

POLITECNICO DI TORINO

Collegio di Ingegneria Energetica

**Master of Science course  
in Energy and Nuclear Engineering**

Master of Science Thesis

# **Comparison of the FRENETIC Code and Serpent-OpenFOAM for the Multiphysics Analysis of Lead-cooled Fast Reactors**



## **Tutors**

prof. Sandra Dulla  
prof. Antonio Cammi  
prof. Piero Ravetto

**Candidate: Ettore Guadagni**  
**Matricola: 224152**

July 2018



# Abstract

Generation IV reactors represent the current frontier of nuclear fission technology development. These reactors are aimed at achieving passive safety features, together with the closure of the fuel cycle. The plant design of ALFRED, i.e. the demonstrator for one of the proposed GEN-IV design, the lead-cooled fast reactor, is being carried out by the Italian Ansaldo Nucleare. The research institution ENEA, in cooperation with Italian universities, is instead taking care of the core design. ALFRED (Advanced Lead-cooled Fast Reactor European Demonstrator) is a 300 MWth, MOX-fueled fast reactor, which has been devised in the framework of the European LEADER project and is currently undergoing the design and preliminary safety assessment phase.

The core design of a fission reactor is a complex procedure which must take into account the intrinsically multi-physics nature of the system. The latter is indeed characterized by a strong interplay between neutronics (NE), thermal-hydraulics (TH) and thermo-mechanics (TM). The temperature field in both fuel and coolant is obviously influenced by the fission power generation, which is a consequence of the neutron distribution in the core. In turn, the temperature field determines the thermo-mechanical stresses in the structures and should therefore be carefully assessed.

To address this problem, a coupled approach is proposed in this work. The full-core NE problem is solved with detail up to the pin level in order to evaluate the fission power distribution, which constitutes the heat source for a detailed CFD model of a chosen FA in the core. The thermal feedback for the selected FA is taken into account based on the evaluated temperature distribution. To provide an adequate evaluation of the temperatures in the rest of the core, for the purpose of evaluating feedback on cross sections, the full-core FRENETIC NE-TH code, recently developed at Politecnico di Torino, is employed. The latter is able to solve the full-core NE-TH problem thanks to simplifying assumptions and averaging procedures. The activities of this thesis will eventually support a benchmark between the developed, detailed tool and the FRENETIC code.



# Table of contents

<b>1. Introduction</b> .....	7
1.1 Energy generation and its issues.....	7
1.2 Generation IV nuclear reactors.....	8
1.3 Aim of the work.....	10
<b>2. Simulation strategy</b> .....	13
2.1 NE-TH coupling in fission reactors.....	13
2.2 Operative simulation procedure.....	14
<b>3. The Serpent Monte Carlo code</b> .....	17
3.1 Introduction.....	17
3.2 Basics of the Monte Carlo method.....	17
3.3 The Monte Carlo method applied to neutron transport.....	19
3.3.1 Serpent simulation procedure.....	19
3.3.2 Collecting results.....	23
3.4 NE Simulation setup.....	24
3.4.1 Approximation of the core temperature distribution.....	28
<b>4. OpenFOAM</b> .....	31
4.1 Introduction.....	31
4.2 CFD solution procedure.....	32
4.3 Simulation details.....	34
3.3.1 Mesh generation and grid independence study.....	34
4.3.2 Simulation setup.....	42
<b>5. The FRENETIC multi-physics code</b> .....	47
5.1 Introduction.....	47
<b>6. Simulation results</b> .....	51
6.1 NE preliminary simulations and benchmark.....	51
6.2 OpenFOAM preliminary simulations results.....	56
<b>7. Conclusions</b> .....	62
<b>Acknowledgements</b> .....	63
<b>References</b> .....	65



# 1. Introduction

## 1.1 Energy generation and its issues

In a world that is getting more and more conscious about its energy demands and their consequences over the environment, the main challenge that my generation has to face is a word that we now hear almost every day, especially when talking about technology: sustainability. While it is true that aiming to sustainability is a very general objective, applicable to almost every aspect of our lives, this becomes particularly important in the field of energy production. Global warming is real – even if someone would rather keep pretending it is not – and if we look at the main sources of CO<sub>2</sub> on the planet it is easy to see that electricity production and heat generation are by far the biggest slice in the pie, accounting for roughly 35% of global CO<sub>2</sub> emissions [1].

