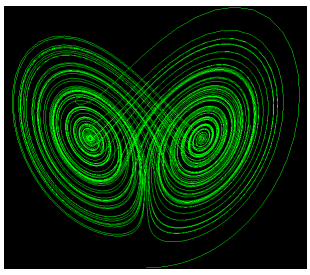
Departments of Mathematics, Physics and DACSO

MASTER THESIS

Modelling for Science and Engineering

Development of a Deterministic Neutron Transport Code Based on the Alya System at Barcelona Supercomputing Center

Carles Riera Augé



Setembre 2016

UAB
Universitat Autònoma de Barcelona

Development of a deterministic neutron transport code based on the Alya system at Barcelona Supercomputing Center

Master in Modelling for Science and Engineering

Universitat Autònoma de Barcelona Barcelona Supercomputing Center

Author: Carles Riera Augé
Directors: Prof. Mervi Mantsinen
Prof. Lluís Batet

September 2016

Abstract

In this thesis we introduce a new deterministic neutron transport code integrated in the Alya system at Barcelona Supercomputing Center, which in the longer term it is intended for multiphysics applications in the field of fusion. In order for this thesis to be self contained, first we introduce the basic physical concepts that are responsible for the production of neutrons in a fusion reactor. We also derive the strong and weak forms of the Neutron Transport Equation. Some simplifications are applied to the equation in order to ease the calculations without losing too much accuracy. Before presenting the numerical results, we also make a brief introduction to the codes used by the author. We perform two numerical calculations with our software and with other tools in order to compare them, proving that the results are accurate. In the first calculation, only 2.53% of the nodes show a difference between the two software (Zephyr and Alya) higher than 2%. In the second one, we are able to detect an exponential decrease with the distance both in the flux and the current modulus distributions. We also carry out a performance analysis of the code, focusing on the scalability and efficiency: the parallelization shows good results up to 64 processes. The last part of this work is dedicated to outline the necessary upgrades that need to be applied to the software in order to improve its capabilities. With this work we provide a good starting point for the further development of the Alya system for multiphysics applications in the fusion domain.

Acknowledgements

First of all, I would like to thank professor Mervi Mantsinen for the wonderful opportunity she gave me by accepting me into her group at the Barcelona Supercomputing Center. It has been an extraordinary experience, not only academically or professionally but also on a personal level. Furthermore, I also want to thank her for her constant dedication and willingness to help me in the process of writing this thesis. I must also thank professor Lluís Batet for the insightful comments and fruitful discussions we had, especially at the beginning of this project.

I could not have achieved half of what I accomplished in this project without the extraordinary people at the Fusion Group. Thanks to Xavi for his patience with my absurd and repetitive questions: from why something was wrong with my computer (it was usually me) to where I could find a pen or how to use the printer. To Albert, for helping me so much with all the problems that I had: he did so much more than he had to and I am truly thankful for that. Let's hope that in the upcoming months I can be useful to him in some way. And to Dani and Nathan, for their useful comments and suggestions on my thesis (thanks to Dani for carefully reviewing the whole work) and for the fun times and conversations we had over lunch so many times.

I also want to thank the Alya team for helping us develop the module. Specially to Matías Ávila, Guillaume Houzeaux and Matías Rivero: without them this project would not have been possible.

Finalment, vull donar les gràcies als meus pares, la Rosa i el Josep Maria, i a la meva germana Maria pel seu suport durant aquests anys i especialment durant aquest últim curs. Gràcies per ajudar-me sempre en tot el que han pogut i animar-me en els moments difícils.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter is divided into three sections: in the first one, we explain the motivation for our work, that is, the reasons why this work is important and useful inside the field of nuclear fusion technology. Afterwards, we explain the main goal of this thesis and the necessary steps to achieve it. Finally, the last section of this chapter describes the structure of this thesis.

1.1 Motivation

Neutronics could be defined as the field that studies the transport of neutrons and the effect that these may have in the materials or structures that they interact with. Neutronics is, in fact, the application of transport theory to the specifics of these particles. It is interesting to notice that, even though they are not radiation, the transport equation that governs their behavior is practically the same as the Radiative Transfer Equation (RTE), as we shall see in chapter 5.

The study of neutron transport is key to understand how any nuclear reactor will behave: not just fusion reactors, but also fission ones. In fact, there are entire books dedicated to the study of neutrons inside a fission reactor (see [1],[2], [3],[4]) from which the author has acquired valuable knowledge that is also applicable to fusion reactors.

There are many reasons why neutron transport is so important, which we summarized as follows:

1. The kinetic energy of the neutrons is the main source to heat up the coolant that will eventually generate electricity by rotating a turbine. Therefore, being able to harvest as much energy as possible and increase the efficiency of the reactor is in

a great deal related to studying where the neutrons will go and how to collect as much of their energy as possible.

2. At the same time, the bombardment of neutrons is one of the main sources of damage to the components of the machine. As a consequence, studying the interactions of neutrons with the materials will help to understand and prevent the effects such as erosion, cracking, embrittlement and sputtering of the machine components.

3. Related to the previous point, neutrons are also responsible for the activation of some materials, that is, making them radioactive. Therefore, it is crucial to understand how this activation may take place in order to minimize it, either by creating low-activation materials or by shielding the parts of the machine where humans might need to interact, in order to reduce their exposition to radiation.

4. Neutrons can also interact and disrupt the electronics of the machine. Therefore, it is of paramount importance to be able to predict how the neutrons may disrupt the electronics in order to place them in safer locations and prevent malfunctions.

5. In order to maximize or minimize the reactions involving neutrons, it is important to consider how their energy will affect the cross section of the reaction (see chapter 4 for the description and discussion of cross sections). Neutron moderation (slowing them down) needs to be understood correctly in order to improve the performance of the reactions.

6. Among the physical reactions, perhaps the most important one would be the tritium breeding. As we shall explain later, tritium is needed for the fusion reaction and it must be obtained from lithium in an efficient way in order for the reaction to take place (see section 2.2 for a brief description of the physics involved).

7. Neutron interactions need to be well understood in order to design effective shielding, which must protect both the superconducting magnets and the staff that might work around the nuclear reactor.

The importance of this field can also be appreciated by the fact that there are conferences in nuclear fusion dedicated entirely to this field. For instance, the author attended the "XI$^{\text{th}}$ ITER Neutronics Meeting" in May 2016, where all the points explained above were thoroughly discussed [5].

Now that we have seen the prominence of this field within the nuclear science community, it is important to explain why it is interesting to develop another neutron transport code.

There are mainly two types of neutron transport codes: Monte Carlo and deterministic. The first type is based on the idea that simulating a huge number of individual particles and their interactions with the material will be able to extract the behavior of the system as a whole. On the contrary, deterministic codes directly solve the Neutron

Transport Equation (see chapter 3). There are already several deterministic codes in the market, such as Attila [6] or Denovo [7]. However, these codes are proprietary. It would be interesting, therefore, to have our own tool that we could adapt to the needs and goals of our group and the companies of our surroundings.

Furthermore, taking advantage of other existing modules in Alya (such as fluid dynamics and thermal flows), in the long term this code could be integrated with other modules so that a multiphysics tool is developed for the field of fusion within the Alya framework.

1.2 Main objective

The main objective of this MSc thesis project was to start the creation of a neutron transport code integrated within the Alya system in the Barcelona Supercomputing Center. In order to achieve that goal, several steps needed to be carried out before that:

- Understand the theory behind the neutron transport processes. More explicitly, we needed to understand the mathematical theory describing the neutron transport, but also the physical aspects behind the mathematics: the fusion processes involved and the mechanisms of harvesting the energy.
- Acquire a general view of the neutronics field within the fusion community: what the main topics were and what the other existing tools were capable of doing.
- Detect the main challenges of the process and decide which simplifications and strategies should be carried out when designing our module.
- Learn how to use Zephyr [8], the program that served as basis for the new module, as well as other pieces of software along the way, such as Milonga [9], GID [10] or Paraview [11].
- Understand the similarities and differences between radiation and neutron transport, in order to define the correct equivalence between the different terms of the transport equations and point out the necessary modifications to model the neutron transport correctly.
- Be able to obtain similar results in Zephyr and Alya and, once that was achieved, create new examples to test the capabilities of the software.

1.3 Structure of the Thesis

The structure in which this thesis has been written follows to a high extent the process undergone by the author in order to tackle the task of building the software.

The first step was to gain understanding of the physical reality behind the model we were trying to solve. That is why Chapter 2 is dedicated to nuclear fusion: its importance as a renewable energy source, the physical reactions involved in the process and the technical facilities designed to harvest the energy yielded by these reactions. Within these topics, we introduce the roles of neutrons as products of the reactions, energy carriers and damage-causing agents.

Once we have introduced the field of nuclear fusion we dive into the mathematical modeling of the neutron transport. Therefore, in chapter 3 we introduce some useful concepts in order to derive the fundamental equation of this work: the Neutron Transport Equation. After the derivation, we carry out the simplifications of the equation in order to solve it. Finally, we also include the derivation of the weak form of the problem, since it is used in the Finite Element method which is the numerical method we use for solving the system.

In chapter 4 we introduce some concepts revolving around the concept of cross section. In particular, we discuss the different kinds of cross sections we need to consider, as well as some necessary transformations needed before being able to use the tabulated values. Chapter 5 is dedicated to explaining the Alya system, Zephyr and the module we have created ourselves. In other words, we summarize the characteristics of the software we have used. Furthermore, since Alya uses the Finite Element Method, we give a very broad view of its basic algorithm.

We present some numerical results obtained with the software in chapter 6. In particular, we first present the transition case between Zephyr and Alya. Afterwards, we move on to a two-dimensional slab of homogeneous material with the one-group discretization in the energy. In the last part of this chapter we focus on the parallelization of the code: we analyze the speed-up and the scalability of the module and we explain a typical trace of the processes taking place during an execution.

Finally, in chapter 7 we expose the conclusions of our work and in chapter 8 we explain the necessary next steps that need to be carried out in order to improve the capabilities of the software and hence the range of problems that it might be able to solve.

Chapter 2

Nuclear Fusion

In this chapter, we introduce several useful concepts regarding the field of nuclear fusion. Furthermore, we have dedicated the first section to explain why research on this topic is crucial. Afterwards, we very briefly describe the physics involved. Finally, we illustrate a simplified design of what a fusion reactor would look like, taking as a reference the ITER machine.

2.1 Importance of nuclear fusion

In this section, we present a brief summary of the current global situation regarding world energy consumption and the role that nuclear fusion can play in the future to ease the need for fossil fuels usage. The contents of this section were mainly extracted from section 1.2 of reference [12].

It is a well-known fact that an increase in the standard of living is directly associated to an increase in the energy consumed. In the past decades, the standard of living has risen not only in the developed countries, but also in other countries traditionally not included in the former category (such as China or India). In addition, the world population has been growing exponentially, from 6.13 billion people in the year 2000 up to 7.35 billion in 2015, that is, in just 15 years there were over one billion more people on the planet. Furthermore, according to the United Nations Population Division [13] we could reach 10 billion individuals by 2060.

For all we have exposed in the previous paragraph, it is clear that the energy consumption will definitely increase in the decades to come. This does not have to be a negative fact per se, but given the current global distribution of energy production in the world (see figure 2.1.1), the preeminent use of fossil fuels will lead to an even more abrupt increase in the greenhouse gases, seriously threatening the global climate of the

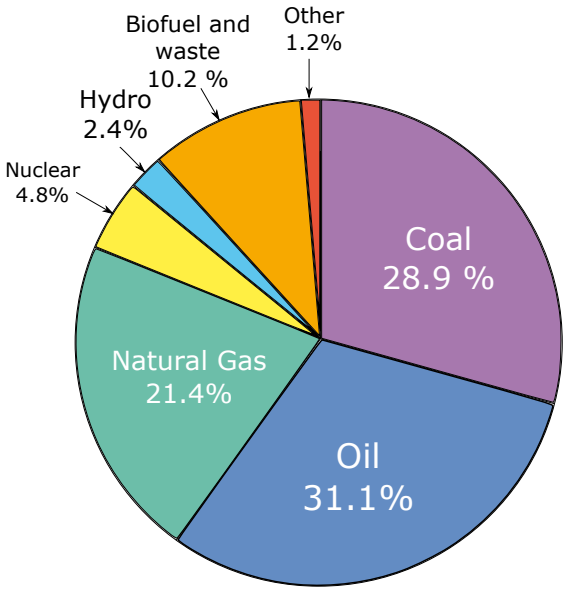


Figure 2.1.1: Distribution of worldwide energy production in 2013 (other includes geothermal, solar, wind, heat, etc.) [14, p. 6]

planet and hence the health of millions of individuals all over the world. Therefore, there is the need to reduce the dependence on fossil fuels and to find alternative energy sources.

2.1.1 Current energy sources

According to [12, p. 4], energy sources can be divided into three categories: fossil, nuclear and sunlight. Energy may also be divided into two categories depending on how it is used: direct usage or electricity production. It is in this second category that nuclear fusion can play a significant role. In the following paragraphs we will discuss the main advantages and drawbacks of the preeminent energy sources nowadays, in order to understand the role that nuclear fusion may play in the upcoming decades.

Let's start with **coal**. As it can be seen in table 2.1.1, there are still big reservoirs of this material worldwide. Its cheap price and easy access make it the preferred option for electricity production worldwide. However, it has also big disadvantages. The greenhouse gases produced during its combustion as well as the emissions produced (ash, sulfur dioxide, nitrous oxide, mercury oxides,...) are all very harmful to health. Despite all these negative aspects, it shall remain as a main contributor to energy production worldwide due to the fact that no obvious superior alternative currently exists [12, p. 7].

Compared to coal, **natural gas** burns much more cleanly (less CO_2 and harmful

Resource	Energy reserves (Quads)	Total usage (y)	Self-usage (y)
Coal	10^5	200	900
Oil	10^4	20	60
Natural gas	10^4	20	100
U235 (standard)	10^4	20	300
U238, Th232 (breeder)	10^7	20 000	
Fusion (D–T)	10^7	20 000	
Fusion (D–D)	10^{12}	2×10^9	

Table 2.1.1: Energy reserves for various primary fuels. Total usage implies 500 Quads. Self usage implies 2001 usage rate. 1 Quad=10^{18}J [12, p. 6]

components) and more efficiently. However, its greenhouse gases emissions are still very high and furthermore, natural gas reserves are much more limited than those of coal and unevenly distributed across the globe. In the short term it will continue to be used for electricity production.

So far we have talked about energy sources used to produce electricity. However, we also have to mention **oil**, which is the main fuel for transportation across the world. As the main advantages, we can mention the efficiency achieved in burning it and the fact that it is quite inexpensive. However, the main disadvantages are, once again, its greenhouse gases production and the scarce and unevenly distributed reserves. In the last recent decades there have been some alternative sources that tried to compete with oil as a fuel for transportation, such as synthetic fuel, ethanol and hydrogen. All of these might be long-term solutions, but in the short term the best strategy seems to be an increase in the usage of hybrid vehicles and more efficient ones.

Despite the public concern on its use, **nuclear fission** has some important advantages. First, it requires much less fuel to produce the same amount of energy, which implies that the reservoirs of the ^{235}U used as fuel may last several hundred years. Furthermore, no greenhouse effects or pollutants are thrown into the environment as waste products, which is a crucial factor in terms of fighting against climate change. However, other radioactive waste products are created, which need to be carefully dealt with. Another major disadvantage is that due to the complexity of the reactor the costs of building one are much higher than in the case of coal thermal plants.

Up to this point we have only talked about non-renewable sources of energy, but it is also worth focusing on the renewable sources, as they are likely to play an important role in the future as a cleaner way of producing energy. **Hydroelectric** power has the big advantage that no emissions are produced. Moreover, large quantities of energy can

be produced in a hydroelectric power plant and continuously over time. Furthermore, the conversion efficiency from the kinetic energy of the water into electricity is high, since no thermal cycle is involved. Finally, the cost of electricity is low, comparable to a coal power plant. Despite that, there is a major drawback: most rivers already have damps, and hence the possibilities of expansion of this energy source are low. Despite that, it will remain as an important energy source in the future.

Wind power has also some of the advantages of hydroelectric power: renewable source, high efficiency in conversion and no greenhouse gases or pollutants emitted. However, some drawbacks appear, as wind is not constant and predicting the harvested power is challenging: the peaks of energy production are difficult to store and huge extensions of windmills would be required to produce a significant amount of power. All these make it difficult for wind to become the main source of electricity. Instead, it should be used as a topping source of power, helping to meet peak demands.

