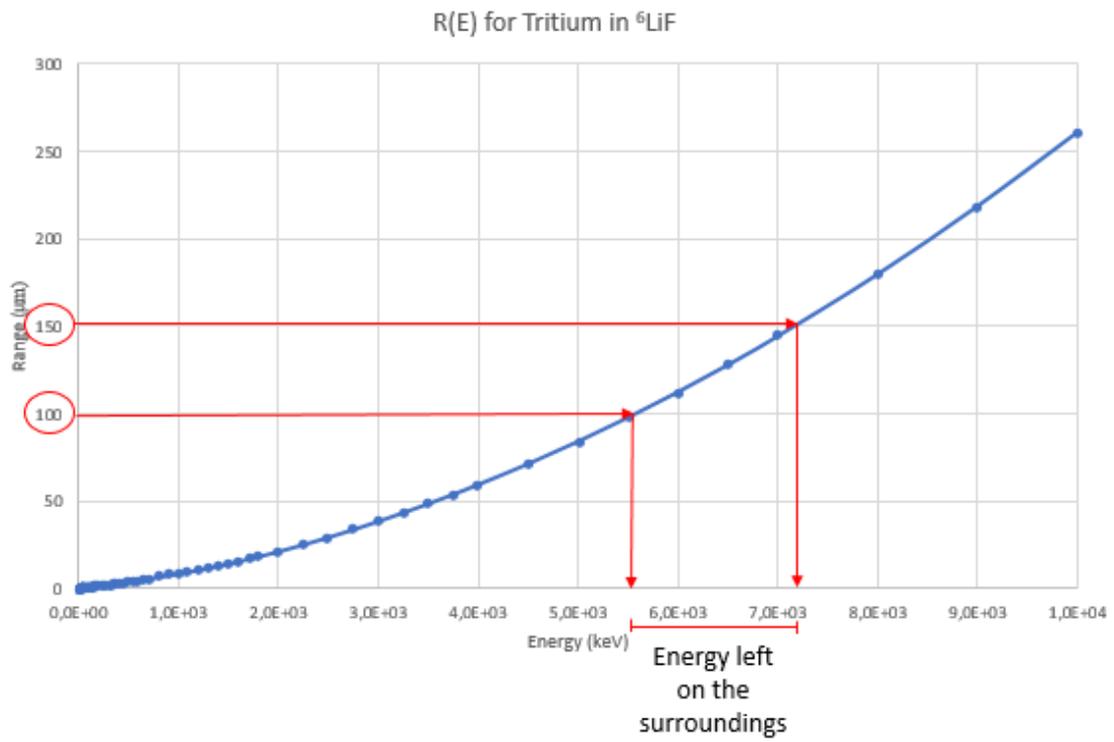


Figure 6: Calculation of the energy deposited by a particle travelling a certain distance

Overall, the program executes the following diagram:

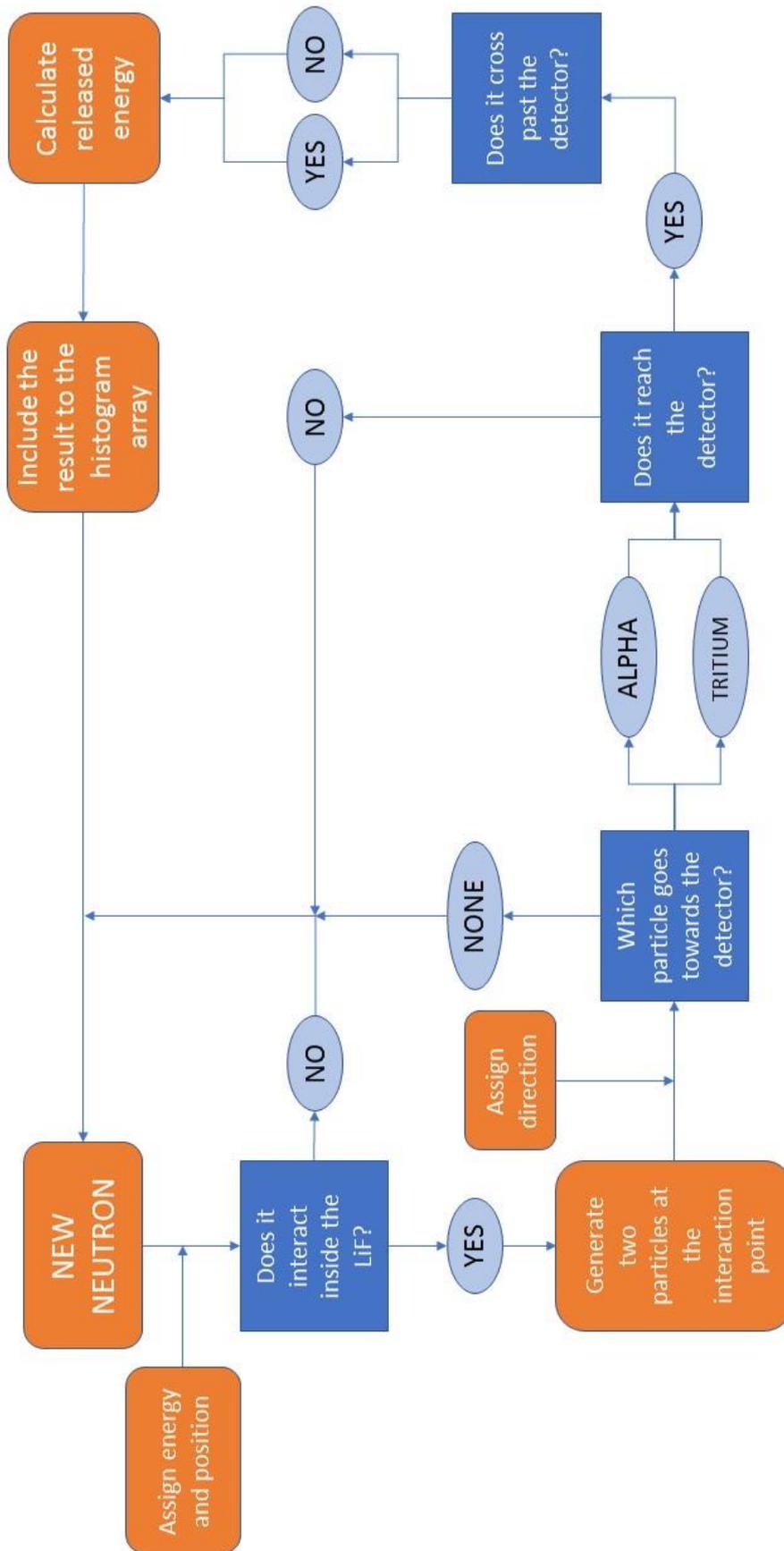


Figure 7: Flux diagram of the program

As can be read on the flux diagram, the routine starts generating a neutron at a random position of the plane $y = h_{6\text{LiF}}$ directed downwards, following the energy distribution seen on Figure 2. Most of the neutron generated pass through the deposit without interacting with the material, thus not generating a signal. The probability of interaction depends on the cross-section of the neutron in the material, as shown in Figure 3. Those cases in which the neutron impacts with a nucleus of Lithium provoke the generation of both a tritium and an alpha particle, both heading in directly opposed ways – the direction is chosen at random in the 3D space -. The program then choses the one moving downwards, and ignores the other. There is the hardly probably case in which the direction of the particles is completely parallel to the plane of the detector, in which case none of the particles would ever interact with the Silicon and therefore no signal would be generated. Once the routine identifies which particle is heading towards the detector, it calculates the range (Figure 4) to see whether it reaches the Silicon or not, In case of the particle being too far to get to the active layer (a graphic explanation of this is shown further on, with Figure 18). If the particle reaches the Silicon it will be considered for the efficiency, regardless of the amount of energy released (signal). To calculate the height of the signal, which is equivalent to the energy deposited by the particle in the environment, the algorithm considers the distance travelled in both 6LiF and Si and following the process seen in Figure 5 deduces the energy signal, which is then included at the array from which the histogram (spectrum) will be elaborated.

A COMPUTATIONAL STUDY

First off, one example of the actual code:

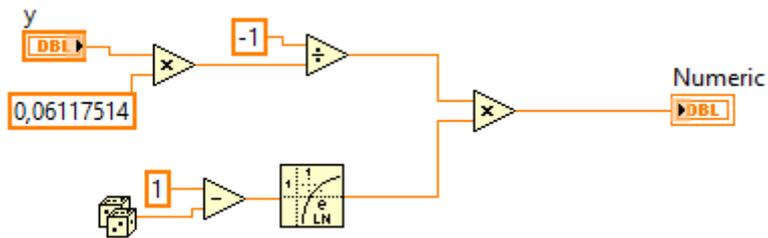


Figure 8: Code example

That's a part of a subroutine, this structure precisely calculates the distance a neutron travels before colliding with a Lithium nucleus, using the formula:

$$Dist = \frac{-1}{\Sigma_{TOT}} \ln(1 - rnd)$$

Labview automatically parallelizes all the possible instances of the program, meaning that if two operations can be done simultaneously, the software automatically uses a second thread to do it in order to lessen the total calculation time. On the previously shown example, the parallelization goes as follows:

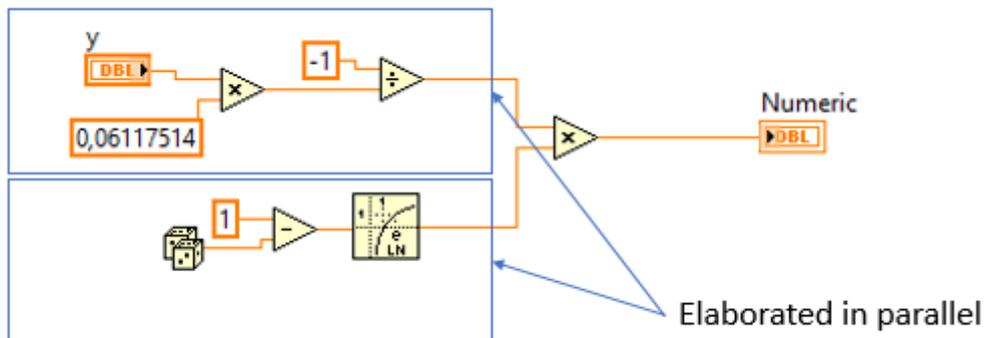


Figure 9: Parallelization Code Example

Depending on the design of the code, the multithread will be done following a loop parallelization (having a loop that must be executed various times, it can be set for a number of “workers” to execute it simultaneously, reducing the time) or feedback nodes (as would be the case of multi-dependency matrices, such as the solution for the Laplace equation for heat). More information on the matter can be found at [3].

The first execution, done to be made as reference, has been with 1E6 particles. From this point, several other operations have been made, in order: 2e6, 3E6, 4E6, 5E6 and 1E7. Each and every of them has had the time measured precisely. A few smaller calculations have also been done in order to include the small regime to the graphics.

On the curve there’s the theoretical evolution to be expected for the time taken for the simulation, taking into account that the program has been already parallelized, the increase on the base number of particles should linearly increment the time required to finish the execution, since every particle is evaluated independently.