This is the reason why the last twenty years have seen a dramatic rise in the renewable energy business [2]; wind, solar and hydroelectric are all emission-free and (apparently) waste-free resources, which grants them a great appeal to the public.

However, environmentalists are now starting to realize that renewables may not be the *panacea* they thought them to be. Problems such as the low energy density and even more importantly the low availability of these sources have shown us that if we really want to efficiently transition to near-zero emissions, covering the Sahara desert with solar panels is not the answer.

Luckily enough, another energy source is now gathering renewed attention, and seems to be an excellent candidate for the solution of the emission problem. Despite its poor reception by the public, nuclear fission power is a zero-emission, low-waste, abundant resource, and the high availability of nuclear plants makes it a great substitute for traditional fossil fuels.

As of now, the latest advancements in nuclear technology form what's called Generation IV (or Gen IV for short).

## **1.2 Generation IV nuclear reactors**

In 2001, the US Department of Energy initiated the Generation IV Forum (GIF), an extensive R&D collaboration program between 14 nations to develop and start deploying (between 2020 and 2030) the next generation of nuclear reactors.

After an initial review of about one hundred different concepts, the GIF has focused its attention on six main reactor projects [3]. The selected systems are:

- **Gas-cooled Fast Reactor (GFR):** a natural uranium and plutonium-fueled, helium-cooled, fast breeding reactor. It will operate around 850 °C, with a closed fuel cycle.
- **Molten Salt Fast Reactor (MSFR):** the key feature of this reactor is to have the fuel (uranium fluoride) dissolved in the coolant and circulating through the core, inside a breeding blanket. The small fissile inventory and the inherent safety granted by the large negative temperature and void coefficients are very promising.
- **Supercritical Water-cooled Reactor (SCWR):** this high-pressure reactor operates above the thermodynamic critical point of water, meaning the coolant doesn't change phase and can be used to directly run the turbine, simplifying the plant considerably. Both a fast and a thermal, water-moderated option are currently being researched.
- **Very High-Temperature Gas Reactor (VHTR):** as the name suggest, this graphite-moderated, helium-cooled thermal reactor operates at very high temperatures, around 1000 °C.
- **Sodium-cooled Fast Reactor (SFR):** it's probably the main technology of interest in GIF, also because it relies on almost 400 reactor-years of previous

## 1. Introduction

---

experience, with SFRs operating in eight countries. It uses MOX as fuel, sodium as coolant, and sometimes includes a U-238 breeding blanket.

- Lead-cooled Fast Reactor (LFR): this last design is similar in a way to the SFR, but uses either lead or a lead-bismuth eutectic as coolant. Its main advantage is the passive safety granted by the natural circulation of lead in the core.

The latter reactor will provide the framework for the present thesis work. The LFR embodies the main principles of Gen IV reactors, aiming at improved levels of sustainability, safety, and economic feasibility with respect to traditional reactors.

The sustainability of the system is ensured by the closed fuel cycle and breeding capabilities of the fast reactor. After being consumed inside the core, fuel is reprocessed and the short-lived fission products are removed; then the fuel is re-introduced in the reactor, with an addition of natural or depleted uranium. This drastically increase the fuel utilization factor;. Moreover, in this case waste is entirely made up of short-lived fission products, thus making it easier to store and dispose of.

While the high sustainability due to the closed fuel cycle is a feature of all fast reactors, the true strength of the LFR lies in its inherent safety. Liquid lead doesn't require pressurization of the core, it doesn't produce hydrogen or other explosive gases, and it does not react exothermically in case of contact with water or air – as is the case for sodium. Moreover, the decay heat removal is ensured by natural circulation. This means that, should the coolant pumps stop working in case of an accident, heat would still be removed from the core.

These passive safety features also reflect heavily on economics. The inert nature of lead and its high boiling point allow for a great simplification of the plant design, eliminating the need for intermediate systems between primary and secondary coolant and pressurizers for the core which are instead requires for sodium-cooled reactors. Moreover, the pool design and the low pressure practically eliminate the possibility of a Loss Of Coolant Accident (LOCA), again simplifying the safety provisions with a positive effect on the cost of the plant.