Finally, we need to mention **solar** power: it is very attractive environment-wise, as no emissions are produced and is obviously renewable. However, it presents some of the drawbacks that wind also has: difficulty of predicting the power harvested, low efficiency and high cost of production, as the materials needed to build the solar panels are expensive to manufacture. Despite that, for residential and commercial heating it can become a very attractive option.

2.1.2 Advantages and drawbacks of nuclear fusion

According to [12, p. 16], there are three main advantages to nuclear fusion as a means for producing electricity, the first one being the fuel reserves. As we shall explain later in this work, the two elements involved in the fusion reaction are deuterium and tritium (two isotopes of hydrogen). The first one can be found naturally in deep oceanic waters, and can be extracted at a low cost: there are enough reserves to power the total consumption of electricity for more than 2 billion years [12]. However, tritium cannot be naturally found on Earth (it is radioactive with a half life of 12 years). Therefore it must be obtained from lithium, particularly the isotope ^{6}Li. Nevertheless, there are reserves of this isotope for more than twenty thousand years (at the current energy consumption rate).

As with the renewable energy sources we mentioned before, fusion power does not emit greenhouse gases or any other harmful chemical. The two products of the reaction are helium and neutrons. The first one is an inert gas that poses no threat. The neutrons, whose modeling is the topic of this thesis, are captured inside the reactor and so they are not a threat. However, they have the capacity of activating the materials of the reactor (that is, making them radioactive). Nevertheless, these activated materials have a short half-life (but should be stored for about 100 years, shorter than the time needed for the fission waste products).

The third major advantage, specially when compared to fission power, is its safety. For a fusion reaction to take place one does not need a very big quantity of fuel. On the contrary, fuel is constantly fed into the machine whilst the helium produced is removed: the total mass of fuel at any instant is very small and therefore meltdown is rendered impossible to occur.

Regarding the disadvantages, they are mostly related to the difficulty of putting this theory into practice, as it constitutes a scientific and technological challenge. First of all, the temperatures needed for fusion to occur are in the order of $150 \cdot 10^6 K$. Furthermore, the hot plasma needs to be held together inside the reactor. In order to do that, superconducting magnets capable of working at higher than usual temperatures need to be developed.

As we mentioned before, the activation of materials place an important role in the building of fusion reactor: there is a need to develop new materials that can withstand the constant bombardment of neutrons without becoming activated or damaged.

Finally, we must mention economics as a major disadvantage. The complexity of the fusion reactor and the development of all the technologies we mentioned before makes it clear that such device will not be a reality at a low price. Hence, with fossil fuels being so cheap nowadays, the competitiveness of this technology in terms of pure economical reasons is not so clear in the short term.

2.2 The physics of fusion

Nuclear fusion is the process by which two or more nuclei collide and bind together to produce a heavier one. This process can be endothermic (absorbs energy) or exothermic (releases energy). We, of course, are interested in knowing under which conditions this process will fall into the second category.

In order to understand under which conditions this process is able to release energy, we need the concept of binding energy, which is the required energy to disassemble a nucleus into separate nucleons. When in a physical reaction the final state (the heavier atom) has a higher binding energy than the lighter atoms, this implies that more energy is needed to disassemble it, and therefore the system is in a more stable state. When this happens, the process is exothermic and the system releases energy in the reaction, in the form of kinetic energy of the fusion products or gamma rays [12, p. 23]. In figure 2.2.1 one can see that this binding energy is steadily increasing for those elements with atomic mass $A < 56$, which sets the domain for exothermic fusion reactions.

This increment in the binding energy is due to the decrease in mass in the system after the reaction, due to the famous Einstein's equation $E = mc^2$. As stated in [12,

p. 23], if we have the reaction:

$$A_1 + A_2 \rightarrow A_3 + \cdots + A_k + E,$$

for certain components A_i with corresponding masses m_i, $i \in \{1, ..., k\}$ then the energy released is:

$$E = [(m_1 + m_2) - (m_3 + \cdots + m_k)]c^2,$$

which is typically of the order of $10 - 100MeV$.

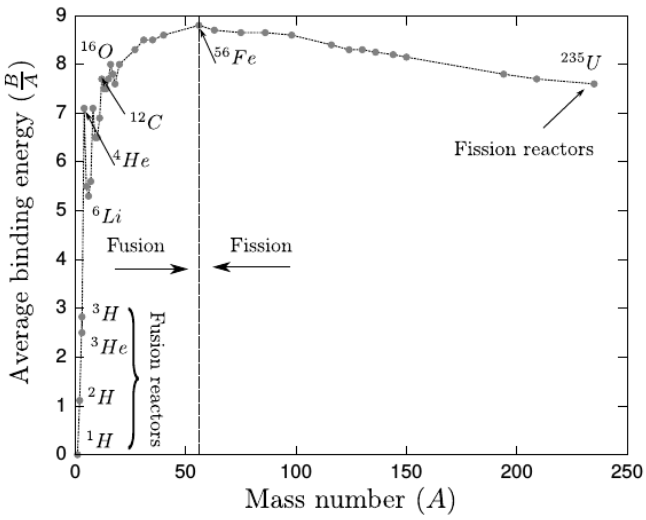


Figure 2.2.1: Average binding energy depending on the mass number (A) [15, p. 2]

In order for ions to fuse, temperatures of the order of hundreds of millions of kelvins need to be achieved, so that ions have enough energy to overcome the electrostatic repulsion and come within the range of action of the strong nuclear force (about $2{\cdot}10^{-15}$m or 2 fm), which will hold them together. At those temperatures matter is in a plasma state, which is characterized by the fact that a large number of atoms have been ionized (the electrons have left the atom orbitals). According to [16, p. 2]: "*A plasma is a quasineutral gas of charged and neutral particles which exhibits collective behavior.*"

Moreover, besides high temperatures a large enough density is also needed, as the electromagnetic repulsion diminishes the probability of collision. Therefore, higher density of particles will imply a higher probability of collision. Finally, confinement is also crucial, as plasma tends to expand due to the high velocities of the nuclei.

There are three nuclear fusion reactions that could be considered for the production

of energy [12, p. 26]: D-D (two branches), D-^{3}He and D-T which we write below:

$$\mathrm{D} + \mathrm{D} \rightarrow {}^3\mathrm{He} + \mathrm{n} + 3.27 \text{ MeV}, \tag{2.2.1}$$

$$\mathrm{D} + \mathrm{D} \rightarrow \mathrm{T} + \mathrm{p} + 4.03 \text{ MeV}, \tag{2.2.2}$$

$$\mathrm{D} + {}^3\mathrm{He} \rightarrow \alpha + \mathrm{p} + 18.3 \text{ MeV}, \tag{2.2.3}$$

$$\mathrm{D} + \mathrm{T} \rightarrow \alpha + \mathrm{n} + 17.6 \text{ MeV}. \tag{2.2.4}$$

Ideally, the first two reactions would be the perfect candidates for the production of energy. Their drawback is that in order for the deuterium nuclei to fuse, they need to be at a very small distance (of the order of an atomic radius). However, in order to do so they need to overcome an extremely high Coulomb potential, rendering them non-profitable.

The D-^{3}He reaction has also some drawbacks. Its main problem is that ^{3}He is not naturally available on Earth. This, combined with the difficulty to initiate the reaction are the reasons why current research is not focused on this reaction.

Finally, we arrive at the D-T reaction. It is the easiest reaction to initiate among the four, but its main drawback is that it involves tritium, a radioactive isotope of hydrogen with a half-life of 12.26 years [15, p. 2] that is not naturally present on Earth. However, tritium can be bred from lithium. More specifically, it can be obtained from ^{6}Li through the following

$$^6\mathrm{Li} + \mathrm{n} \rightarrow {}^4\mathrm{He} + \mathrm{T} + 4.8 \text{ MeV}. \tag{2.2.5}$$

Since deuterium can be obtained from sea water, the overall reserves for this kind of reaction are in fact limited by the lithium reserves. However, this should not be a concern either, because according to [12, p. 17]: "*Geological estimates indicate that there is on the order of 20000 years of inexpensive ^{6}Li available on Earth (assuming total world energy consumption at the present rate)*".

It should be noted that in equations 2.2.1 and 2.2.4 neutrons appear as products of the reactions and in equation 2.2.5 they are a key element in order for the reaction to take place. It can therefore be seen how important these particles are in the whole process of fusion reactions: on the one hand, they are the carriers of the kinetic energy that later on will be transformed into electricity. On the other hand, neutrons are also necessary to produce the fuel (tritium) that will be used in the fusion reaction.

Besides that, neutrons coming out of the reactions are responsible for the damage in the machine components and in consequence their trajectories and effects need to be well understood (a particularly worrisome example of this would be the effects of the neutrons on the electronics of the machine). Moreover, neutrons are also responsible for the activation of materials: effective shielding structures must be designed in order to protect the people that might need to interact with the machine from the neutrons themselves but also from the radioactive decay coming from the activated materials.

2.3 Description of a fusion reactor

As we mentioned in the previous section, there are certain conditions that need to be fulfilled in order to achieve nuclear fusion. Regarding confinement of the plasma, there are basically two different approaches: inertial confinement and magnetic confinement [17, p. 19].

In the first approach, a pellet containing a mixture of deuterium and tritium is heated and compressed by directing lasers or ion beams onto the sample. As a consequence of the energy received, the first layer evaporates, creating a reaction force inwards that compresses the inner layers of material, achieving high enough temperature and pressure. Its name (inertial confinement) comes from the fact that if this process takes 10^{-11} to 10^{-9} seconds the ions cannot move appreciably due to their own inertia [15, p. 3].

In the magnetic confinement, magnetic fields are used to contain the plasma and extend the time that particles are close to each other, maintaining the high pressure and density needed. Within this category we can further divide the designs into open or close configurations. For the sake of simplicity, we shall not explain all the possible configurations, but the interested reader may take a look at [17, pp. 19-24]. Within the closed configuration, we have the tokamak and the stellarator (c.f. figure 2.3.1). They both have toroidal geometry, but the sterallator is twisted over itself in order to increase the stability of the plasma.

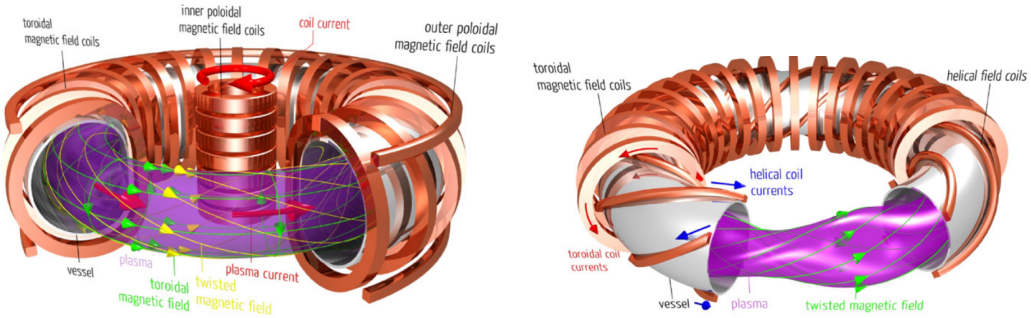


Figure 2.3.1: Simplified models of a tokamak (left) and a stellarator (right) [15, p. 5].

In this section we are going to briefly describe the main parts of a tokamak fusion reactor. We have based most of the descriptions below on the ITER machine (see figure 2.3.2. For a detailed description of each of the components, including 360° models of some of the components, the reader is referred to [18]. There are also machines based on the stellarator model (such as Wendelstein-7 at IPP, see [19]), but for the sake of simplicity we are only going to focus our attention on the former.

The production of electricity in a tokamak (which comes from an acronym in Russian standing for "toroidal chamber with magnetic coils") uses the same procedure as in most

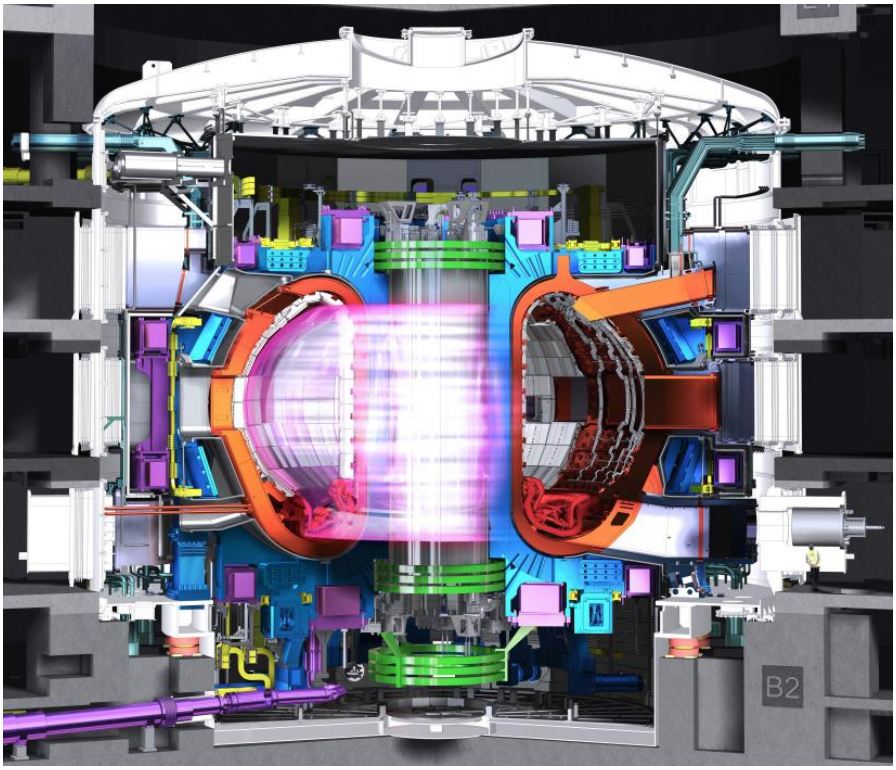


Figure 2.3.2: General design of the ITER reactor [18]

of the other power plants, that is, harness the heat generated in the interior of the reactor in order to produce steam that will move a turbine. This, in turn, will produce the desired electricity. The difference is, of course, the way in which this heat is produced.

A major difference between a tokamak and any other power reactors is the shape of the vacuum chamber (where the fusion reactions are contained), which is toroidal. This is the most suitable shape in order to control the plasma with superconducting magnets and confine it away from the walls.

With the help of auxiliary heating methods, the plasma reaches temperatures of around $150 \cdot 10^6$K. At this point the fusion reactions we describe in the previous section start to take place and the thermal energy is released.

There are several interesting parts in the tokamak. In the next paragraphs we briefly describe them, paying special attention to the blanket-shield structure due to its close relation with the field of neutronics.

The first part we shall describe is the **vacuum vessel**, which is where the fusion reactions take place. It is a hermetically sealed container in the shape of a torus, where

the particles spiral around continuously.

Creating a vacuum inside this container is crucial, since it reduces the possible impurities that might be present, improving the reaction rates and the stability of the plasma. Circulating through the walls of this vessel there will be cooling water in order to remove the heat and generate electricity. It is also worth mentioning that there will be ports, or openings, in order to access the main chamber. They are needed for repairing or changing any components that might get damaged without needing to dismantle the whole structure, and also for diagnostic purposes.

As a final remark on this component, one should bear in mind that the larger the vacuum chamber the easier it is to achieve the conditions to produce significant power, as the plasma becomes easier to confine and higher energies can be achieved more easily.