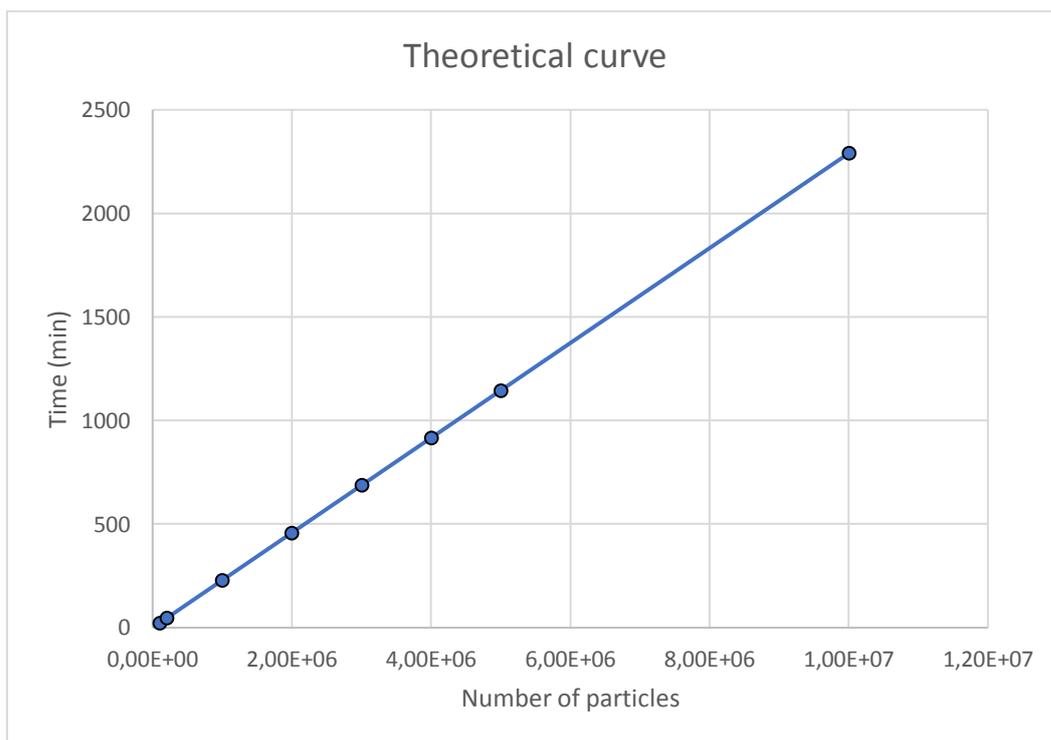


Figure 10: Theoretical time curve

The expected time for a large simulation (about 10 times the original value) should be around a day and a half, considering the duration of the first execution – about 4 hours

-.

Contrarily to what was expected, the amount of time taken for the executions after 2E6 vastly differs from what was expected, following Figure 4:

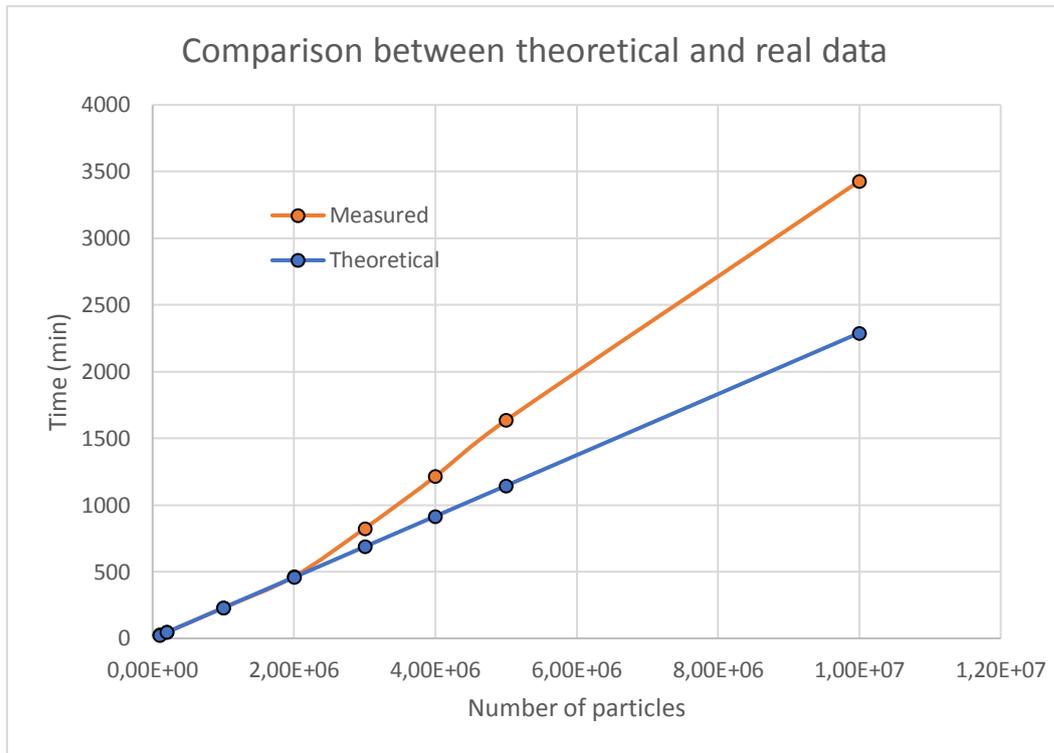


Figure 11: Comparison between theoretical and real time

As can be appreciated, the time required to compute the experiment for a large number of particles is greater than expected. Once above the 2-million threshold, the curve changes to increase linearly faster than the theoretical hypothesis. The biggest difference, clearly, being about 50% on the 10-million point.

The code essentially makes the same operation for each particle, indistinctively from the number of particles there are, and stores the result in an array which is later on transformed into a histogram². Since the time required to analyse a single particle can be considered always as a constant, the variable causing trouble on the system could be traced to the memory. As the experiment grows, the array stored in the memory with the results becomes greater, and is passed from one iteration to the next, therefore provoking a slow-down once this array is too big.

In order to solve this issue, two things could be done: either modify the software in order to avoid this iterative passing of the results, or improve the hardware by increasing the

² This histogram is the energy spectrum.

memory, allowing for a bigger array to be transmitted without saturating the capacity of the computer.

A viable modification to the software that would allow for the program to have a linear evolution instead of exponential would be to make the algorithm copy the result of each iteration in a file initially opened. Such file would be read at the end of the experiment to create the histogram. This action would clearly slow the execution with small experiments, but as the time needed for each iteration would remain constant, as it wouldn't depend on the dimension of the file, with high-numbered executions the evolution would be directly proportional to the number of particles, and thus it would end up saving time. A theoretical approximation of such behaviour can be seen on the following graph:

RESULTS

Once the program was finished, it was executed repeatedly, to get the neutron spectra varying the ${}^6\text{LiF}$ height, in order to find the layer volume that lead to the maximum efficiency of the detector. A total of 30 simulations have been done, from 2 to 60 micrometres, considering the depletion zone of 300 micrometres, the detector having an effective area of $1\text{cm} \times 1\text{cm}$ and a total number of particles generated equal to $5\text{E}6$. All the output spectra are shown in Appendix [X], but some of the relevant ones will be described here.