All these characteristics make the LFR a very promising design for Gen IV, which is why the EU strongly invested in R&D for this type of reactor. The framework in which this thesis falls is the LEADER project (Lead-cooled European Advanced DEmonstration Reactor) [4], a joint European project coordinated by Ansaldo Nucleare aiming to design an industrial size plant (ELFR) and produce a concept for a scaled demonstrator, i.e. a smaller, less powerful reactor. The present work refers specifically to the latter, the Advanced Lead-cooled Fast Reactor European Demonstrator (ALFRED).

## 1.3 Aim of the work

Experimental investigation of various subsystems of nuclear reactors is of paramount importance for supporting design activities and safety studies. However, alongside the experimental efforts, reliable numerical simulations are also required to effectively support the aforementioned activities, dramatically reducing the time required to achieve relevant results for the real system.

Simulating the behavior of a nuclear fission reactor is a remarkable challenge. Indeed, this system is characterized by a tight interaction between different physical processes, extremely large differences in time scales and a significant spatial heterogeneity. In particular, the most important phenomena occurring in the reactor may be grouped under the following “branches” of physics:

1. Thermo-mechanics (TM)
2. Thermal-hydraulics (TH)
3. Neutronics (NE)

In brief, NE studies are required to estimate the fission power distribution in the reactor. TH evaluates the thermal response of the system to that heat source, taking into account the effect of the coolant flow and of thermal conduction in the structures. TM, finally, evaluates the effects of the temperature field on the structures in terms of stresses and deformations. These processes are clearly coupled, since (only

mentioning two effects) the temperature distribution influences the macroscopic cross sections which determine the neutron distribution in the reactor, and deformations in turn influence both TH and NE. Moreover, these phenomena occur on largely different space and time scales. This is evident if, for example, we consider that the average lifetime of a neutron in a fast fission reactor is in the order of  $10^{-7}$ - $10^{-6}$  seconds, while the typical time scales for heat diffusion inside a fuel pin are in the order of  $10^0$ - $10^1$  seconds [5]. The fission reactor is therefore a clear example of a *multi-physics, multi-scale* system.

In the present thesis, the focus is on coupled NE and TH analysis. The goal is to produce a numerical tool for the detailed, coupled NE-TH 3D simulation of the core of an LFR, with specific reference to the ALFRED design. It is clear that such a detailed tool will not provide results which are useful in the design phase, where fast-running, lumped-parameters or reduced dimensionality tools are more suitable. It is rather intended for providing detailed information concerning a chosen design.

The NE module simulates the whole core of the reactor, by means of a Monte-Carlo simulation considering the detailed reactor geometry and material composition. At the end, the power distribution relative to a single Fuel Assembly (FA) is retrieved and serves as input for the TH module. Due to computational reasons, a full 3D CFD simulation of the whole core would not be feasible. Therefore, the TH module only simulates a single FA.

The results of the CFD analysis are then compared with the FRENETIC code, a computational tool developed at Politecnico di Torino capable of performing coupled full-core NE and TH analysis for fast liquid metal-cooled reactors while maintaining a relatively low computational cost thanks to suitable approximations. This comparison provides a numerical benchmark for the FRENETIC code, which can be useful for assessing the validity of the adopted approximations.

The final step of the work involves the effective coupling between the two modules. With an iterative procedure, the NE and the TH module are ran alternatively until

### *1.3 Aim of the work*

---

convergence is reached. Results are then again compared with a full coupled run of the FRENETIC code.

The next chapters are organized as follows. The simulation strategy is discussed in chapter 2. The two modules, NE and TH, are described respectively in chapters 3 and 4; chapter 5 focuses on the FRENETIC code. Finally, results are presented in chapter 6 and conclusions are summarized in chapter 7.

## 2. Simulation strategy

### 2.1 NE-TH coupling in fission reactors

As briefly discussed in the previous section, neutronics and thermal-hydraulics are strongly coupled in a fission reactor. It is of capital importance to correctly understand the mechanisms through which these two phenomena influence one another, to accurately reproduce the reactor behavior in a numerical simulation.