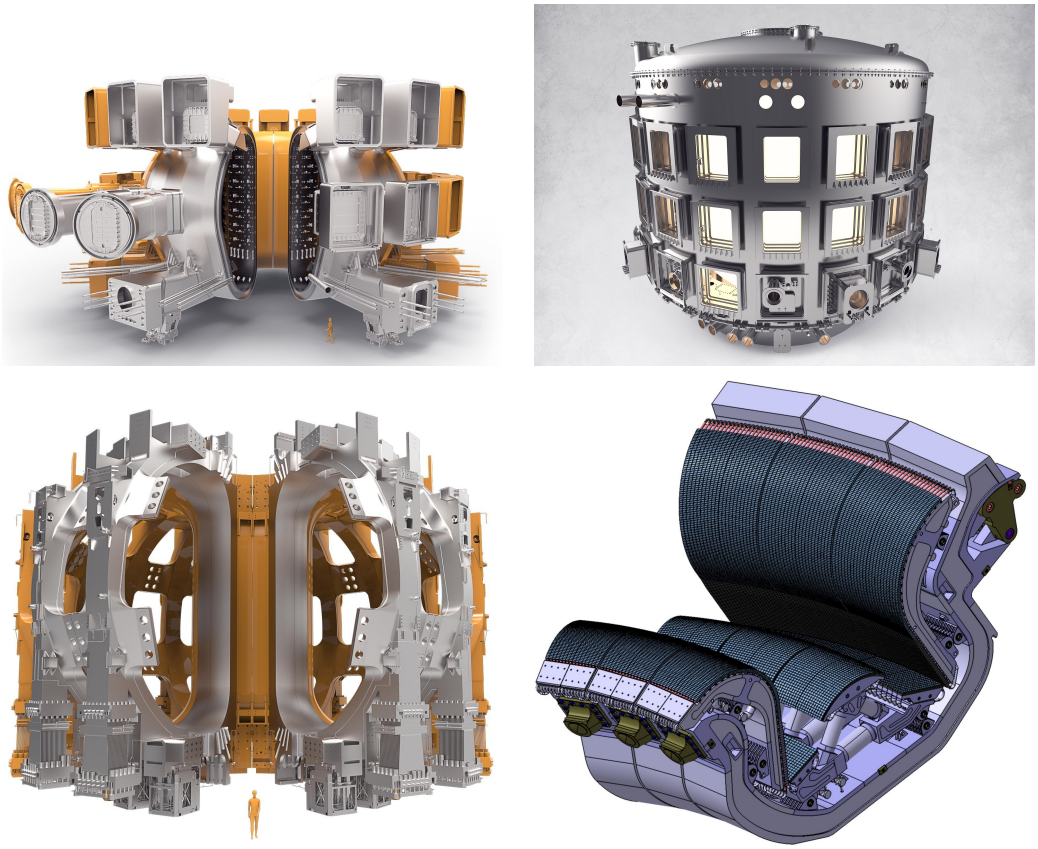


Figure 2.3.3: Different designs for ITER components. From left to right and top to bottom: vacuum chamber, cryostat, toroidal field magnets and divertor [18].

In order to confine the plasma and achieve the high temperatures needed, **superconducting magnets** need to be used. Magnets become superconducting when cooled to low temperatures (around 4K). In this state, they can carry high current and produce

stronger magnetic fields, dissipating zero ohmic power during steady state [12, p. 86] and hence the only energy required is used to maintain them at those low temperatures. This low energy requirement helps the machine improve the overall power balance, that is, the difference between the energy required to make the machine work and the energy output of the machine.

The magnets must be placed outside the vacuum vessel, protected by the blanket-shield structure in order to preserve their superconducting properties (the neutrons and the gamma rays coming from the inside of the reactor would raise their temperature).

Both the vacuum vessel and the magnets will be confined inside the **cryostat**, which is another high-vacuum pressure chamber, designed to preserve the conditions described above. It will also have penetrations both to allow maintenance operations but also to provide access for all the components related to diagnostics, auxiliary heating, cooling systems and the removal of inner components from the vacuum vessel.

At the bottom of the vacuum vessel there will be a **divertor**. Its purpose is to extract heat and residues (ash) produced in the reactions and hence minimize plasma contamination. It will have to sustain a lot of heat and neutron bombardment, and so it must be very resilient. In fact, as it is said in reference [18]: *"The plasma-facing components of the ITER divertor will be exposed to a heat load that is ten times higher than that of a spacecraft re-entering Earth's atmosphere"*. Therefore, it is estimated that the cassette assemblies of the divertor will have to be changed at least once in the lifetime of the ITER machine.

As in this work we are presenting a new neutronics tool, we are specially interested in describing the **blanket and shield structure** of the reactor, because it is in those regions where the interactions between the neutrons and the materials are key to the behavior of the machine.

In figure 2.3.4 one can see a simplified model for this component. The reader should bear in mind that all the functions separated in the model are actually entangled and are not separated from one another: figure 2.3.4 is just a simplification of the actual structure.

The first wall is the part of the shield that contains the plasma source. Next to it we can see the neutron multiplier: its mission is to increase the number of neutrons traveling across the blanket. It is important to have these extra neutrons in order to compensate the losses due to material interference and many other causes.

A moderator is also crucial for the process of tritium breeding: its role is to reduce the energy of the incoming neutrons. That is because in order to maximize the probability of a neutron colliding with a ^{6}Li atom and generating tritium, the neutrons need to be slowed down (these slow neutrons are called thermal neutrons). Obviously, there must also be a tritium breeding region, where the concentration of ^{6}Li is highest in order to

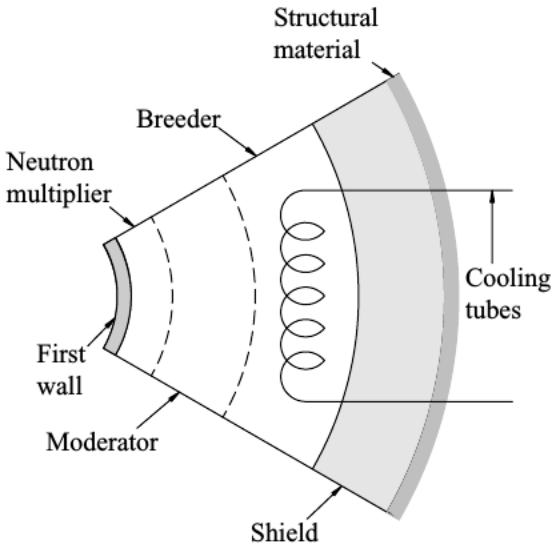


Figure 2.3.4: Simplified model of the blanket-shield structure [12, p. 93]

produce tritium.

At the outermost layer, we have the shield. It is a crucial component, since it has to prevent all neutrons from escaping the reactor, as well as to absorb the gamma rays that might have been produced by other nuclear reactions. It is required for this component to have a near-perfect absorption, mainly for two reasons. First of all, in order to protect the superconducting magnets (which work at around 4K) from heating up and losing their superconducting properties. Secondly, but not less important, to protect the people who might work outside the reactor from the radiation emitted from the inside.

Finally, we need to mention the cooling system. Its mission is to extract the heat produced from the fusion reactor. It is this heat that will be transferred to a thermal conversion system in order to ultimately produce the desired electricity, accomplishing the goal for which the fusion reactor was designed.

Although it cannot be appreciated in figure 2.3.4, it should be noted that the blanket-shield structure has a modular construction, that is, it can be assembled and disassembled by parts, in order to simplify the tasks of maintaining and repairing the machine.

Chapter 3

The Neutron Transport Equation

The aim of this chapter is to present the mathematical theory of neutron transport. The first section is dedicated to the derivation of the Neutron Transport Equation (NTE) from basic physical assumptions. Later on, we explore the simplifications that we can carry out so that we lose the least amount of precision but at the same time calculations become much more manageable. Once we have simplified the equation, we explore the basic boundary conditions that should be considered when tackling any neutron transport problem. Finally, we also deduce the weak form of the equation. This form of the equation is used in the Finite Elements Method, which is the method we will use to numerically solve the problem.

3.1 Definitions

Prior to the derivation of the neutron transport equation, let us define some quantities that will be useful later on.

Definition 3.1.1. *The **angular neutron density** n is such that*

$$n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) d\boldsymbol{r} d\boldsymbol{\Omega} dE$$

is the number of neutrons in the volume $d\boldsymbol{r}$ around the point $\boldsymbol{r}$ traveling within the cone $d\boldsymbol{\Omega}$ around direction $\boldsymbol{\Omega}$ with energies in $(E,\ E + dE)$ at time t.

Definition 3.1.2. *Integrating the angular neutron flux over the whole sphere we get the **standard neutron density** $N(\boldsymbol{r}, E, t)$, which accounts for the number of neutrons at $(\boldsymbol{r}, t)$ with energy E and per unit volume.*

Definition 3.1.3. *The **angular neutron flux density** φ is the product of the velocity*

module and the angular neutron density defined above:

$$\varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) = v(E) \cdot n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) = \sqrt{\frac{2E}{m}} \cdot n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)$$

where m is the mass of the neutron with kinetic energy E.

The quantity $\varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)d\boldsymbol{r}dEd\boldsymbol{\Omega}$ can be interpreted as the number of neutrons in a differential volume $d\boldsymbol{r}$ about $\boldsymbol{r}$ and energies in $(E, E+dE)$ traveling within $d\boldsymbol{\Omega}$ around $\boldsymbol{\Omega}$ per unit area perpendicular to that direction at time t. Notice that this is not a vector quantity but a scalar one.

Definition 3.1.4. *Integrating the angular neutron flux density over the directions we get the* ***neutron flux density****:*

$$\phi(\boldsymbol{r}, E, t) = \int_{4\pi} \varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)d\boldsymbol{\Omega}$$

which can be interpreted as the neutrons with energy E crossing a sphere of unitary radius centered at $\boldsymbol{r}$ per second.

Another interpretation of this quantity can be made if we express it like this: $\phi(\boldsymbol{r}, E, t) = v \cdot N(\boldsymbol{r}, E, t)$ is the total distance traveled per second by all the neutrons in a unit volume.

Definition 3.1.5. *The* ***angular neutron current density*** *is defined as:*

$$\boldsymbol{j}(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) = \boldsymbol{v} \cdot n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) = \boldsymbol{\Omega} \cdot \varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)$$

This quantity can be used to calculate the number of neutrons crossing an arbitrary plane that is not necessarily perpendicular to $\boldsymbol{\Omega}$, represented by the quantity $\boldsymbol{j} \cdot \hat{\boldsymbol{n}}$, where $\hat{\boldsymbol{n}}$ is the normal vector to the surface.

Definition 3.1.6. *Integrating the angular current density over the directions we obtain what is called the* ***neutron current density****:*

$$\boldsymbol{J}(\boldsymbol{r}, E, t) = \int_{4\pi} \boldsymbol{j}(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)d\boldsymbol{\Omega} = \int_{4\pi} \boldsymbol{\Omega} \cdot \varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)d\boldsymbol{\Omega}$$

Definition 3.1.7. *The* ***differential scattering cross section*** Σ_s *is such that*

$$\Sigma_s(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}, t)d\boldsymbol{\Omega}'dE'$$

is the probability per unit length that a neutron traveling in the direction $\boldsymbol{\Omega}$ with energy E is scattered from a cone $d\boldsymbol{\Omega}'$ around $\boldsymbol{\Omega}'$ and whose energy was between E' and $E' + dE'$.

Definition 3.1.8. *The **differential total cross section** Σ_t is such that*

$$\Sigma_t(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) d\boldsymbol{\Omega} dE$$

is the probability per unit length that a neutron traveling within the cone $d\boldsymbol{\Omega}$ around $\boldsymbol{\Omega}$ with energy between E and $E + dE$ interacts in any way with the material at position $\boldsymbol{r}$ and time t.

In the same way, we can define $\Sigma_f(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)$ as the **differential fission cross section**, where in this case we consider the probability of a fission event specifically.

Regarding these last two definitions, we are going to further develop the theory about cross sections in chapter 4.

3.2 Derivation

The change in the number of neutrons inside a constant volume V can be expressed as:

$$\frac{\partial}{\partial t} \int_V n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) dV = \int_V \frac{\partial n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)}{\partial t} dV$$

This term has to be equal to the production minus the loss of neutrons inside a volume. Among the production terms, we have three main contributions: the neutrons appearing as a consequence of atomic fission (coming from the neutron multiplier in the blanket of the reactor we described in chapter 2), the neutrons coming from other directions and energies that are scattered in the direction and energy that we are considering and finally an external source term. On the other hand, the loss of neutrons can be due to scattering to other energies and directions and leakage of neutrons across the boundary of the volume we are considering.

The losses due to interactions can be expressed with the term:

$$\int_V \Sigma_t(\boldsymbol{r}, E, t) \varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) dV$$

The leakage term is written as:

$$\int_A \boldsymbol{j}(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) \cdot d\boldsymbol{A} = \int_V \nabla \cdot \boldsymbol{j}(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) dV = \int_V \nabla \cdot (\varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) \cdot \boldsymbol{\Omega})\, dV$$

where in the first equality we used the divergence theorem and in the second one the definition of the angular current density.

The neutrons generated by fission events inside the domain can be calculated as follows:

$$\int_V \frac{\chi(E)}{4\pi} \int_0^\infty \int_{4\pi} \nu(E') \Sigma_f(\boldsymbol{r}, E', t) \varphi(\boldsymbol{r}, E', \boldsymbol{\Omega}', t) d\boldsymbol{\Omega}' dE' dV$$

where $\chi(E)$ is the energy distribution of the outgoing fission neutrons (we assume they are isotropic in all directions) and $\nu(E)$ is the number of neutrons released per fission event. Integrating over the directions, the expression above becomes:

$$\int_V \frac{\chi(E)}{4\pi} \int_0^\infty \nu(E')\Sigma_f(\boldsymbol{r}, E', t)\phi(\boldsymbol{r}, E', t)dE'dV$$

The production due to scattering from other energies and directions can be expressed as:

$$\int_V \int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}, t)\varphi(\boldsymbol{r}, E', \boldsymbol{\Omega}', t)d\boldsymbol{\Omega}'dE'dV$$

Putting all the terms together (plus another term taking into account the external sources $s(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)$) and paying attention at their signs we get:

$$\begin{aligned}
\int_V \frac{\partial n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)}{\partial t} dV &= -\int_V \boldsymbol{\Omega} \cdot \nabla\varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)dV - \int_V \Sigma_t(\boldsymbol{r}, E, t)\varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}), t)dV \\
&+ \int_V \int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}, t)\varphi(\boldsymbol{r}, E', \boldsymbol{\Omega}', t)d\boldsymbol{\Omega}'dE'dV \\
&+ \int_V \frac{\chi(E)}{4\pi} \int_0^\infty \int_{4\pi} \nu(E')\Sigma_f(\boldsymbol{r}, E', t)\varphi(\boldsymbol{r}, E', \boldsymbol{\Omega}', t)d\boldsymbol{\Omega}'dE'dV \\
&+ \int_V s(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)dV
\end{aligned}$$

As this must hold for any volume V, we can get rid of the volume integrals. Furthermore, we can express $n(\boldsymbol{r}, E, \boldsymbol{\Omega}, t) = \frac{1}{v}\varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}, t)$. Omitting the dependencies of the variables to ease the notation, the neutron transport equation becomes:

$$\frac{1}{v}\frac{\partial\varphi}{\partial t} + \boldsymbol{\Omega} \cdot \nabla\varphi + \Sigma_t\varphi = \int_0^\infty \int_{4\pi} \Sigma_s\varphi d\boldsymbol{\Omega}'dE' + \frac{\chi}{4\pi} \int_0^\infty \int_{4\pi} \nu\Sigma_f\varphi d\boldsymbol{\Omega}'dE' + s \quad (3.2.1)$$

It is worth noticing that in case we did not have a homogeneous material, all the cross sections appearing in the equation should also depend on the material present in the domain we are considering.

3.3 Simplifications

Equation 3.2.1 is an integro-differential equation dependent on seven variables (three spatial, two directional, energy and time). Solving it is a formidable task even with current supercomputers. Therefore, before we can start solving it we need to take into account several assumptions in order to simplify the problem.

3.3.1 Time dependence

As we want to develop our software to model neutrons created from fusion events, it is an interesting exercise to check the speeds at which these neutrons travel. As it can be seen in reference [12, p. 29], the energy of the neutrons coming out of the D-T reaction is 14.1 MeV. At this energies, we do not need to consider relativistic effects yet [1, p. 14]. Considering then the classical velocity $v = \sqrt{\frac{2E}{m}}$ substituting the energy and the neutron mass [1, p. 737] we get velocities of around $0.17c$. Therefore, at those energies the term from equation 3.2.1:

$$\frac{1}{v}\frac{\partial \varphi}{\partial t} \simeq 0$$

and hence can be neglected in front of the others. However, neutrons suffer a thermalization process: when they interact with the surrounding material, they lose energy over every interaction, transferring kinetic energy to the medium. Thermal neutrons are considered to have a kinetic energy of about 0.025 eV. If we calculate the classical velocity associated to this energy, then we find that $v_{thermal} \simeq 69$ m/s. Such a slow velocity (compared to the previous one) would make it necessary to consider the term we just omitted above.

Despite that, as a first approach we shall solve the stationary solution from now on. It is worth noticing that in this case we will be considering an infinite velocity of propagation. However, this is not that far away from reality when the domains we are interested in are at most tens of meters long and the velocity at which the neutrons travel are in the order of tens of thousands of kilometers per second. Hence, in case we want to consider a time-dependent neutron source we shall solve a sequence of stationary states for discrete times with the state of the neutron source at those specific times. In any case, we will consider that the system reaches equilibrium immediately.