The first spectrum to describe is the one obtained while the ${}^6\text{LiF}$ layer is 4 micrometres tall.

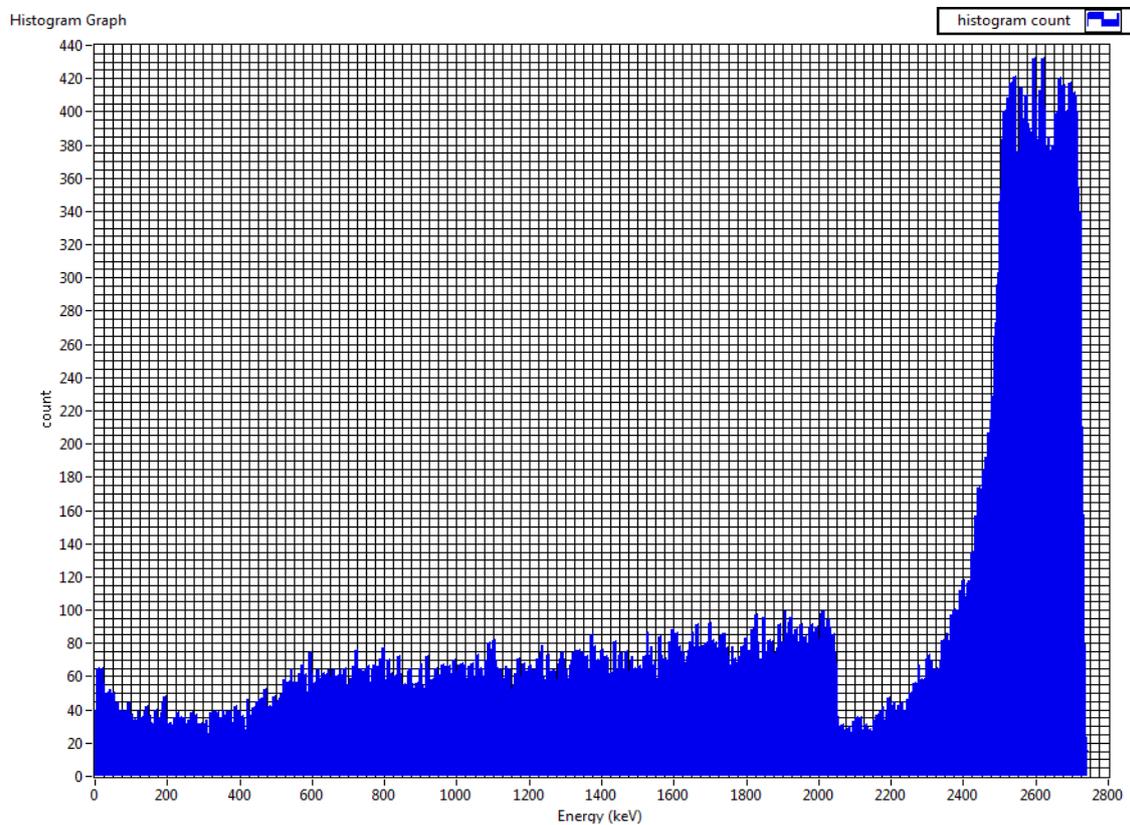


Figure 12: 4 μm ${}^6\text{LiF}$

From this spectrum two things may be appreciated: first of, there are two clear points, one right after 2000 keV, and another one slightly after 2750 keV. Those two points are the endpoints of the alpha and tritium released by the reaction. The difference between the height of both regions is justified as most of the tritons generated don't lose much energy on the ${}^6\text{LiF}$ layer before getting to the active depletion volume. On the other hand, alpha particles on ${}^6\text{LiF}$ have a range of $6\mu\text{m}$, therefore it is expected the energy distribution they get when arriving to the silicon volume to be fairly uniform, as the

particles generated on the very start will have lost most of their energy, while the ones at the last points will have all of their kinetic energy intact upon the arrival.

It is expected to see the height difference between both peaks reduced as the 6LiF layer becomes taller, such effect is due to various reasons: a higher 6LiF layer means that the tritons generated on the first micrometres will have to pass a longer distance, thus getting to the detector with less energy, leaving a smaller signal.