It is quite easy to see why the neutron flux distribution has a strong influence on the thermal-hydraulics of the reactor. The neutron distribution inside the core determines the fission distribution – intuitively, a zone with a higher neutron density will have a higher fission density – and consequently the volumetric power distribution, which constitutes a source term in the heat transport equations. In fact, in a nuclear engineering framework, to talk about neutron flux or volumetric power distribution is essentially the same thing, as one can easily retrieve the former from the latter through a simple proportionality factor, obtained by multiplying the energy production per fission by the macroscopic fission cross section.

The effect of the coolant and fuel temperature distribution on the neutronics is less obvious and requires a deeper understanding of reactor physics. In fact, temperature effects are related to the concept of cross sections.

A cross section, as will be better explained in Section 3.3, is a measure of the probability a neutron has of undergoing a certain reaction with a nucleus of the medium in which it is traveling. Cross sections depend on many factors, including the neutron energy, the medium in which the neutron is traveling, and, most importantly for the case in exam, its temperature. In particular, the temperature of the medium has both a direct effect on the cross section (a temperature increase corresponds to a decrease in the cross section) and an indirect one, which is even more important, due to a phenomenon called *Doppler broadening*.

Without going too much in detail, it is sufficient to know that the Doppler broadening is related to the relative velocity between an incoming particle (the neutron) and its target (the nucleus). In fact, although the neutrons are moving much faster, the nuclei in a reactor are also in constant motion due to their thermal energy. Therefore, the temperature of the medium (fuel, coolant or structural material) increases the kinetic energy of its nuclei and modifies the value of its cross section through the effect of Doppler broadening.

In reactor physics, these temperature effects are accounted for by appropriate coefficients called *feedback coefficients*, which directly link the change in fuel or coolant temperature to a change in fission rate. In particular, it is usually of crucial importance for a reactor to have negative temperature coefficients – i.e. to have a reduction of fission rate corresponding to a temperature increase – in order for the reactor to be stable.

The foregoing considerations explain the tight coupling between NE and TH in fission reactors. In order to reliably simulate such a system, a numerical code should foresee coupled NE-TH evaluations. As it is the case for the vast majority of multi-physics codes, an efficient data exchange between the NE and the TH modules of a numerical simulation is crucial. In the next section, the implementation of this information exchange between the codes used in this work is presented.

## 2.2 Operative simulation procedure

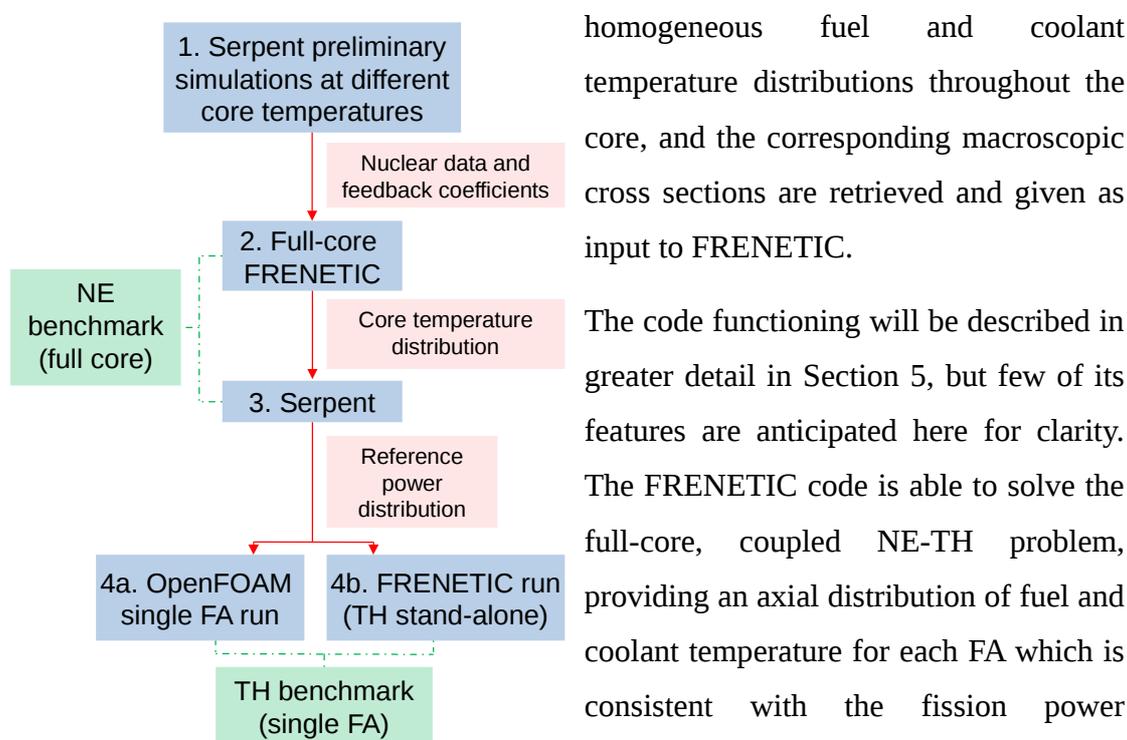
Due to the strong coupling which has just been discussed, a problem arises: it is impossible to have a reliable neutronic simulation without a consistent temperature distribution, which in turn would need a consistent fuel power distribution to be obtained.