3.3.2 Neutron sources

The final aim of the software is to calculate the neutron flux at any specific location of a domain. In particular, the walls of a fusion reactor. Since the walls of the reactor are not made of radioactive material [20, table 1] in principle we do not need to consider more neutrons entering our domain due to fission. However, as we explained in chapter 2, the blanket has a neutron multiplier layer, in which the number of neutrons is increased. Therefore, if we wanted to accurately model the wall of the reactor we would have to take this term into account. Nevertheless, as we are still in the first stages of development of the software, it is a reasonable assumption to neglect this term. In consequence, we shall consider that neutrons come from the plasma confined inside the walls, considering them as coming from an external source to the domain. From this perspective, the term:

$$\frac{\chi(E)}{4\pi}\int_0^\infty \int_{4\pi} \nu(E')\Sigma_f(\boldsymbol{r}, E')\varphi(\boldsymbol{r}, E', \boldsymbol{\Omega}')d\boldsymbol{\Omega}'dE'$$

in equation 3.2.1 can be omitted.

With this two assumptions, we can write a simplified version of the neutron transport equation as follows:

$$\boldsymbol{\Omega} \cdot \nabla\varphi + \Sigma_t(\boldsymbol{r}, E)\varphi = \int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})\varphi d\boldsymbol{\Omega}' dE' + s(\boldsymbol{r}, E, \boldsymbol{\Omega}) \quad (3.3.1)$$

3.3.3 Energy dependence

The energy of the neutrons is greatest when they are emitted, since they are highly energetic when they are produced. Afterwards, a process of moderation needs to take place in order to decrease their energy so that they interact more easily with the materials that constitute the reactor. This allows us to define a maximum energy (E_{max}) that the neutrons can effectively attain. As we mentioned before, those created from the fusion process usually come out with around 14.1MeV and therefore we will consider this as our maximum energy.

In order to deal with the energy dependence of the equation, we are going to formulate the multi-group transport equations [2, Ch. 7], which consists of dividing the energy range of the particles in several intervals and considering the properties of both the material and the neutrons constant inside each of these intervals.

Therefore, we select $\{E_g\}$ for $g \in \{1, ..., G\}$ such that $E_g \in (0, E_{max}]$ and that $E_g < E_{g-1}$ (so the highest energy is E_1). For each interval (E_g, E_{g+1}) we define the following quantities:

$$\begin{aligned}
\varphi_g(\boldsymbol{r}, \boldsymbol{\Omega}) &\equiv \int_{E_g}^{E_{g+1}} \varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}) dE \equiv \int_g \varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}) dE \\
\Sigma_{t_g}(\boldsymbol{r}, \boldsymbol{\Omega}) &\equiv \varphi_g^{-1} \int_g \Sigma_t(\boldsymbol{r}, E)\varphi(\boldsymbol{r}, E, \boldsymbol{\Omega}) dE \\
\Sigma_{s_{g'g}}(\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}) &\equiv \varphi_g^{-1} \int_g \int_{g'} \Sigma_s(\boldsymbol{r}, E' \to E, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})\varphi(\boldsymbol{r}, E', \boldsymbol{\Omega}') dE' dE \\
s_g &\equiv \int_g s(\boldsymbol{r}, E, \boldsymbol{\Omega}) dE
\end{aligned}$$

With this formulation, we are able to write equation 3.3.1 as the following set of equations:

$$\boldsymbol{\Omega} \cdot \nabla\varphi_g + \Sigma_{t_g}(\boldsymbol{r}, \boldsymbol{\Omega})\varphi_g = \sum_{g'=1}^{G} \int_{4\pi} \Sigma_{s_{g'g}}(\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})\varphi_{g'} d\boldsymbol{\Omega}' + s_g(\boldsymbol{r}, \boldsymbol{\Omega}), \quad g = 1, .., G \quad (3.3.2)$$

Notice that these G equations are linked together by the scattering term. Nevertheless, we have been able to get rid of the energy dependence. However, we must be very careful when defining these energy intervals: the cross-sections of certain materials can significantly vary in a very short energy range (see e.g. the total cross section of ^{235}U in figure 3.3.1) and very little in other ranges, so we need to look at those profiles carefully instead of defining the intervals a priori.

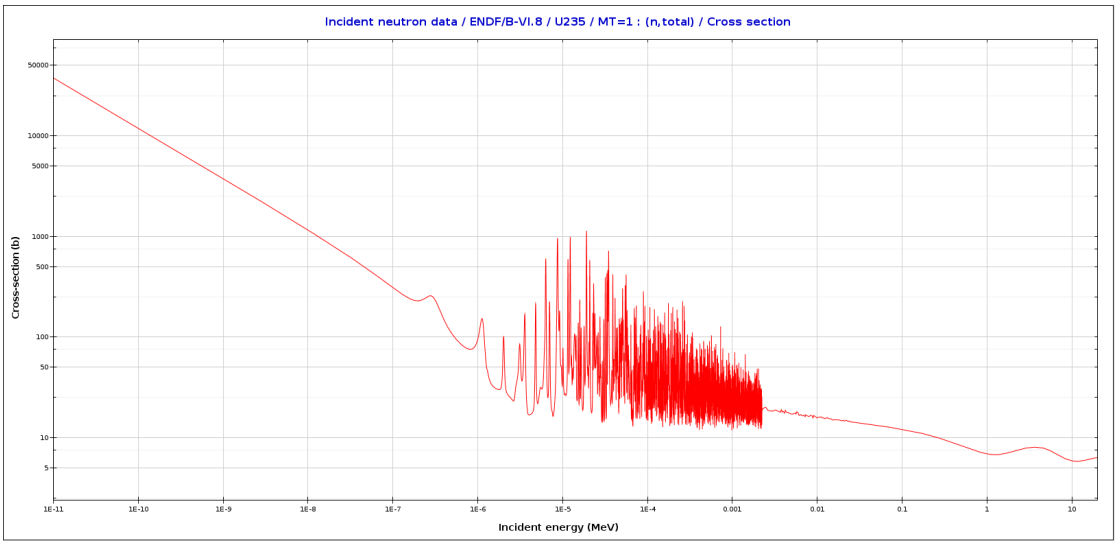


Figure 3.3.1: Total cross section (barns) vs. energy (MeV) for ^{235}U

In the first stage of our work, we shall use the **one-group** approach, which consists in choosing $G = 1$. Therefore, the quantities we defined above are just the means of the physical parameters over the whole set of energies that the neutrons might have. This is obviously a simplistic of approach, but it shall be useful to start developing the software and getting familiar with it before we move on to more realistic cases.

3.3.4 Angular dependence

In order to deal with the angular dependency of the equation, we shall use the Discrete Ordinates Method (S_N). Very succinctly, this method consists on choosing a set of discretized directions and solving the NTE for each of these directions. Obviously, we need to change the equation in such a way that incorporates transfer terms between these directions, as the scattering term links them together.

In order to deduce the discretized equation, let's start by noticing that the integral

of a scalar function $f(\mathbf{\Omega})$ over all directions can be approximated as:

$$\begin{aligned}\int_{4\pi} f(\mathbf{\Omega})d\mathbf{\Omega} &= \sum_{m=1}^{M}\int_{\Omega_m} f(\mathbf{\Omega})d\mathbf{\Omega} = \sum_{m=1}^{M}\frac{\int_{\Omega_m} f(\mathbf{\Omega})d\mathbf{\Omega}}{\int_{\Omega_m} d\mathbf{\Omega}}\int_{\Omega_m} d\mathbf{\Omega} \\ &= \sum_{m=1}^{M}\langle f(\mathbf{\Omega})\rangle_m \cdot \Delta\Omega_m = 4\pi\sum_{m=1}^{M}\omega_m \cdot \langle f(\mathbf{\Omega})\rangle_m \\ &\simeq 4\pi\sum_{m=1}^{M}\omega_m \cdot f(\mathbf{\Omega}_m) = 4\pi\sum_{m=1}^{M}\omega_m \cdot f_m\end{aligned}$$

for a discrete set of normalized weights $\sum\omega_m = 1$ and noting that $\sum\Delta\Omega_m = 4\pi$. Now if we define:

$$\varphi_{mg}(\boldsymbol{r}) \equiv \varphi_g(\boldsymbol{r}, \mathbf{\Omega}_m),$$

following the same procedure we used for the energy dependence we can discretize the integral over the directions in the scattering term of equation 3.3.2 as follows:

$$\mathbf{\Omega}_m \cdot \nabla\varphi_{mg} + \Sigma_{t_{mg}}\varphi_{mg} = \sum_{g'=1}^{G}\sum_{m'=1}^{M}\Sigma_{s_{m'm,g'g}}\varphi_{m'g'} + s_{mg}, \quad g = 1,..,G \quad m = 1,..,M \tag{3.3.3}$$

where we removed the spatial dependence for the sake of simplicity. Notice that from one equation (3.2.1) we move to $M \cdot G$ equations linked by the scattering term.

With this we have been able to remove the dependence on the angle. Nevertheless, now we can only evaluate the flux at a finite number of directions, limiting the solutions we can obtain.

In order for the S_N method to be well defined, the set of directions over which the equation will be evaluated and the quadrature weights must be chosen wisely. For instance, the set of directions $\mathbf{\Omega}_m = (\Omega_{mx}, \Omega_{my}, \Omega_{mz})$ must be unitary vectors, that is:

$$\Omega_{mx}^2 + \Omega_{my}^2 + \Omega_{mz}^2 = 1$$

and the weights must be normalized to one:

$$\sum_{m=1}^{M}\omega_m = 1.$$

There are several ways of selecting such directions. In our case, we use the complete symmetry quadrature [4], in which the directions are symmetric in each octant (the subscript N in S_N stands for the number of directions we choose per octant). Under this condition the number of directions we can choose becomes more limited. We are not going to derive which directions must be used for the different values of N, but the interested reader may find this deduction in [4, Ch. VI, sec. 3].

There is another popular method for dealing with the angular dependence, which is much cheaper computation-wise: the P_L approximation method. It consists in expanding the angular neutron flux into spherical harmonics and Legendre polynomials. However, we will not use this method due to its limitations. As mentioned in [4, p. 146]: "*The unwieldy form of the spherical harmonics equation does not lend them to easy use in practical calculations. Only in slab and spherical geometry, where there is azimuthal symmetry (...) and where the spherical harmonics become Legendre polynomials, have the above P_l equations found some general applications. In any other geometry, the azimuthal dependence has to be included, which directly complicates the equations. Therefore, one seldom meets approximations of order higher than P_1 for xy or cylindrical geometry*".

Even though in this Master's thesis we will only deal with simple geometries for which the P_L method would be the most efficient, our goal with this software is that it may be used in much more complicated geometries. Therefore, thinking ahead we decided that it was best to use a method that could be useful in the upcoming stages of the process.

3.4 Boundary conditions

In order to have a well-defined mathematical problem, we need to specify some boundary conditions on the domain where we solve equation 3.3.1. Generally speaking, there are three types of boundary conditions: Dirichlet, Neumann and Robin.

In the first type, we specify the value of the neutron flux φ at the boundary. Neumann boundary conditions are those where we specify the value of the neutron current. Finally, in Robin boundary conditions we specify a linear combination of the flux and the current at the boundary.

In general, we can specify different boundary conditions at different regions of the boundary. However, we must be careful at the points where these regions coincide in order not to impose contradictory conditions. Furthermore, in the multigroup discretization of the energy, one cannot specify different types of boundary conditions for different groups in the same region [21, p. 14].

Two boundary conditions that are commonly used in neutron transport problems are:

- **Vacuum condition**: Physically speaking, when we specify this condition we are assuming that the neutrons that cross such boundary will never come back to the domain of interest. Mathematically , this is expressed as:

$$\varphi(\boldsymbol{r}, \boldsymbol{\Omega}, E) = 0, \;\; \boldsymbol{n} \cdot \boldsymbol{\Omega} < 0, \boldsymbol{r} \in \Gamma$$

where Γ is the boundary of the domain.

- **Mirror condition**: imposing this condition implies that once the neutrons reach the boundary they will be reflected in a specular fashion. Mathematically, this is expressed as:

$$\varphi(\boldsymbol{r}, \boldsymbol{\Omega}_r, E) = \varphi(\boldsymbol{r}, \boldsymbol{\Omega}, E), \text{ if } \boldsymbol{n} \cdot \boldsymbol{\Omega} < 0, \boldsymbol{r} \in \Gamma$$

where $\boldsymbol{\Omega}_r$ is the mirror reflection of $\boldsymbol{\Omega}$ with respect to the boundary.

3.5 Weak form of the equation

In this section we follow the same derivation of the weak form of the equation as in [22, sec II.A]. This is the form of the equation that is needed in order to apply the Finite Element Method and that is the reason why we develop it now.

Let $\mathcal{R}$ be the spatial domain over which the neutron transport equation is defined. Let $V = \mathcal{R} \times \mathcal{S}^2 \times [0, \infty)$, where $\mathcal{S}^2$ is the unit sphere, be the phase-space of the problem. We define

$$\begin{aligned} \Gamma &= \partial\mathcal{R} \times \mathcal{S}^2 \times [0, \infty) \\ \Gamma^+ &= \{(\boldsymbol{r}, \boldsymbol{\Omega}, E) \in \Gamma | \boldsymbol{\Omega} \cdot \boldsymbol{n} \geq 0\} \\ \Gamma^- &= \{(\boldsymbol{r}, \boldsymbol{\Omega}, E) \in \Gamma | \boldsymbol{\Omega} \cdot \boldsymbol{n} \leq 0\} \end{aligned}$$

as the boundary of our domain, the inner boundary and the outer boundary respectively ($\boldsymbol{n}$ is the outward normal vector to the surface). Let us define the problem:

$$\begin{cases} \boldsymbol{\Omega} \cdot \nabla\varphi + \Sigma_t(\boldsymbol{r}, E)\varphi = \int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}, E' \to E)\varphi d\boldsymbol{\Omega}' dE' + s(\boldsymbol{r}, \boldsymbol{\Omega}, E) \\ \varphi(\boldsymbol{r}, \boldsymbol{\Omega}, E) = \varphi_0(\boldsymbol{r}, \boldsymbol{\Omega}, E), \qquad \text{on } \Gamma^- \end{cases} \tag{3.5.1}$$

Let us define the Sobolev spaces:

$$\begin{aligned} H^0 &= \{\psi | \int_V |\psi|^2 d\boldsymbol{r} d\boldsymbol{\Omega} dE < \infty\} \\ H^1 &= \{\psi | \int_V \left(|\psi|^2 + |\nabla\psi|^2\right)\} d\boldsymbol{r} d\boldsymbol{\Omega} dE < \infty\} \end{aligned}$$

We define the inner product as:

$$(\varphi, \psi) = \int_V \varphi(\boldsymbol{r}, \boldsymbol{\Omega}, E)\psi(\boldsymbol{r}, \boldsymbol{\Omega}, E) d\boldsymbol{r} d\boldsymbol{\Omega} dE.$$

With this we can define the L^2-norm and the 1-norm:

$$||\psi||_0 = (\psi, \psi)^{1/2}$$
$$||\psi||_1 = ((\psi, \psi) + (\nabla\psi, \nabla\psi))^{1/2}.$$

We also need to define the boundary inner products and norms:

$$(\varphi, \psi)_\pm = \int_{\Gamma^\pm} |\boldsymbol{\Omega} \cdot \boldsymbol{n}| \varphi(\boldsymbol{r}, \boldsymbol{\Omega}, E)\psi(\boldsymbol{r}, \boldsymbol{\Omega}, E) d\boldsymbol{r} d\boldsymbol{\Omega} dE$$
$$||\psi||_\pm = (\psi, \psi)_\pm^{1/2}.$$

With all these definitions we shall be able to formulate equation 3.3.1 in its weak form. Multiplying the equation by an arbitrary solution ψ and integrating over V:

$$\int_V \boldsymbol{\Omega} \cdot \nabla\varphi \cdot \psi d\boldsymbol{r} d\boldsymbol{\Omega} dE + \int_V \Sigma_t(\boldsymbol{r})\varphi \cdot \psi d\boldsymbol{r} d\boldsymbol{\Omega} dE =$$
$$\int_V \psi \left(\int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}, E' \to E)\varphi d\boldsymbol{\Omega}' dE' \right) d\boldsymbol{r} d\boldsymbol{\Omega} dE + \int_V s(\boldsymbol{r}, \boldsymbol{\Omega}, E)\psi d\boldsymbol{r} d\boldsymbol{\Omega} dE$$

Defining the collision operator as:

$$K\psi = \Sigma_t\psi - \int_0^\infty \int_{4\pi} \Sigma_s(\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}, E' \to E)\psi d\boldsymbol{\Omega}' dE'$$

and the inner products above, the equation can be stated as:

$$(\boldsymbol{\Omega} \cdot \nabla\varphi, \psi) + (K\varphi, \psi) = (s, \psi)$$

Integrating the first term on the left by parts we have:

$$-(\varphi, \boldsymbol{\Omega} \cdot \nabla\psi) + (\varphi, \psi)_+ - (\varphi, \psi)_- + (K\varphi, \psi) = (s, \psi)$$

Imposing the boundary conditions we end up with the weak form of the equation:

$$-(\varphi, \boldsymbol{\Omega} \cdot \nabla\psi) + (\varphi, \psi)_+ + (K\varphi, \psi) = (\varphi_0, \psi) + (s, \psi), \qquad \forall\psi \in H^1 \tag{3.5.2}$$

A solution φ that is valid for any ψ is said to be a weak solution of the equation. What this refers to is the fact that equation 3.3.1 is being satisfied in an integral sense, rather than a pointwise sense. Therefore, a weaker requirement needs to be fulfilled. However, if the solution φ is differentiable then it is also a solution of 3.3.1 (the proof of this equivalence can be found in [22, p. 17]).