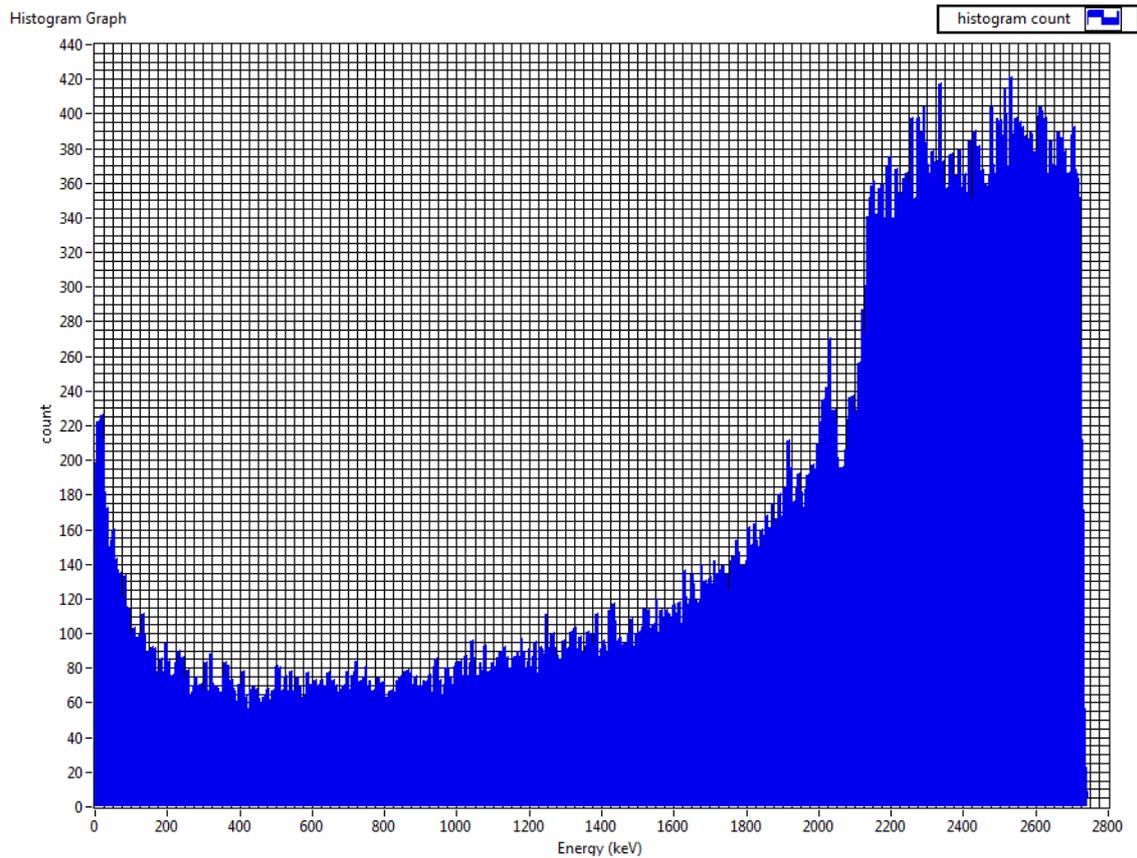


Figure 13: 10 um 6LiF

It can be appreciated how on the case of Figure 2, having a 6LiF layer of 10 μm , the alpha region is no longer visible. This is due to the fact that the triton peak has widened, as the particles generated now have a higher chance to release energy on the 6LiF environment before getting to the detector. This effect of the tritium peak has "eaten" the alpha region, covering it. It should reappear further on, as the tritium still should diminish on.

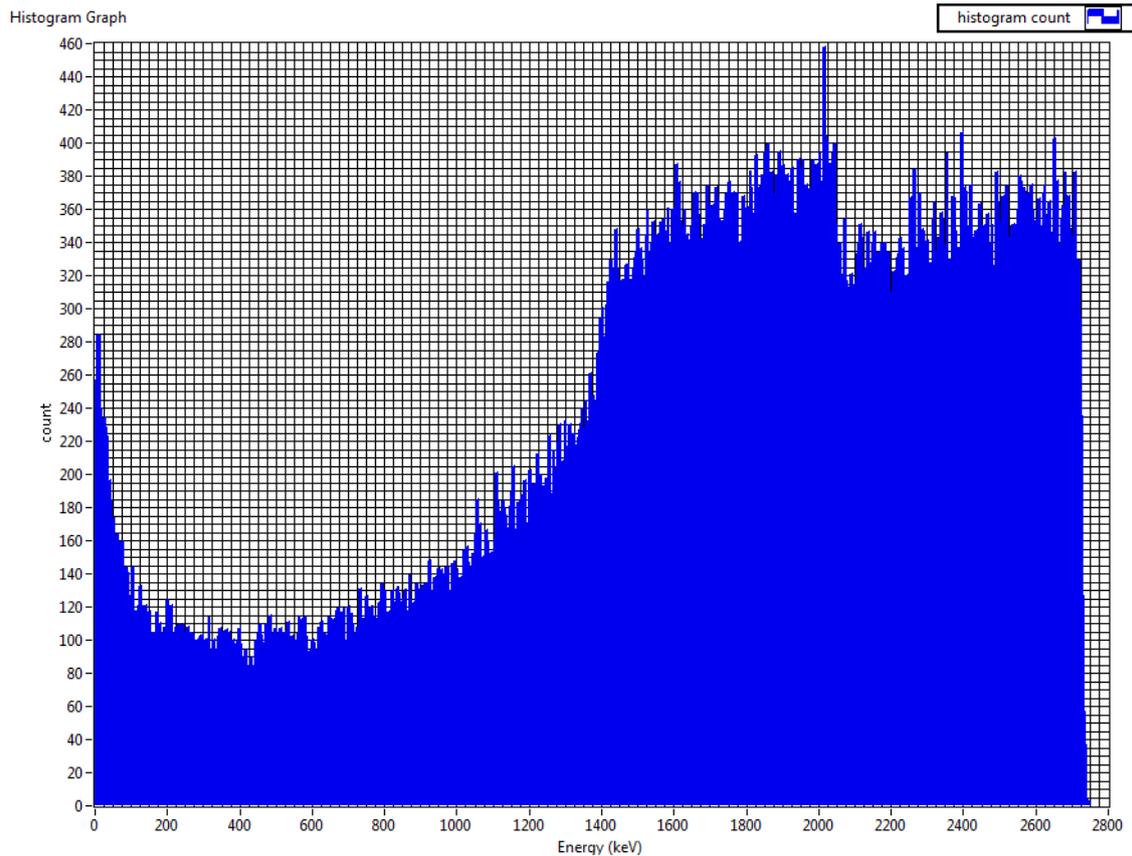


Figure 43: 20 um 6LiF

As predicted, with a 6LiF layer of 20 μm , the alpha peak is distinguishable again. This doesn't happen because suddenly the alpha particles are more present, they actually have the same shape as before. The spectrum from the alpha barely changes from the 6 μm onwards. If we consider the thermal neutron spectrum as the superposition of two independent spectra, the one from the alpha and the one from the tritium, it can be appreciated how the contribute from the tritium keeps smoothing until – theoretically – the 33 μm , which is its range on 6LiF. From this point onwards, the only variation that should be seen on the spectra is a gradual abatement, due to the particles generated on the first layers of 6LiF unable to cross the whole distance to the detector. Such effect shall also be appreciated on the efficiency.