One of the classical ways to solve this problem is to just assume a starting condition and then iterate between the NE and TH analysis until convergence is reached. However, this is not feasible to the level of detail this work aims at achieving – i.e. 3D NE simulation with detail up to the pin level and 3D CFD simulation of fuel

## 2. Simulation strategy

assemblies – at least for the computational resources made available for the present work. Actually, nowadays this type of simulation is becoming affordable, thanks to advanced parallelization strategies and large clusters. However, a more elegant approach is suggested in the present thesis. Since the CFD thermal-hydraulics code only simulates a single FA and not the full core as would be required by the NE module, the temperature distribution for the rest of the core is provided by the FRENETIC code. Notwithstanding the lower level of detail, indeed, the latter is capable of providing a self-consistent, full-core temperature distribution, which can be used by Serpent for evaluating the Doppler broadening of cross sections. In this way, as will be explained in the rest of this section, FRENETIC provides a framework for the Monte Carlo (NE) – CFD (TH) coupling.

As shown in **Figure 2.1**, before running the aforementioned FRENETIC simulation which will provide the full-core temperature distribution for Serpent, the nuclear data must be generated. In particular, to properly account for feedback effects, at least three preliminary Serpent simulations are required. These are conducted by assuming



**Figure 2.1.** Simulation flowchart

homogeneous fuel and coolant temperature distributions throughout the core, and the corresponding macroscopic cross sections are retrieved and given as input to FRENETIC.

The code functioning will be described in greater detail in Section 5, but few of its features are anticipated here for clarity. The FRENETIC code is able to solve the full-core, coupled NE-TH problem, providing an axial distribution of fuel and coolant temperature for each FA which is consistent with the fission power generation, evaluated by means of a coarse mesh 3D diffusion approach.

## 2.2 Operative simulation procedure

---

Therefore, the code provides a *single* value of fuel temperature and coolant temperature for each assembly. The spatial detail of such a calculation is much lower than the one provided by the CFD approach, but it is sufficient to provide a reliable estimate of the temperature distribution in the core.

This information on the temperature distribution is then passed to the NE code, which is run again (point 3 of the flowchart in Figure 2.1), this time to retrieve an accurate full-core power distribution. At this point, a first benchmark of the NE module of FRENETIC against the highly reliable Monte-Carlo simulation performed by Serpent can be conducted (first green box).

Finally, the power distribution evaluated by Serpent is provided as input to both the OpenFOAM CFD solver and the TH module of FRENETIC, to perform the second and last part of the benchmark (lower green box). In this way, a fair and representative comparison between the two NE codes and the two TH codes, respectively, can be achieved.

## **3. The Serpent Monte Carlo code**

### **3.1 Introduction**

Serpent is a 3D continuous energy Monte Carlo particle transport code developed at VTT Technical Research Centre of Finland [6]. It can perform a wide range of traditional reactor physics calculations, such as spatial homogenization, criticality calculations, fuel cycle studies, as well as coupled multi-physics calculations, when paired with a thermal-hydraulics or CFD code, as in the present work.