Chapter 4

Cross sections

As we gathered information to familiarize ourselves with the problem, we realized the paramount importance of the cross sections when creating a neutron transport tool. Not only are there several concepts that need to be well understood, but the relations between those and the transformations that need to be applied to the raw data before they can be used are not trivial at all.

In this chapter we introduce some concepts that will be useful to understand the possible interactions between neutrons and the material and how the probability and outcome of these interactions can be quantified.

Definition 4.0.1. *The **total microscopic cross section** σ_t of a nucleus is a characteristic area proportional to the probability of occurrence of a nuclear reaction: the bigger the area the more probable the event.*

Usually the microscopic cross section is expressed in *barns* (b) and $1b = 10^{-28}m^2$. We use the term *total* to point out that we are considering any interaction between the incident neutron and the atom. However, we could also consider the probability of a specific event, as we shall see later. As it is pointed out in [12, p. 46], in order to correctly estimate the microscopic cross sections one needs not only to take into account the classical Coulomb repulsion between the particles but also quantum effects.

These microscopic cross sections depend on the target nucleus properties and also on the energy of the incident neutron, creating resonance effects as it can be clearly seen in figure 3.3.1. Furthermore, they also depend on the temperature T of the medium, which defines how the atoms move due to thermal agitation and, consequently, a Doppler effect takes place (see [23]). Therefore, in fact we should write $\sigma_t(E, T)$.

In the deterministic approach of neutron transport, we do not work with microscopic cross section but with their macroscopic counterparts, which may be defined as follows:

Definition 4.0.2. *The **total macroscopic cross section** Σ_t of a medium is such that*

$$\Sigma_t \cdot dx$$

is the probability that a neutron traveling a distance dx interacts with an atom of the material. In other words, it is the number of expected collisions per neutron and unit of distance.

The macroscopic cross sections depend on the energy of the incident neutron and the properties of the medium. As these usually depend on the spatial domain, in general these cross sections depend both on energy and space, i.e. $\Sigma = \Sigma(\boldsymbol{r}, E)$.

From the microscopic cross sections, one can obtain the macroscopic ones as follows: if we define the **number density** as the number of nuclei of an element i per volume unit: $ND_i = \frac{\rho_i N_A}{A_i}$, where ρ_i and A_i are its density and atomic number and N_A is Avogadro's number, then we can define the **macroscopic cross section** of type x of element i as:

$$\Sigma_{x,i}(\boldsymbol{r}, E) = ND_i(\boldsymbol{r}) \cdot \sigma_{x,i}(E)$$

where $\sigma_{x,i}(E)$ is the microscopic cross-section of type x and element i.

As we mentioned before, we can generalize the concept of cross section by defining the cross section of event k (Σ_k). In our particular case, this k could be specified as:

- $t \rightarrow$ **total**: any event can happen.
- $c \rightarrow$ **radioactive capture**: the neutron is absorbed into the nucleus creating a heavier isotope.
- $f \rightarrow$ **fission**: the neutron triggers a fission reaction.
- $a \rightarrow$ **absorption**: the neutron is absorbed by the nucleus, independently of the outcome (capture and fission are consequences of absorption).
- $s \rightarrow$ **scattering**: the neutron is scattered into another direction or energy. It can be further split into elastic or inelastic scattering.

Regarding this last type of event, it is useful to define the scattering cross section again:

Definition 4.0.3. *The **differential scattering cross section** Σ_s is such that*

$$\Sigma_s(\boldsymbol{r}, E' \rightarrow E, \boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega}, t) d\boldsymbol{\Omega}' dE'$$

is the probability per unit length that a neutron traveling in the direction $\boldsymbol{\Omega}$ with energy E is scattered from a cone $d\boldsymbol{\Omega}'$ around $\boldsymbol{\Omega}'$ and whose energy was between E' and $E' + dE'$.

In order to obtain the numerical values for the different cross sections, we use the Evaluated Nuclear Data File (ENDF) libraries. As it is said in reference [24, p. 4]: *"The ENDF libraries are a collection of documented data evaluations stored in a defined computer-readable format that can be used as the main input to nuclear data processing programs"*. By evaluated data it should be understood that cross sections are experimentally measured, combined with the predictions of nuclear model calculations and finally parametrized and tabulated to produce the evaluated data sets [24, p. 2].

Also in reference [24] one can see all the reactions, parameters and materials stored in these libraries. We are not going to explain how they work, as this is beyond the extent of this work, but we are going to point out a couple of considerations that need to be taken before using the numerical values stored in the files.

In order to perform these transformations there are several tools, one of them being NJOY [25]. As it is said in [26]: *"The NJOY Nuclear Data Processing System is a modular computer code designed to read evaluated data in ENDF format, transform the data in various ways, and output the results as libraries designed to be used in various applications. Each module performs a well defined processing task. The modules are essentially independent programs, and they communicate with each other using input and output files, plus a very few common variables"*. Just by the sheer number of modules (about 24) available inside NJOY one realizes how many changes are needed before the data is ready to use. We will not describe every single one of these transformations, but simply point out a couple of the most relevant ones for this work.

The first one is **Doppler Broadening**. As the material has a certain temperature, the atoms inside vibrate continuously. As the probability of an event occurring depends on the relative velocity of the neutron and the nucleus, the fact that the nucleus is vibrating augments the range of energies of the neutron that may be resonantly absorbed. This effect can be seen in figure 4.0.1.

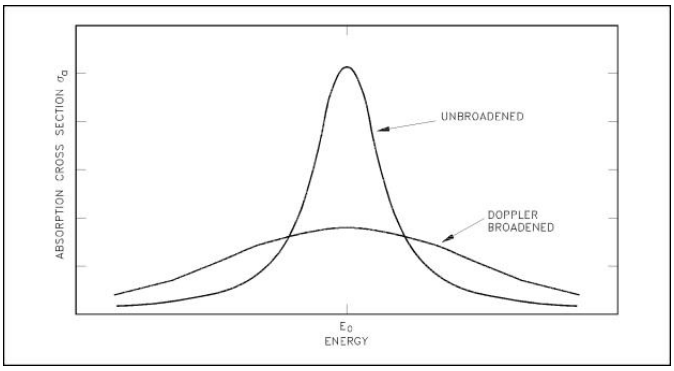


Figure 4.0.1: Doppler broadening for the cross section around energy E_0 [23]

The second transformation we discuss is creating the **multigroup libraries**. As it

was explained in chapter 3, in the multigroup approach cross sections are considered to be constant within that group. Therefore, an averaging process needs to take place for every interval of energies defining each group. Furthermore, the scattering cross sections are no longer scattering from continuous to continuous variables, but from a finite basis of energies and directions. Therefore, scattering matrices need to be created for any partition of the direction and the energies (this process is carried out by the module GROUPR, as it can be seen in [26]).

Chapter 5

The codes

5.1 The Alya System and Zephyr

In the introduction it was explained that our goal with this work was to start the creation of a new module for neutron transport simulations within the Alya system. In this section we briefly describe the features of Alya and how the new module will integrate into the code.

As it is said in [27]: "*The Alya System is a Parallel Computational Mechanics (CM) code with two main features. Firstly, it is specially designed for running with the highest efficiency standards in large scale supercomputing facilities. Secondly, it is capable of solving different physics, each one with its own modelization characteristics, in a coupled way.*"

Alya has a modular architecture, that is, it is composed by modules, services and a kernel. Each module will solve a particular physical problem. For example, there are modules dedicated to incompressible and compressible fluid dynamics, turbulence, non-linear solid mechanics, species transport, excitable media, thermal flows and N-body collisions. Our aim is to create another module that will solve the neutron transport equation. Furthermore, through explicit coupling the system is able to solve multiphysics problem, which means that in the longer term we should aim for a coupling of our neutronics module with other existing modules.

In addition to modules, the Alya system consists of a number of services. They are independent procedures that can be called by the different modules and the kernel. We mention parallelization and optimization techniques as examples of services.

Finally, we have the kernel, which contains the core of Alya. It has anything that might be needed in order to run an individual problem sequentially. Among these, we

must mention: data and geometry reading, mesh and geometry computation, calling for modules and services, control of the workflow and setting the output of the result files. It also has some basic mathematical tools.

Some input data files need to be created in order to specify the required services and modules used in each particular problem. For example, let's assume that we want to solve a problem named `test`. Then we need to create three files for the kernel:

- `test.dat`: contains the general data for the run, such as the modules and the services we want to use.
- `test.dom.dat`: contains the domain input data (mesh description), such as the dimensions, the solving strategy, the geometry and the boundary conditions.
- `test.ker.dat`: contains the input data for the Kermod. This would be like a mandatory module that can sometimes solve some simple PDE's and is in charge of coupling them. In this file we would find the shared variables among the modules.

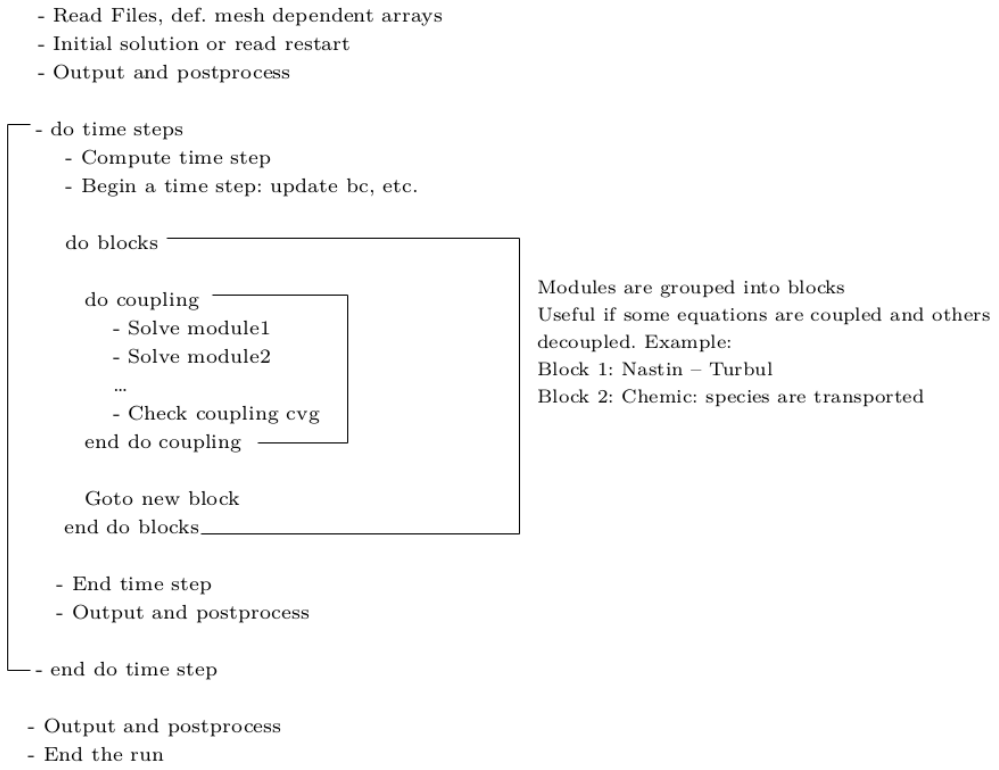


Figure 5.1.1: General workflow of the Alya system [27].

Apart from these files, there must be a file for each module we are using, where its specific parameters must be specified. In our example, this file would be named `test.neu.dat`. Should the problem require some services as well, some extra files referring to those services must be added as well. The only service we used was the parallelization, which is the only one that does not require a specific extra file, as it is specified in the `*.dat` file.

The scheme of the workflow in Alya can be viewed in figure 5.1.1. As it can be seen, there are outer loops (time iterations) and inner loops in which the equations of the modules are solved for each time step (in case several modules are at use they are organized in blocks). Should there be several coupled modules, some more iterations would need to be performed.

In our case, however, as we have not coupled our module with anything else, the workflow scheme is much simpler, as it can be seen in figure 5.1.2. Notice that even though there is a time step, we do not iterate over time or compute the size of the time step, since for the moment we are only solving the stationary case.

- Read Files, def. mesh dependent arrays
- Initial solution or read restart
  - Begin a time step: update BC, etc
  - Solve module: Neutronics
  - End time step
- Output and postprocess
- End the run

Figure 5.1.2: Workflow for the Neutronics module.

The next code that we are going to mention is Zephyr [8], which is a Stabilized Finite Element Method program developed at CIMNE (International Center for Numerical Methods in Engineering, Castelldefels, Spain). It allows to implement finite element schemes of any system of partial differential equations that can be expressed in a Convection Diffusion Reaction form. In this case, Dr. Matías Ávila adapted the code in order to model radiation transport. As it can be seen in [8, p. 121], the equation modeling the monochromatic radiative transfer is the following one:

$$\boldsymbol{\Omega} \cdot \nabla u_\lambda(\boldsymbol{r}, \boldsymbol{\Omega}) + (\kappa_\lambda + \sigma_{\Omega\lambda}) u_\lambda(\boldsymbol{r}, \boldsymbol{\Omega}) - \frac{\sigma_{\Omega\lambda}}{4\pi} \int_{4\pi} \phi(\boldsymbol{\Omega}' \to \boldsymbol{\Omega}) u_\lambda(\boldsymbol{r}, \boldsymbol{\Omega}') d\boldsymbol{\Omega}' = \kappa_\lambda I_{b\lambda} \quad (5.1.1)$$

If we compare it with the one-group approximation of equation 3.3.2:

$$\boldsymbol{\Omega} \cdot \nabla \varphi(\boldsymbol{r}, \boldsymbol{\Omega}) + \Sigma_t(\boldsymbol{r}, \boldsymbol{\Omega}) \varphi(\boldsymbol{r}, \boldsymbol{\Omega}) = \int_{4\pi} \Sigma_s(\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega}) \varphi(\boldsymbol{r}, \boldsymbol{\Omega}) d\boldsymbol{\Omega}' + s(\boldsymbol{r}, \boldsymbol{\Omega}) \quad (5.1.2)$$

we can clearly see the similarities and equivalences of the two equations in table 5.1.1.

Radiation transport	Neutron transport
$u_\lambda(\boldsymbol{r}, \boldsymbol{\Omega})$	$\varphi(\boldsymbol{r}, \boldsymbol{\Omega})$
$\kappa_\lambda + \sigma_{\Omega\lambda}$	$\Sigma_t(\boldsymbol{r}, \boldsymbol{\Omega})$
$\kappa_\lambda I_{b\lambda}$	$s(\boldsymbol{r}, \boldsymbol{\Omega})$
$\frac{\sigma_{\Omega\lambda}}{4\pi}\phi(\boldsymbol{\Omega}' \to \boldsymbol{\Omega})$	$\Sigma_s(\boldsymbol{r}, \boldsymbol{\Omega}' \to \boldsymbol{\Omega})$

Table 5.1.1: Comparison between the radiation transport equation and the neutron transport equation.

Zephyr was the basis over which the new Alya module was started. That is the reason why we include it in this work: the first benchmark consisted in checking that the Alya module and Zephyr yielded the same results for the same example. The results of this comparison will be presented in chapter 6.

5.2 The Alya neutron transport module

In this section we describe the module we have started to develop. As we have mentioned several times, its purpose is to calculate the neutron flux and current over a certain domain given some initial and boundary conditions.

The capabilities of this module are still under development. The present version has the following features:

- Two-dimensional domains have been tested.
- As we mentioned in chapter 3, only stationary problems are solved.
- Discrete ordinates method is used for the angular discretization. More precisely, S_8 and S_{10} discretizations are implemented.
- The one-group approximation is used to discretize the energy dependence.
- Only homogeneous materials can be modeled.
- Isotropic neutron sources can be included at the boundaries of the domain.