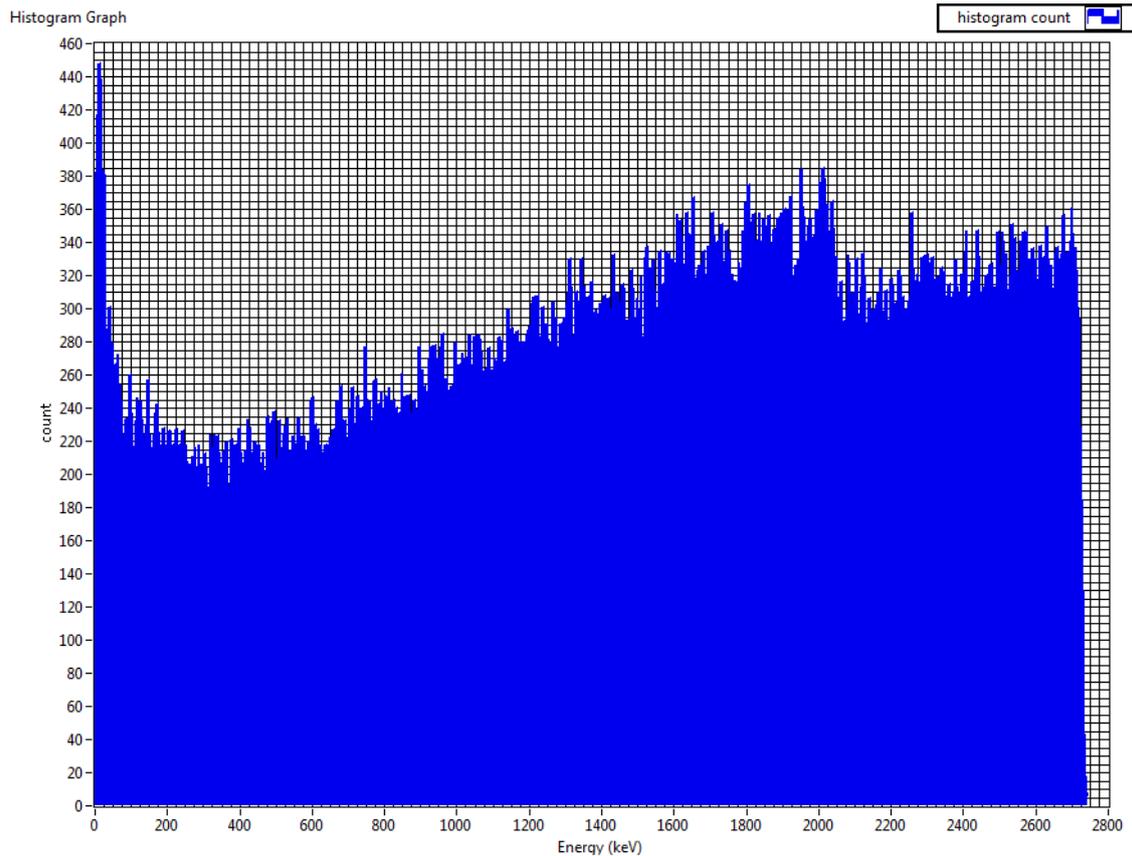


Figure 15: 32 um 6LiF

Out of all the executions done thus far, 32 μ m has proven to be the one with the highest efficiency value, as will be shown further on, equivalent to 5,62%. According to the theory, the maximum value of efficiency should be found at the range of the tritium, which is 33 μ m, and from there on there should be a linear decrease of the efficiency coefficient³. Having this result, one more execution has been done, doubling the amount of neutrons generated in order to have a more precise value for the efficiency and also

From the perspective of the spectrum, from here on we should not see variations, other than the total number of points in each channel. The form of the spectrum shall remain the same, as there won't be a change on the relative probability between both products from the reaction to reach the detector. In other words, if at 33 μ m for each alpha particle generating a signal there are 5 tritons, at 60 μ m, 80 μ m and above we shall find the same proportions. This is due to the fact that all the neutrons that collide in a distance greater than 33 μ m from the detector have a 0% chance⁴ of creating a product than can reach the Silicon and therefore create a signal for the spectrum.

³ Under very high LiF layer condition the behaviour of the efficiency curve may not be linear, but it's not a case worth studying, as it is way far the region of interest.

⁴ This is because of the definition of range: "the distance after which the probability of finding the particle is 0".

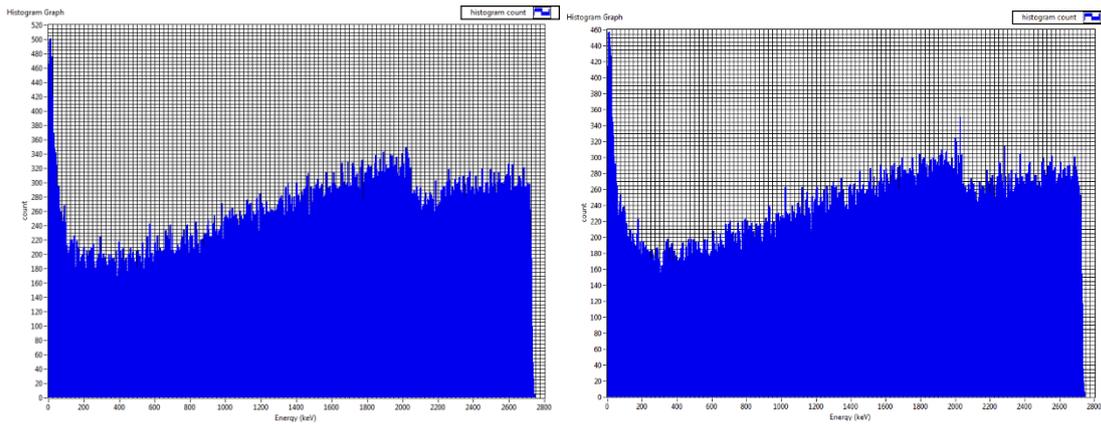


Figure 16: 48 μm 6LiF (left) and 60 μm 6LiF (right)

Just to further prove the previous affirmation, we can see that the shape of the spectrum doesn't change while varying the 6LiF height once above the 33 μm level. All the simulations have been done using 5 million thermal neutrons generated randomly. Even though this number may seem elevated, considering the resolution of 1024 channels of the histogram, it barely gives a mean of 250 counts per channel – taking into account also the fact that the maximum efficiency is around 5.6% -. In order to create a more defined spectrum for each case, it would be needed to have at least 10 times the amount of counts per channel, which would imply an increase of a full order of magnitude on the original number of neutrons used. The problem in this case is on the computational time required to do the calculations. To calculate a single spectrum using 5 million neutrons, the program takes roughly 26 hours. Attempting to make the calculation with 50 million particles would directly multiply that number by 10, thus implying at least 11 days to fulfil a single spectrum⁵. Among other tasks, future work on this project will investigate the way to improve the calculation time required for each neutron, in order to lessen the time required to create each spectrum.