Serpent development started in 2004, and since the first release in 2007 as part of a doctoral thesis [7] the code has been gradually updated and perfected. During the years, also thanks to its simple and intuitive functioning, it has been extensively used in the neutronic community and is included in countless papers and dissertations. The version used in the present work is the 2.1.29 release.

The main idea behind the code and the Monte Carlo methodology in general, is to not solve the integro-differential transport equation nor the simplified diffusion equation. Instead, Serpent aims at directly reproducing the physics of the problem by simulating a large number of particles (the neutrons) and their interactions with the materials composing the reactor.

To better understand the code functioning, a brief introduction to the Monte Carlo method is provided in the next section.

### **3.2 Basics of the Monte Carlo method**

The Monte Carlo method is a numerical statistical method used to calculate averages of stochastic systems. It was first developed in the 1940s and has become increasingly popular together with the improvements of computers and their computational power. A very trivial and intuitive way to describe the MC approach is the following: if one

had to guess the probability of rolling two six-sided dice and obtaining two ones, the MC way of doing that would be to make a computer virtually roll the two dice a sufficient number of times, and then to calculate the ratio of “successful throws” over total throws. Essentially, starting by simple known probabilities – in this case, the 1/6 probability of rolling one on a die – the MC method allows to numerically evaluate arbitrarily complex stochastic processes. A brief explanation of the underlying mathematical principles is now offered [8].

Given a stochastic event, which can result in a certain (finite or infinite) number of outcomes, we can associate to each of these outcomes a numerical value; this value will be called *random variable*. For example, for a six-sided die, it is trivial to see the number of outcomes for the stochastic event of “rolling the die” is six; therefore, the associated random variable  $x$  will have six possible discrete values.

We can also define a probability distribution  $f(x)$  in such a way that the probability for the random variable to take a value between  $x$  and  $x+dx$  is:

$$dP=f(x)dx \quad 3.1$$

This means that the probability for  $x$  to take a value between  $a$  and  $b$  – or, which is the same, the probability for the event to have an outcome between  $a$  and  $b$  – is given by:

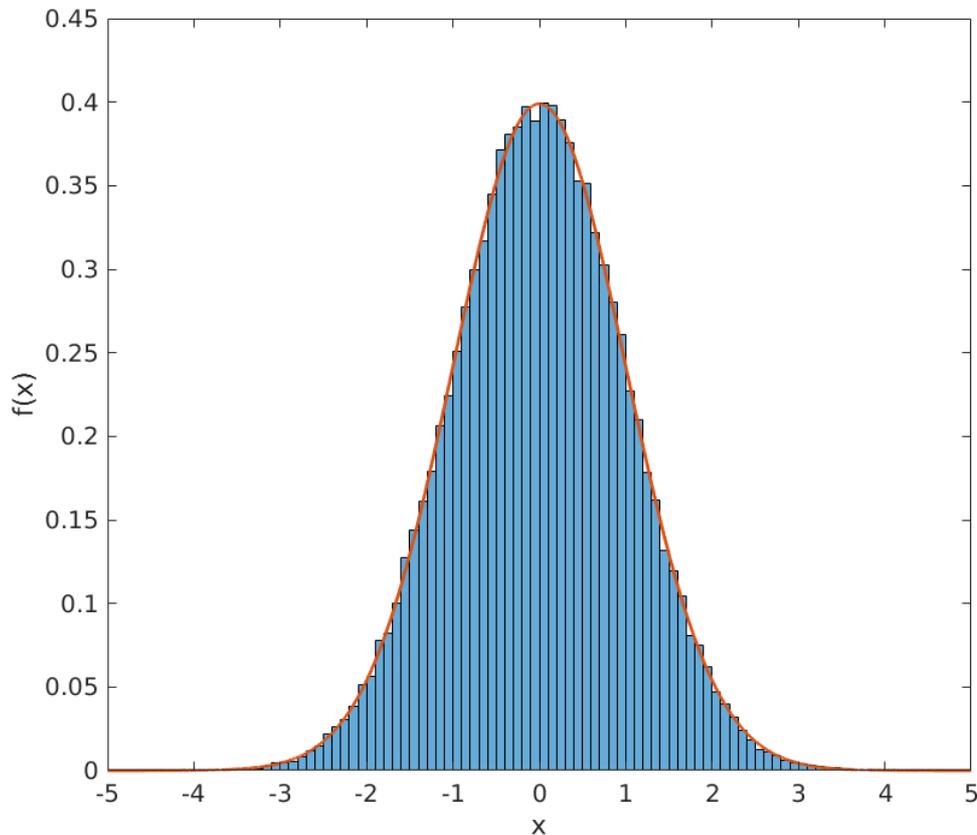
$$P(a<x<b)=\int_a^b dP=\int_a^b f(x)dx \quad 3.2$$