The input data files used in the module are very similar to the ones we described in Alya. For each example, we use the `.dat, .dom.dat` and `.ker.dat` that we have explained in the previous section. Furthermore, we have the `.neu.dat`, where we specify certain parameters of the module, such as whether S_8 or S_{10} should be used, the stabilization method, the maximum number of iterations before convergence, the convergence tolerance, the algebraic solver and preconditioner, the outputs of the program (flux and

current) and the type of boundary conditions that we want to apply to each boundary defined in the mesh.

The numerical method implemented in Alya is the Finite Element Method. Therefore, we made use of the already existing numerical routines to develop the module. Hence, in the next section we shall briefly describe how this method is conceived, although we are not intending to give a detailed explanation of each step.

5.2.1 Finite Element Method

Our aim is to very succinctly give a general idea on how the Finite Element Method works. In no case we shall develop a rigorous nor complete description of the method since it is beyond the scope of this work. The author thoroughly recommends the reading of other sources in order to better understand the method. In particular, references [21] and [28] were very helpful to the author.

The Finite Element Method (FEM) is based on a variational form of the differential equation derived using the method of residuals. In case of equation 3.3.1, the residual is defined as:

$$\mathcal{R}(\varphi) = \mathbf{\Omega} \cdot \nabla\varphi + \Sigma_t\varphi - \int_0^\infty \int_{4\pi} \Sigma_s \varphi d\mathbf{\Omega}' dE' - s$$

and is equal to zero if φ is the exact solution. Therefore, for an approximate solution $\varphi_h \approx \varphi$ the magnitude of the residual $\mathcal{R}(\varphi_h)$ measures the accuracy of the solution.

Multiplying the residual by a weighting function and integrating through the whole domain we have:

$$\int_\Omega w\mathcal{R}(\varphi) d\boldsymbol{x} = 0 \quad \forall w \in \mathcal{W}$$

where $\mathcal{W}$ denotes the space of admissible test functions (e.g. the Sobolev space $H_0^1(\Omega)$).

In finite element method, an approximate solution φ_h is defined as a linear combination of basis functions $\{\eta_i\}$:

$$\varphi_h(\boldsymbol{x}, t) = \sum_j \varphi_j(t)\eta_j(\boldsymbol{x}).$$

whose properties are the following:

- there exist nodes $\boldsymbol{x}_i \in \Omega$ such that $\eta_i(\boldsymbol{x}_i) = 1$ and $\eta_i(\boldsymbol{x}_j) = 0 \ \forall j \neq i$
- the restriction of η_i to each cell is a polynomial function of local coordinates.

Normally, these nodal points $\boldsymbol{x}_i$ are located at the vertices of the elements of the spacial mesh. In figure 5.2.1 we can see a linear basis function.

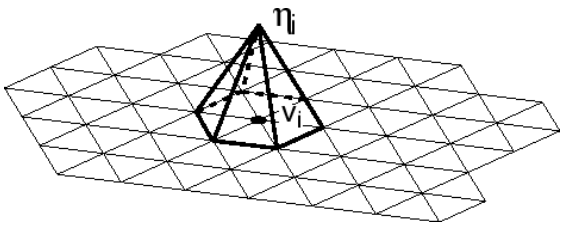


Figure 5.2.1: Depiction of a linear basis function η_j.

Let $\{\psi_i\}$ be the set of test functions. Taking $\psi_i = \eta_i$ we apply the standard Galerkin method. If instead we take $\psi_i \neq \eta_i$ then these finite element approximations are called Petrov-Galerkin methods. In our module, the latter formulation is used. Notice that, due to the way the basis functions are defined, we are not only defining the solution for the nodal points, but also inside the elements, which implies that the approximate solution is defined over the whole domain.

As it is described in [28, p. 84], the generic FEM algorithm goes as follows:

(i) **Problem statement**: derivation of the governing equations (with appropriate simplifications); possible types of initial/boundary conditions.

(ii) **Variational formulation**.

(iii) **Discretization**: Galerkin approximation to the continuous integral-form equation.

(iv) **Localization**: decomposition of global matrices and vectors into element contributions.

(v) **Generation** of a finite element **mesh** for the given computational domain.

(vi) **Assembly of the global algebraic system** from the element matrices and vectors.

(vii) Incorporating problem-specific **boundary conditions** into the algebraic system.

(viii) **Numerical solution** of the algebraic system using direct or iterative methods.

(iv) Repeating the appropriate computational steps in applications to time-dependent problems, domains with moving boundaries and non-linear problems.

(x) **Postprocessing**: computation of derived quantities and visualization.

Throughout chapter 3 we have developed steps **(i)** and **(ii)** so we do not need to explain them again.

Step **(iii)** consists in defining our solution as a linear combination of basis functions, substituting it into the weak form of the equation and then rewriting it in matrix formulation: the unknowns will be the coefficients of the linear combination proposed as solution. Furthermore, the coefficients of the algebraic system must be computed.

As explained in [28, p. 87]: "*The FEM is based on the idea of using local basis functions to construct interpolation polynomials defined on small subdomains which represent the elements of a given computational mesh*". Therefore, in step **(iv)** we need to find the local basis functions associated with an element, which are called the Lagrange basis polynomials for the local interpolation problem. We also need to find the relation between the global basis functions $\{\eta_i\}$ in terms of the local ones. As a consequence, we also split the elements of the global matrices (which involved integration over the whole domain) into a sum of integrals over each element, which constitute what we call the element matrices.

In our case, step **(v)** is carried out by GiD [10], a pre and post processor that has the capability of mesh generation: we design the domain we want to discretise, adding the corresponding Alya conditions, and the software generates the mesh according to the specified parameters.

In step (iv) we computed the local matrices for each element. Therefore, before we can solve the system, we need to assemble the global matrices (step **(vi)**) in order to apply the numerical iterative methods. This is normally done in a loop over the elements adding the contributions of each local matrix. In general, the global matrix will have a null entry in row i and column j if the supports of the basis functions η_i and η_j do not overlap.

Step **(vii)** consists in substituting the appropriate rows in the global matrices by rows of the identity matrix so that the Dirichlet boundary conditions that we want to impose are fulfilled.

In order to carry out step **(viii)**, we use the Generalized Minimal Residual Method (GMRES) [29]. Very briefly, if the system we want to solve is $Ax = b$, and we define the order-r Krylov subspace as $\mathcal{K}_r(A, b) = \text{span}\{b, Ab, A^2b, ...A^{r-1}b\}$ then the method finds the vector $x_r \in \mathcal{K}_r$ such that the residual $\mathcal{R}_r(x_r) = Ax_r - b$ is minimized. In order to ease the convergence of this method, we precondition the system re-scaling the algebraic equations in such a way so that the same value will always appear in the diagonal of the global matrix.

Since we are solving the stationary equation we do not need to perform further iterations and therefore step **(ix)** is not executed. Finally, we use GiD in order to carry out the post-processing of the data, that is, step **(x)**. In chapter 6 we show some examples of data post-processing.

Chapter 6

Results

In this chapter we present the numerical results obtained with the Alya neutron transport module. The first section is dedicated to the transition case between Zephyr and Alya, by which we checked that very similar results were obtained using both tools. Afterwards, we present a simple case designed from scratch to run in Alya: a two-dimensional homogeneous slab of material with isotropic scattering and using the one-group approximation for the energy discretization. Finally, we carry out a simple performance analysis to see how efficiently the code can be parallelized.

It is useful to recall the system that we are trying to solve in all the examples that we shall see later on. Expression 3.3.3 was the system of equations that were obtained once both the direction and the energy variables had been discretized. As we are applying the one-group approximation instead of the multigroup approach in the energy, such system of equations can be written as:

$$\mathbf{\Omega}_m \cdot \nabla \varphi_m + \Sigma_{t_m} \varphi_m = \sum_{m' \neq m} \Sigma_{s_{m'm}} \varphi_{m'} + s_m, \quad m = 1, .., M \tag{6.0.1}$$

where φ_m is the flux evaluated at direction $\mathbf{\Omega}_m$, Σ_{t_m} is the total cross section evaluated at the same direction and $\Sigma_{s_{m'm}}$ is the scattering cross section from direction $\mathbf{\Omega}_{m'}$ to direction $\mathbf{\Omega}_m$. Therefore, we present the solution of the system of equations 6.0.1 for different geometries in the following sections.

Before we analyze each example in particular, the author wants to acknowledge the fact that the following are simple and idealized cases, very far away from the physical reality we would like to model. Nevertheless, the difficulty and challenges of our work do not lie in the solution of the cases we present, but in the fact that a new tool (module) had to be integrated within an already existing framework (Alya). In other words, the bulk of the work was to integrate the set of equations into the framework and link the existing tools (such as the numerical methods for solving the equations) to the new

module rather than solving the following cases. Nevertheless, these calculations had to be carried out in order to prove that the module was working correctly and yielding accurate results.

6.1 Comparison between Zephyr and Alya

In this section, we show the comparison between the results obtained in Zephyr and Alya. The case we shall solve was proposed by Dr Matías Ávila, who provided the author with the mesh showed in figure 6.1.1: it is a $5m \times 3m$ outer rectangle with a $0.4m$-side square inside.

The physical explanation of the problem is the following: both the outer and inner (the sides of the inner square) boundaries emit neutrons (radiation). The inner part of the domain is filled with a homogeneous material with $\Sigma_s = 2m^{-1}$ and $\Sigma_t = 4m^{-1}$, which implies that the material is scattering and absorbing neutrons isotropically. Our goal, therefore, is to calculate the distribution of neutrons inside the domain.

Notice in figure 6.1.1 that quadrilateral elements of different sizes were chosen in order to take advantage of the quadrilateral geometry of the whole domain. It is also worth noticing how the left side of the mesh is finer than the right side in order to capture the expected more complex distribution due to more emitting boundaries in a smaller space.

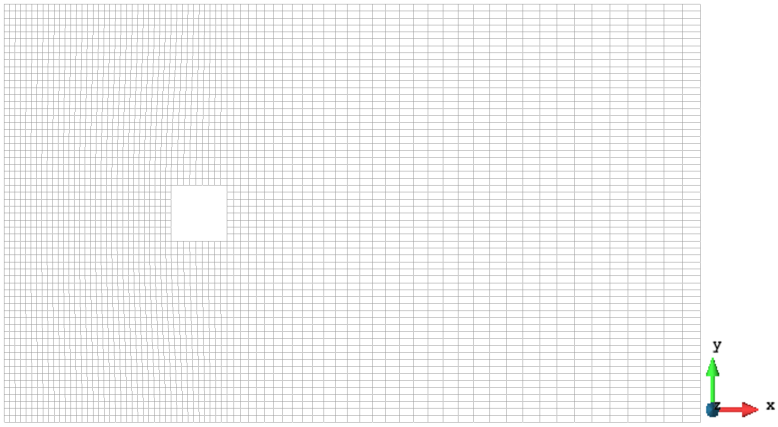


Figure 6.1.1: Mesh used in the benchmark with Zephyr (4580 nodes).

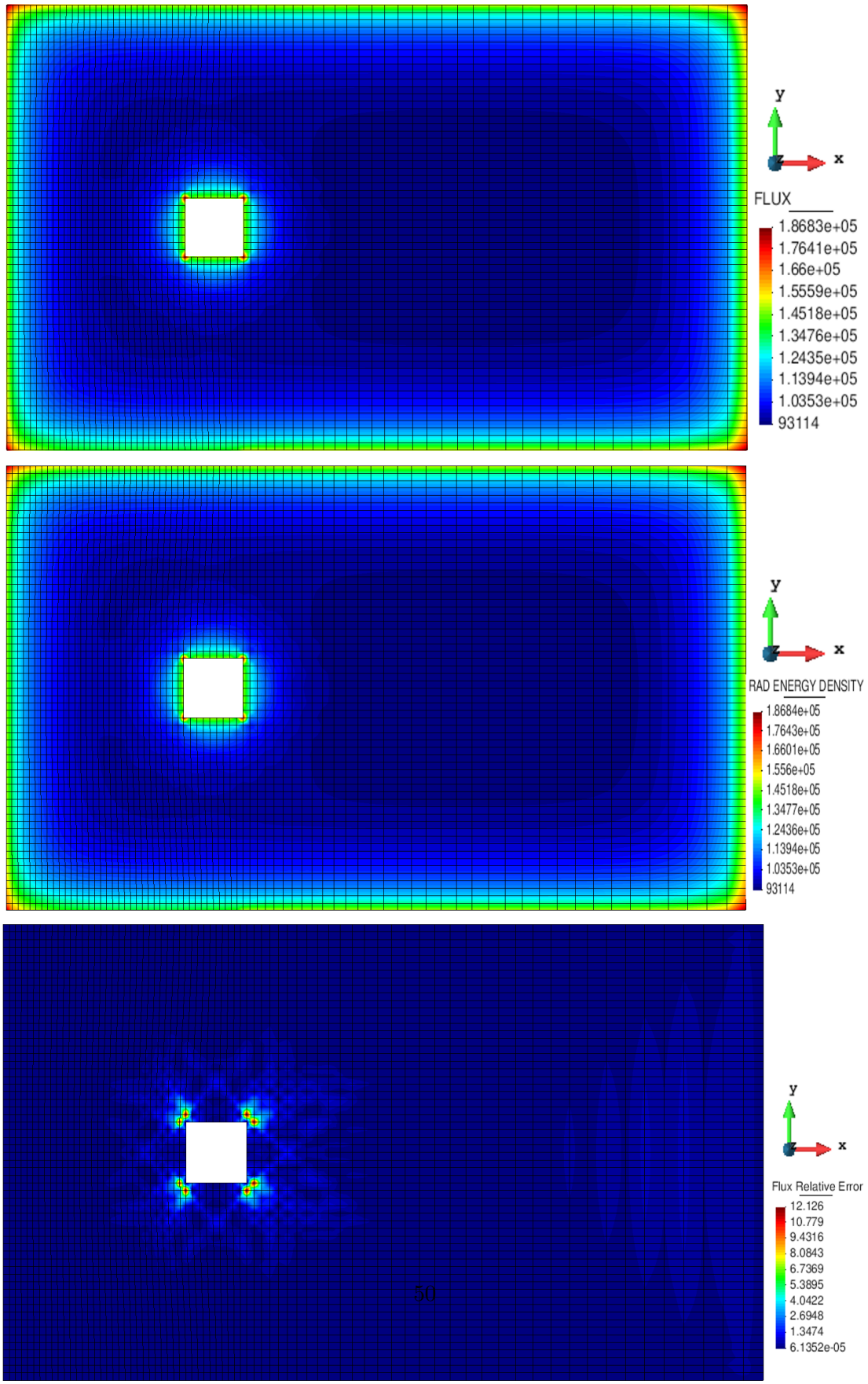


Figure 6.1.2: Alya (top) vs Zephyr (middle) neutron flux distributions with the relative error in percentage (bottom) (boundary flux=$5.67 \cdot 10^4 n \cdot m^{-1} s^{-1}$)).