The important variable that allows to take relevant data is the number of counts, which directly relates to the efficiency, that is the quotient between the number of events and the original amount of neutrons generated.

⁵ The time required for the execution in relation to the number of neutrons is further analyzed in the computational study.

6LiF(μm)	Counts	Efficiency
2	47649	0,95
4	88429	1,77
6	119111	2,38
8	139542	2,79
10	158996	3,18
12	177148	3,54
14	193418	3,87
16	208763	4,18
18	223089	4,46
20	162230	4,73
22	247742	4,96
24	258238	5,16
26	266138	5,32
28	273772	5,48
30	279269	5,59
32	281010	5,62

33	563999	5,64
34	280312	5,61
36	276606	5,53
38	273480	5,47
40	270029	5,40
42	266892	5,34
44	264414	5,29
46	260626	5,21
48	257256	5,15
50	254681	5,09
52	251846	5,04
54	247909	4,96
56	245486	4,91
58	242981	4,86
60	239102	4,78

Figure 17: Total counts and efficiency at each 6LiF height

As theoretically predicted, the maximum efficiency value is found at the point where the 6LiF height equals the range of the tritium in the material, 33 μm .

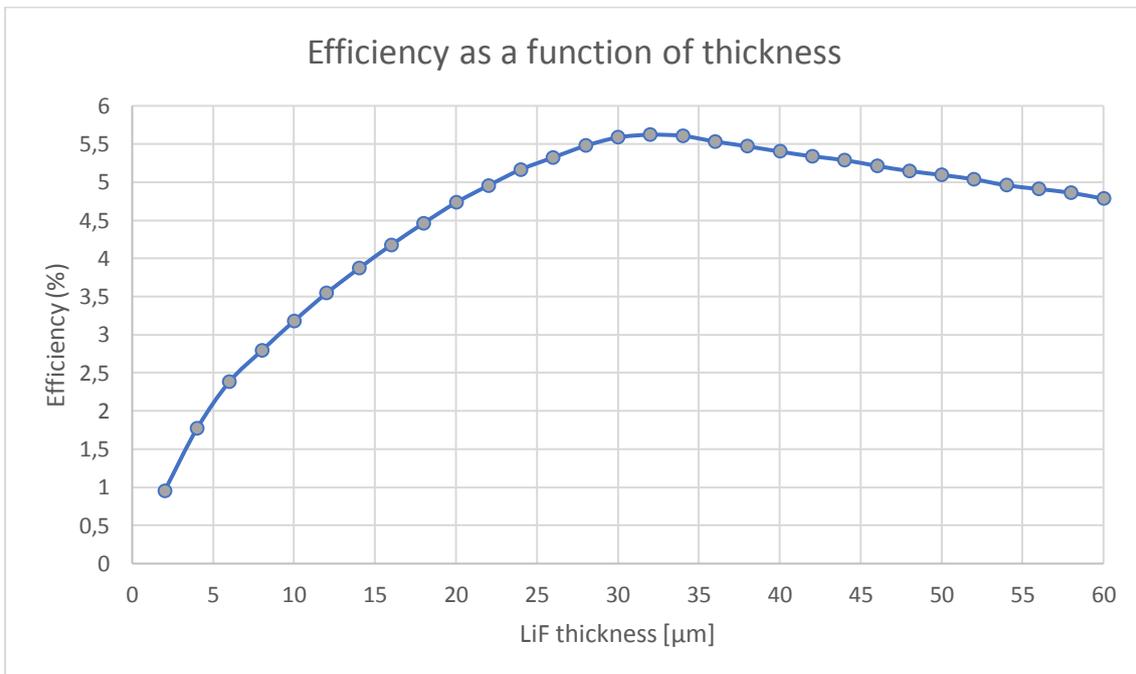


Figure 18: Efficiency graphic

Analysing the curve, it can be appreciated that is divided in three different regimes: before $6\mu\text{m}$, between $6\mu\text{m}$ and $33\mu\text{m}$, and beyond. To justify this division each section must be seen independently:

1. $< 6\mu\text{m}$: Both alpha particles and triton generated on the 6LiF successfully get to the Si detector, hence generating a signal.
2. $6\mu\text{m} < 6\text{LiF} < 33\mu\text{m}$: Since the height of the 6LiF is superior to the range of the alpha particles in the material, only those particles created on the last $6\mu\text{m}$ will get to the detector, while all the tritons created will arrive.
3. $> 33\mu\text{m}$: As the 6LiF height goes beyond the range of the tritium in 6LiF , the efficiency is constrained to linearly diminish, as only a the portion of reactions that happen on the last $33\mu\text{m}$ and generate a tritium on the favourable direction will create a signal, and only those that happen on the last $6\mu\text{m}$ and create either alpha will be counted.

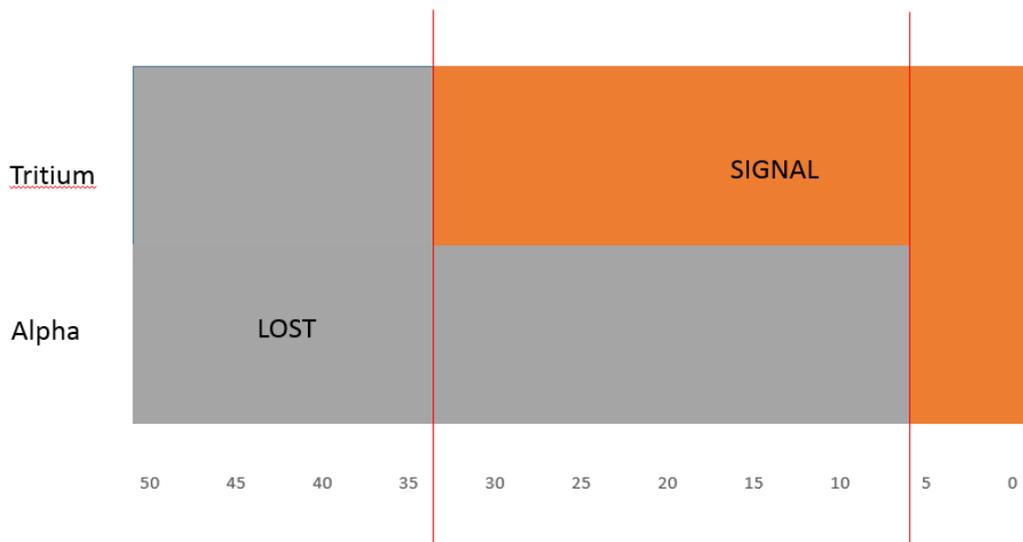


Figure 98: Graphical representation of the particles range

This image shows graphically the foretold division. It can be appreciated how all the reactions that take place further than $33\mu\text{m}$ are lost, as none of the products has the means to arrive to the detector.