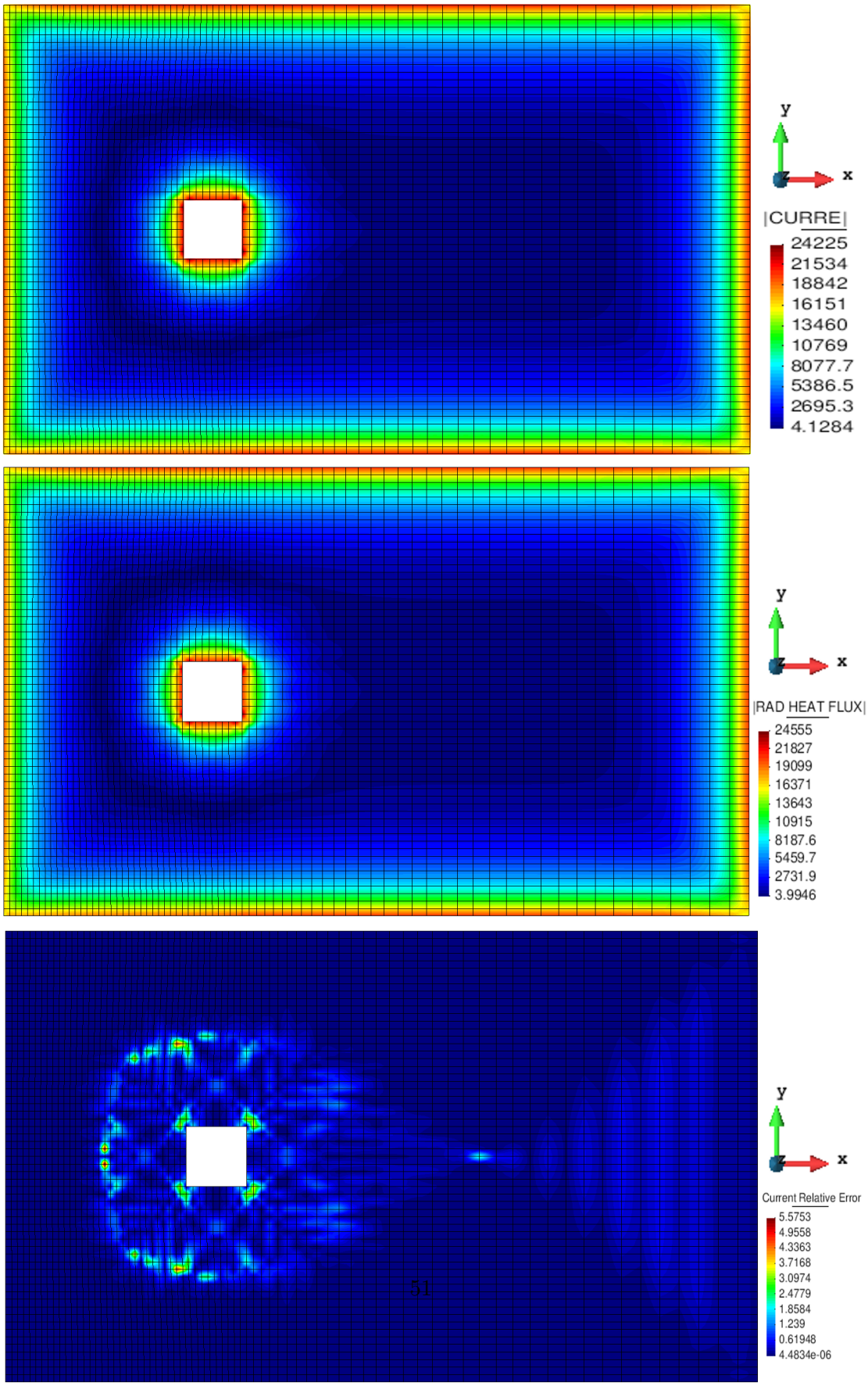


Figure 6.1.3: Alya (top) vs Zephyr (middle) neutron current modulus with the relative error in percentage (bottom) (boundary flux=$5.67 \cdot 10^4 n \cdot m^{-1} s^{-1}$).

The physical behavior of the system is in agreement with what we could expect. Starting with the neutron flux (figure 6.1.2) as the neutron source is evenly distributed throughout the borders, we can see how the neutron distribution is more or less balanced on the boundaries. It is worth mentioning how, at the corners, there is a higher number of neutrons due to the fact that the two surfaces are closer together and therefore the number of neutrons crossing that area is also higher. We can also observe the same phenomenon at the vertices of the inner square: it also makes sense that the flux is higher in that region, since the two emitting walls are also together and therefore the number of neutrons crossing that regions is also higher due to the contributions of both walls. Finally, the region with lower neutron flux is the one furthest away from the walls, which also matches what we could expect.

If we turn now to the neutron current (figure 6.1.3) we can see a similar distribution. Nonetheless, now we see a decrease in the neutron current at the corners of the domain, both at the outer and inner boundaries. The explanation for this phenomenon is simple: the neutron current is not a scalar quantity, but a vector. In this sense, if there are two streams of neutrons in opposite directions the neutron current will be zero at that region. Therefore, at the corners we have two isotropic sources of neutrons very close together, which implies that some of the neutrons coming from one source will cancel those coming from the other one. Hence, the neutron current in that direction will be smaller and therefore the modulus of the vector is also smaller than in those regions where we do not have two distinct sources.

As we can see in figures 6.1.2 and 6.1.3, the two results are in good agreement. In the flux distribution we can see how there are eight nodes surrounding the inner square that show a higher than average error and which influence the error of the distribution in that region. One explanation for this phenomenon could be that there is a high gradient of the flux in that region, meaning that if after the iterations the values did not agree exactly, later on in the interpolation this errors could be propagated and even enhanced.

On the other hand, in the current modulus we can see that the higher errors can be found in the same areas as in the flux distribution. Even though it might seem that there is a greater error, it must be pointed out that the scale range is much narrower in this case. In fact, the greatest error is about 5.6%, whereas in the flux distribution the maximum error was more that twice that figure.

6.2 One-dimensional mono-energetic isotropic scattering

In this section we present the first case specifically designed for the Alya module. It consists of a two-dimensional thin slab of homogeneous material with $\Sigma_s = 2m^{-1}$ and $\Sigma_t = 4m^{-1}$ in which the height of the domain is ten times larger than its width (in this case $10cm \times 1cm$) so that the effects of the boundaries can be minimized at the central region. The left wall is constantly emitting neutrons, and vacuum boundary conditions were implemented in the other three boundaries. Our goal, again, is to see the neutron flux and current distribution, specially at the central region of the domain.

What we would expect is a decreasing flux and current with the distance to the left boundary. In fact, as it is pointed out in [1, p. 240] the solution to this problem should follow an exponential decay with the distance with respect to the source.

Figure 6.2.1: Detail of the mesh used in the two-dimensional slab case (10201 nodes and 10000 elements).

In order to simulate this case, we used the mesh portrayed in figure 6.2.1: notice that we used quadrilateral elements with the same proportions as the domain in order to capture the rapid changes in the x-coordinate and at the same time not spend too much resources on calculating the variations on the y axis, since we should expect such changes to be minimal.

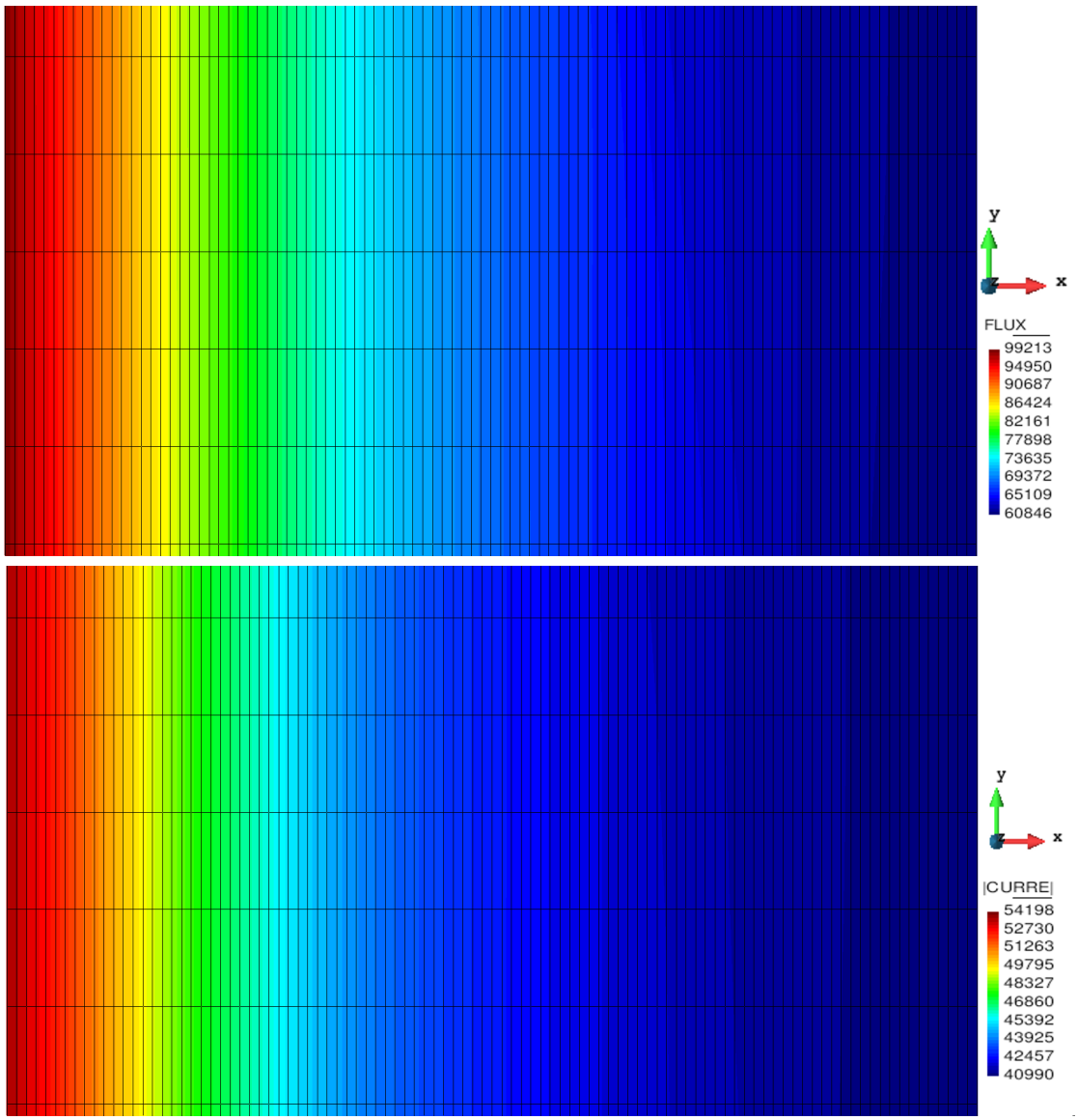


Figure 6.2.2: Neutron flux (top) and current modulus (bottom) distributions (boundary flux=$5.67 \cdot 10^4 n \cdot m^{-1} s^{-1}$).

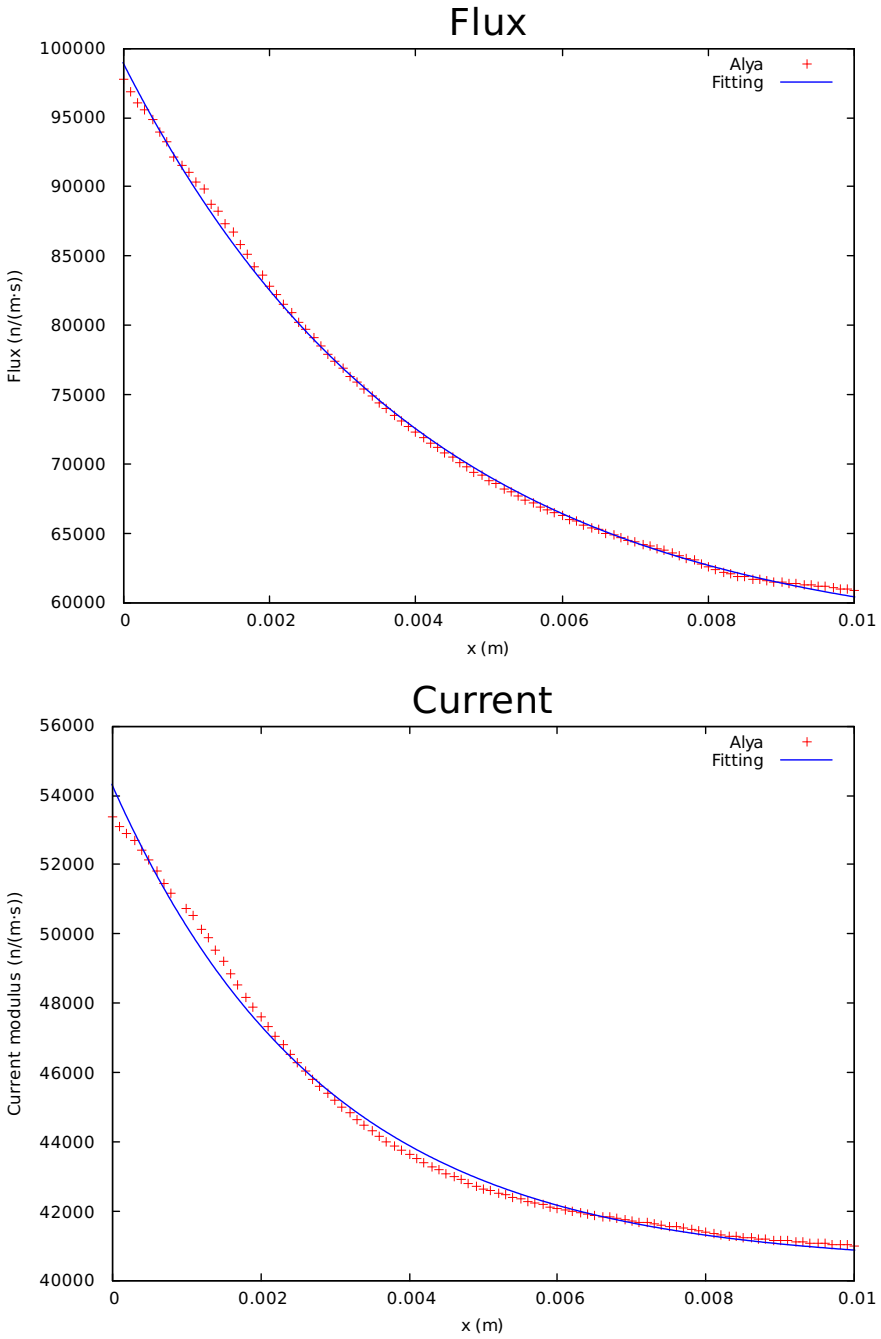


Figure 6.2.3: Neutron flux (top) and current modulus (bottom) with respect to distance at the central nodes with exponential fitting.

The numerical results obtained with Alya can be seen in figure 6.2.2, in which the central region of the domain is portrayed. It can be observed how both the flux and the current modulus have a rapid decrease in magnitude with the distance, as we should expect. However, with these images we are not able to fully distinguish the type of decay that is occurring, and so we filtered the data of the central nodes of the domain and plotted them in one-dimensional graphs, which can be seen in figure 6.2.3.

Unfortunately, we were not able to compare this case with any other software. Nevertheless, we can see how both magnitudes decay exponentially, although the flux is following the trend better than the current. However, we can see how, for very short distances with respect to the source, the decay of the current seems to follow a linear trend rather than an exponential. Further insight needs to be gained about the numerical results of this case: comparisons with other software are needed in order to correctly understand whether the behavior of the system is accurate enough or if there are any deviations with respect to the correct behavior.

6.3 Performance analysis

One of the strengths of the Alya system is its high capacity for scalability. As it can be seen in [30], the system has been efficiently scaled in more than 100,000 cores at the BlueWaters supercomputer. Since Alya is a framework aimed at efficiently solving multiphysics problems in parallel architectures, specially focused in performance and scalability, we find it interesting to analyze the behavior of the neutron transport module.

Before discussing the performance analysis, a few remarks should be made about the parallelization in Alya. First of all, the implemented parallelization strategy is the master-slave technique. What this means is that when the parallelization takes place, one process (the master) is in charge of distributing the data and coordinating the other processes. These (the slaves) are the ones carrying out the calculations. Slaves (and the master) also communicate among themselves through messages using the MPI standard, which we will briefly describe later on in this section.

The speed-up is defined as the execution time that it takes for the program to run with one process divided by the time that it takes to run with N processes. Ideally, if we executed the same case with four processes instead of one, the speed-up would be four. However, this ideal improvement is almost unattainable: the more processes we have the more communications need to take place. Furthermore, the time dedicated to the initialization of the variables and arrays cannot be improved by adding more processes, so there is a certain part of the total time that will not be reduced by parallelization.

Moreover, we also need to mention the efficiency, which is defined as the speed-up divided by the number of processes being used. If the parallelization was ideal the efficiency would be one. However, we decided to calculate the efficiency divided by the

number of processes minus one. The reason being that the process with the master role is not performing any calculations but merely communicating and distributing the workload. Bear in mind that we compare the performance with the sequential version, in which all the nodes are dedicated to computation and there are none dedicated to communication.

Implementing this parallelization is done via the MPI standard [31], which provides synchronization and communication functionalities between a set of processes with distributed memory (meaning that not all processes can access the same memory and therefore need to exchange data among them). The Alya system is written in FORTRAN language: MPI is implemented as a library within the code from which we can call the functions we may need once we have included it, as we would do with any other library.

MPI provides a wide variety of functions and abilities. Describing them is far beyond the scope of this work, but since we analyze the trace (the sequence of MPI functions that take place during the execution of a program) for the Alya module we shall introduce some very basic definitions of point-to-point communications in order to ease the task for the reader.

Point-to-point communications refers to communications that take place between specific MPI processes. `MPI_Send` allows one process to send a message to a second specified process; `MPI_Recv` will allow one process to receive a message from a specified process. Finally, collective functions involve communications between all the processes in one group. `MPI_Bcast` (broadcast) will take data from one process and send it to the rest of the processes.

In order to correctly analyze the scalability of the code, we need to create a big enough case so that the workload is large enough to be divided among many processes, in such a way that the computing time represents a major part of the execution time. Therefore we created a new mesh: it has the same shape as the one we showed in figure 6.2.1 but instead of having 10201 nodes, we create one with 10 million elements and about the same number of nodes (occupying a total of 2.3 GB in memory). We run this case for several numbers of processes, and we show the numerical results in table 6.3.1.