CONCLUSIONS AND FURTHER WORK

This software has been developed with research purposes for the INFN (Istituto Nazionale di Fisica Nucleare), as part of the work of the LEMRAP group, and even though it has already given very positive results, there's still a lot of work to do, in order to improve it in the future, widening its capabilities. Imminently, one modification that will take the results one step closer to the reality is the adjunction of a "dead layer" between the deposit and the Silicon. This layer is essentially a thin cape (a couple micrometers thick) which causes particles that arrive there but not cross to not generate a signal, and thus the name "dead layer". Another modification that will be required is to modify the input conditions creating, instead of a plain-parallel source, an isotropic irradiation to the whole system, which will allow the comparison between both methods.

The main objective of the work, which was to simulate the behaviour of the detector under a flux of thermal neutrons in order to find what the optimal ${}^6\text{LiF}$ thickness is has been successfully achieved, and the results are coincidental with those expected theoretically. The maximum efficiency is found at a thickness of $33\mu\text{m}$. This is also proof that under this situation, the code developed shows a better behaviour, with better results than those that can be found at the current literature.

A secondary study that has been made over the software, which was the computational experiment, has shown that the behaviour of the program at big sized executions isn't linear as expected, but instead it shows an exponential evolution, indicating the need to work on both the memory and the routine in order to optimize the software in the future. One modification already being tested on the algorithm is the implementation of a diverse way to pass through the information, which would allow the program to avoid repeatedly wasting time to load the results array. By copying the output of every iteration to an temporal external file, and analysing the whole list at the end to create the histogram, the time required for each iteration would remain constant throughout the whole execution, meaning a linear growth in terms of computation time instead of the exponential case at display now. The downside with this system is that the execution time for small simulations (<2M neutron) becomes bigger, as the time required to write a file is longer than simply read/write an array. Despite this point, the advantages of the file-transfer seem to be promising. The immediate lines of work on the computation field of the software will follow this line.

BIBLIOGRAPHY

- [1] Barth, R.F.; Soloway, A.H.; Fairchild, R.G. (1990). "Boron neutron capture therapy for cancer". *Scientific American*. **263** (4): 100–3, 106–7. [Bibcode:1990SciAm.263d.100B](#).
- [2] Barth, R.F.; Vicente, M.G.H.; Harling, O.K.; Kiger, W.S.; Riley, K.J.; Binns, P.J.; Wagner, F.M.; Suzuki, M.; Aihara, T.; Kato, I.; Kawabata, S. (2012). "[Current status of boron neutron capture therapy of high grade gliomas and recurrent head and neck cancer](#)". *Radiation Oncology*. **7**: 146. [doi:10.1186/1748-717X-7-146](#). [PMC 3583064](#) . [PMID 22929110](#).
- [3] Grosswendt, B. (1990). Dependence of the photon backscatter factor for water on source-to-phantom distance and irradiation field size. *Physics in Medicine and Biology*, **35**(9), 1233.
- [4] Moore, M. J., Kasper, J. S., & Menzel, J. H. (1968). Pulsed LINAC Neutron Diffraction. *Nature*, **219**(5156), 848-849.
- [5] Bohr, N., & Wheeler, J. A. (1939). The mechanism of nuclear fission. *Physical Review*, **56**(5), 426.
- [6] Hilscher, D., & Rossner, H. (1992). Dynamics of nuclear fission. In *Annales de physique* (Vol. 17, No. 6, pp. 471-552). Les Editions de Physique.
- [7] Hawkesworth, M. R. (1977). Neutron radiography. Equipment and methods. *Atomic Energy Review*, **15**(2), 169-220.
- [8] Briesmeister, J. F. (1986). *MCNP--A general Monte Carlo code for neutron and photon transport*. Los Alamos National Laboratory.
- [9] Pelowitz, D. B., Durkee, J. W., Elson, J. S., Fensin, M. L., Hendricks, J. S., James, M. R., ... & Wilcox, T. A. (2011). *MCNPX 2.7 E extensions* (No. LA-UR-11-01502; LA-UR-11-1502). Los Alamos National Laboratory (LANL).
- [10] Vizcaíno, J. R. L., & Sebastián, J. P. (2011). *LabView: entorno gráfico de programación*. Marcombo.
- [11] Travis, J., & Kring, J. (2006). *LabVIEW for everyone: Graphical programming made easy and fun (National instruments virtual instrumentation series)*. Prentice Hall PTR.
- [12] Xuejun, X., Ping, X., Sheng, Y., & Ping, L. (2007, July). Real-time digital simulation of control system with LabVIEW simulation interface toolkit. In *Control Conference, 2007. CCC 2007. Chinese* (pp. 318-322). IEEE.
- [13] <http://w3.lnf.infn.it/laboratori/lemrap/> 04/05/2017
- [14] Metropolis, N (1987). The Beginning of the Monte Carlo Method. Special Issue 15. *Los Alamos Science*
- [1] Meyer, P., Fromm, M., Chambaudet, A., Laugier, J., & Makovicka, L. (1995). A computer simulation of n, p conversion and resulting proton tracks etched in CR39 SSNTD. *Radiation measurements*, **25**(1-4), 449-452.
- [2] Makovicka L., Ph. D. Thesis, Uni. Limoges, 1987, N°17-87.

[3] www.ni.com/white-paper/6099/en/