More specifically, we run the same case for $2^k + 1$ processes for $k = 1, ..., 7$. We choose these figures so that the number of slaves is always a power of two, since there is always one process carrying out the master role.

In figure 6.3.1 we have plotted the speed-up in a logarithmic scale. It should be noted that we considered the number of processes to be the number of slaves. We have plotted the results up to 128 processes: the problem is too small to be run efficiently for the following powers of two. In the range we have plotted, we can see that the problem scales accurately as it follows the ideal trend (shown in green) except for the last value (128) where the trend is lost. However, in figure 6.3.2 the performance is more intuitively

Number of MPI processes	Time (s)	Speed-up	Efficiency	Efficiency-1
1	22390	1	1	1
3	11766	1.90	0.63	0.95
5	6700	3.35	0.67	0.84
9	3389	6.61	0.73	0.83
17	1929	11.61	0.68	0.73
33	1046	21.41	0.65	0.67
65	600	37.32	0.57	0.58
129	394	56.83	0.44	0.44

Table 6.3.1: Execution time, speed-up, efficiency for different number of processes. Last column represents the efficiency calculated with the number of slaves.

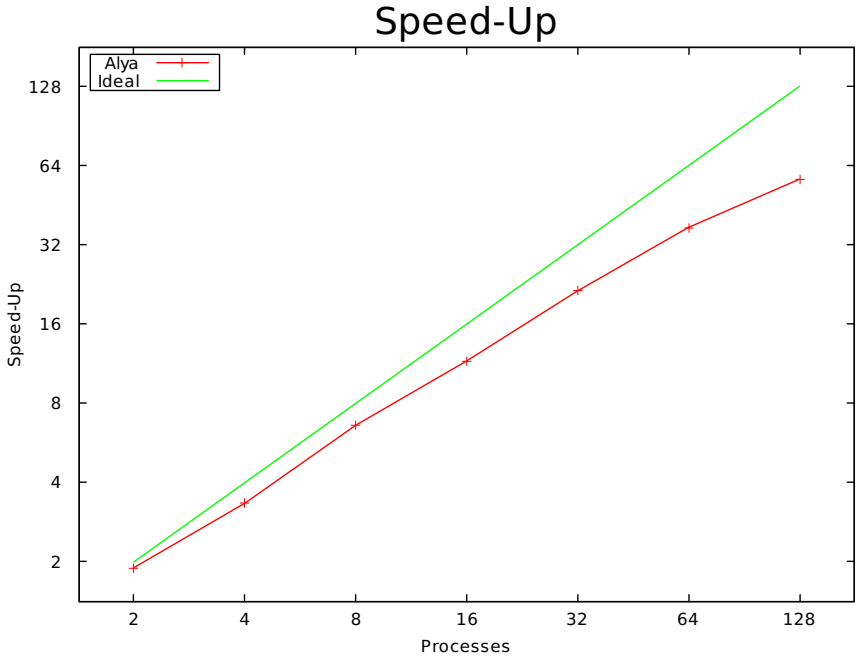


Figure 6.3.1: Comparison between ideal and Alya Speed-Ups in logarithmic scale.

visualized: if the parallelization was ideal we would have an efficiency of 1 for all cases. Therefore, those which are far away from it show poorer performance. We plotted figure 6.3.2 with the same criterion as in figure 6.3.1, that is, considering the number of slaves instead of the number of processes to calculate the efficiency.

Figure 6.3.3 shows the types of MPI tasks that take place during the execution of the previous case in the Alya module using four processes. The first colored bar shows the master process and the other three are the slaves. The initialization lasts until the end of the wide white stripes (slaves): they are waiting for the master to send the data to start

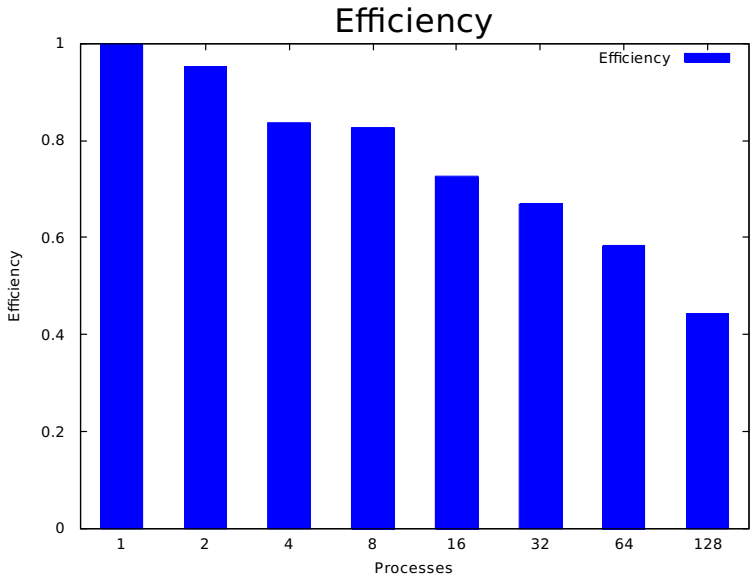


Figure 6.3.2: Efficiency for different number of processes.

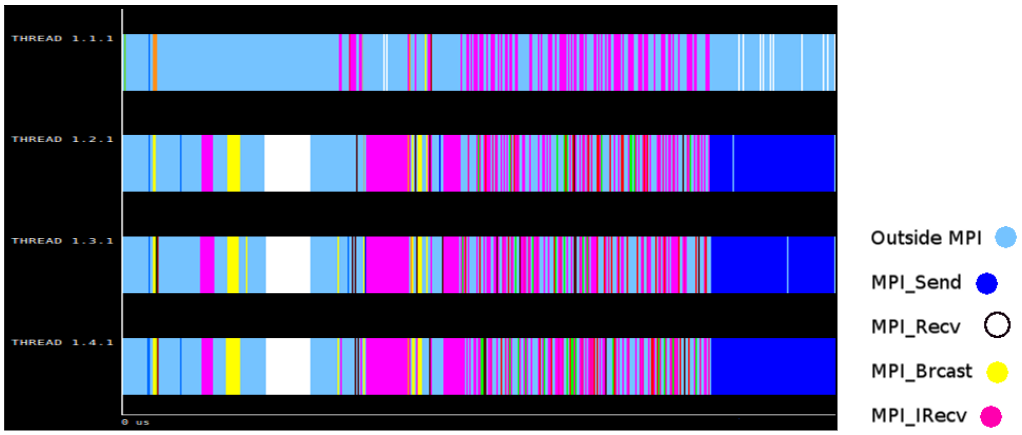


Figure 6.3.3: Execution trace of 4 MPI tasks visualized with PARAVER [32].

the calculations. After that we can see how the calculation of the solution is carried out: there is a constant feedback between the master and the slaves, which are requesting packages of information and performing calculations in an alternate fashion (very thin pink and fair blue stripes, denoting the change in the tasks). After the calculations have been made, the three slaves try to send the results to the master (wide dark blue stripes). When the master is ready to receive the information, it gets transferred. Afterwards, there are other processes outside MPI carried out by the master, such as producing the output files (which are not shown in figure 6.3.3).

Chapter 7

Conclusions

The main goal of this project was to create a software that would be able to solve the neutron transport equation using a deterministic approach. Furthermore, this software had to be integrated within the Alya system developed at Barcelona Supercomputing Center in order to provide the first step towards the development of multiphysics applications in the fusion domain.

Prior to developing the software itself, we required deeper understanding of the field and of what had been done before: several approaches had been tried in previous software and it was a necessary step to decide which way to go.

In particular, as it was mentioned in Chapter 3, we had to choose which discretization method over the directions had to be performed. We finally chose the Discrete Ordinates method, and experience has proven that it was the better choice: almost every deterministic code uses this discretization and it can be useful in a wide variety of cases.

The main goal mentioned above has been accomplished, that is, we have been able to create a functioning module integrated within the Alya framework with the following features: two-dimensional domains, stationary solutions, Discrete Ordinates method (S_8 and S_{10}), one-group approximation for the energy dependence, homogeneous materials and isotropic neutron sources.

In order to test the accuracy of the results, we performed a comparison between this module and the Zephyr code. We checked that only 2.53% of the nodes show a difference higher than 2%, which shows that the results agreed to a high extent: the differences were lower in the case of the neutron current (compared to those in the neutron flux) and only in a handful of nodes did we see a significant difference between the two software.

Afterwards, we decided to create a simple case from scratch just for the Alya module:

a two-dimensional thin slab of homogeneous material with an isotropic neutron source on one side. The results agreed with what we would expect, that is, there was an exponential decay of the flux and current with respect to the distance to the source. We extracted the data from the central nodes and fitted exponential trends to visualize them better.

As one of the strengths of the Alya system is its high scalability, we also carried out a performance analysis on the module. In order to do so, we created a much larger mesh than the ones of the examples we showed, so that the computational time of the sequential version would become much greater. This way we could parallelize the work a great deal and the different processes would still have a large quantity of operations to carry out. We saw how the module had very good scalability and efficiency if the amount of calculations was big enough for the number of nodes. In our test, for 64 nodes the efficiency was still around 60%.

Chapter 8

Further work

The software we have presented in this work is still under the first stages of development. Consequently, several capabilities and features need to be progressively added in order to increase the number and complexity of cases to be solved.

First of all, multigroup formulation for the energy discretization must be implemented. In fact, the module is already prepared to iterate over the number of energy groups, but the implementation of the multigroup approximation cannot only be solved with this loop. For instance, as we increase the number of energy groups we also need to increase the number of parameters, as the total and scattering cross section might differ from one group to another. Furthermore, the source term must also be split into energy groups so that different neutron fluxes of every energy can be introduced into the system.

So far, we have only modeled two-dimensional domains. It is an obvious next step to extend the capability of modeling three-dimensional domains in order to represent more realistic cases.

Anisotropic scattering also needs to be modeled: isotropic scattering in the lab reference frame can only be accurate in very restricted cases and therefore this implementation should be one of the firsts next steps to be carried out.

For the moment, we can only specify geometries with isotropic properties, that is, we can only model domains constituted by one material: being able to define different materials in different regions of the domain would also increase the possibilities of solving more complex cases.

Regarding boundary conditions, reflective boundary conditions must be implemented, as they are widely used together with vacuum boundary conditions, which we have already implemented. Another aspect that should be improved is the fact that the neutron

source is coupled with the boundary conditions in the sense that if we want to apply a certain boundary condition we also need to specify the source at that point. We must decouple those two terms in order to gain flexibility when defining the problem.

These improvements that we have mentioned so far require a new way of introducing the parameters of the problem: as we increase the complexity of the problem, the number of parameters increases accordingly, and therefore the input file structure that we have used in the first version of the software will render obsolete.

Related to that, when applying the multigroup approach with anisotropic scattering there is the need of a tool that takes the raw data from the ENDF files and transforms it into the data used by the software itself: we shall require this tool and therefore we need to consider either programming it ourselves or acquiring one of the current tools already developed.

The process of creating a new geometry is quite tedious and time consuming, even for the simple geometries we have been modeling so far. As we want the software to model realistic geometries (with intricate shapes and borders, irregular geometries and multiple materials) the process of designing the geometry or transforming a CAD model into a mesh that can be read by the software needs to be simplified and optimized.

At the same time that we implement this upgrades we need to carry out the benchmarking of the code: at the moment the software is too simple to be able to perform the standard benchmark cases, but as we improve it we must validate each upgrade.

Finally, once we have been able to implement all of the above, it would be time to couple this module with other capabilities of Alya, such as the fluid dynamics module or the thermal flows module. This way we would have the multiphysics application that we aim to.

Bibliography

[1] John R. Lamarsh and Anthony J. Baratta. *Introduction to Nuclear Engineering*. Prentice Hall, 3rd edition, 2001.

[2] James J. Duderstadt and William R. Martin. *Transport Theory*. John Wiley & Sons, 1st edition, 1979.

[3] James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis*. John Wiley & Sons, 1st edition, 1976.

[4] Rudi J.J. Stamm'ler and Máximo J. Abbate. *Methods of Steady State Reactor Physics Nuclear Design*. Academic Press, 1983.

[5] XI ITER Neutronics Meeting in Karlsruhe. https://www.inr.kit.edu/733.php. Accessed: 14-06-2016.

[6] Attila Radiation Transport Toftware. http://www.transpireinc.com/html/attila/. Accessed: 06-07-2016.

[7] Thomas M. Evans, Alissa S. Stafford, Rachel N. Slaybaugh, and Kevin T. Clarno. Denovo: A new three-dimensional parallel discrete ordinates code in scale. *Nuclear Technology*, 171(2):171–200, 2010.

[8] Matías Ávila. *Nonlinear subgrid finite element models for low Mach number flows coupled with radiative heat transfer*. PhD thesis, Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports de Barcelona, Universitat Politècnica de Catalunya, September 2012.

[9] Germán Theler. *MILONGA: a free nuclear reactor core analysis code*. TECNA S.A., Instituto Balseiro, July 2011.

[10] GiD: the personal pre and post processor. http://www.gidhome.com/. Accessed: 22-07-2016.

[11] ParaView software description. http://www.paraview.org/. Accessed: 25-07-2016.

[12] Jeffrey P. Freidberg. *Plasma Physics and Fusion Energy*. Cambridge University Press, 1st edition, 2015.

[13] Department of Economic United Nations and Population Division (2015) Social Affairs. World population prospects: The 2015 revision. https://esa.un.org/unpd/wpp/. Accessed: 13-06-2016.

[14] *Key World Energy Statistics*. International Energy Agency, 2015.

[15] Dani Gallart Escolà. Computational analysis of ion cyclotron resonance frequency heating for demo. Master's thesis, Department of Nuclear Engineering, Universitat Politècnica de Catalunya, June 2015.

[16] Francis F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer, 3rd edition, 2016.

[17] Xavier Sáez Pous. *Particle-in-Cell Algorithms for Plasma Simulations on Heterogeneous Architectures*. PhD thesis, Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona, Spain, 2016.

[18] ITER machine description. http://www.iter.org/mach. Accessed: 09-06-2016.

[19] Wendelstein-7 machine description. http://www.ipp.mpg.de/16900/w7x. Accessed: 09-06-2016.

[20] FDS Team Y. Wu. Cad-based interface programs for fusion neutron transport simulation. *Fusion Engineering and Design*, 84:1987–1992, 2009.

[21] Germán Theler. Difusión de neutrones en mallas no estructuradas: comparación entre volúmenes y elementos finitos. Facultad de Ingeniería, Universidad Nacional de Buenos Aires, December 2013.

[22] William Russell Martin. *The Application of the Finite Element Method to the Neutron Transport Equation*. PhD thesis, University of Michigan, 1976.

[23] Doppler Broadening description. http://www.nuclear-power.net/glossary/doppler-broadening/. Accessed: 07-07-2016.

[24] M. Herman and A. Trkov. *ENDF-6 Formats Manual*. National Nuclear Data Center, Brookhaven National Laboratory, Upton, NY 11973-5000, June 2009.

[25] R.E. MacFarlane, D.W. Muir, R.M. Boicourt, and A.C. Kahler. *The NJOY Nuclear Data Processing System, Version 2012*. Theoretical Division, Los Alamos National Laboratory, February 2015.

[26] ENDF, NJOY, and Applications. http://t2.lanl.gov/nis/njoy/njoy01.html. Accessed: 07-07-2016.

[27] Alya system user's documentation. http://bsccase02.bsc.es/alya/. Accessed: 15-06-2016.

[28] Dmitri Kuzmin and Jari Hämäläinen. *Finite Element Methods for Computational Fluid Dynamics. A practical guide.* SIAM. Computational Science & Engineering, 1st edition, 2007.

[29] Youcef Saad and Martin H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[30] Mariano Vázquez, Guillaume Houzeaux, Seid Koric, Antoni Artigues, Jazmin Aguado-Sierra, Ruth Arís, Daniel Mira, Hadrien Calmet, Fernando Cucchietti, Herbert Owen, Ahmed Taha, Evan Dering Burness, José María Cela, and Mateo Valero. Alya: Multiphysics engineering simulation toward exascale. *Journal of Computational Science*, 14:15–27, 2016. The Route to Exascale: Novel Mathematical Methods, Scalable Algorithms and Computational Science Skills.

[31] MPI documents. https://www.mpi-forum.org/docs/docs.html. Accessed: 04-08-2016.

[32] Paraver: a flexible performance analysis tool. http://www.bsc.es/computer-sciences/performance-tools/paraver/general-overview. Accessed: 01-08-2016.