The function  $f(x)$  is called *probability density function* or PDF. By definition of probability, the total probability of the space of events must always be equal to 1, therefore we obtain a normalization constraint for PDFs:

$$P(-\infty<x<\infty)=\int_{-\infty}^{\infty} f(x)dx=1 \quad 3.3$$

From what we said up to here, we can conclude that by knowing the PDF associated to a certain event, if we can sample random values according to that PDF we are, in fact, simulating the event itself. Luckily, there are a number of ways to effectively sample a random value according to a known probability distribution. For example,

**Figure 3.1** shows the distribution of 100,000 random numbers generated according to a Gaussian PDF.



**Figure 3.1.** Distribution of 100,000 random numbers sampled according to a Gaussian PDF with  $\sigma=1$  and  $\mu=0$ . The PDF is reported in red.

## 3.3 The Monte Carlo method applied to neutron transport

### 3.3.1 Serpent simulation procedure

It is now clear that, if we are able to decompose the very complex mechanism of neutron transport into a series of basic stochastic events, for which the associated PDFs are known, we should be able to accurately simulate the whole phenomenon. Therefore, let us take a closer look at what happens, at a microscopic level, to a neutron traveling in a medium [9].

### 3.3 The Monte Carlo method applied to neutron transport

---

A neutron is a heavy particle with no electric charge. Because of that, when it travels inside matter, it is not affected in its path by any kind of electromagnetic force, such as the one produced by electrons. In fact, for a neutron, electrons are basically invisible and the only interactions with the medium happen when the neutron travels sufficiently close to an atomic nucleus to feel the influence of strong nuclear force. Due to the very short range of the aforementioned force (about  $10^{-15}$  m, same order of magnitude as an atomic nucleus diameter), the path of a generic neutron in a medium is conceptually similar to that of a ball on a pool table: a series of straight segments that change their orientation only when another ball (nucleus) is hit.

Actually, the possible interactions between neutrons and atomic nuclei are more complicated than the impact between two balls. When a neutron hits a nucleus, it briefly enters it, creating a new compound nucleus in an excited energy state, that after a very short time collapses back to a stable energy level. The way this collapse happens defines the type of interaction. The possibilities are many, but they can be grouped in three main types: capture – the excited nucleus emits energy in the form of a photon or some other particle, no secondary neutrons are emitted and the incident neutron becomes part of the nucleus – fission – the excited nucleus divides itself in two smaller nuclei, emitting one or more secondary neutrons in the process – and scattering – the nucleus discharges the energy of the impacting neutron by emitting one or more secondary neutrons.

These two phenomena – namely, the path traveled by a neutron until an interaction happens, and the type of interaction following – can both be approached in a stochastic way, and described in terms of probability. The fundamental quantity required to do that is the cross section.

Consider a neutron of energy  $E$ , traveling in the medium  $M$  for a distance  $dx$ . The probability for the neutron to have an interaction of type  $j$  is:

$$dP_j = \Sigma_j(E, M) dx \quad 3.4$$

Where  $\Sigma_j$  is the cross section for the  $j$ -th interaction, and is a function of both the neutron energy and the atomic composition of the medium  $M$ . We can eliminate the  $M$

dependency by supposing that the material is spatially homogeneous – i.e. it doesn't change in  $dx$ ; moreover, by noting that the neutron energy is constant between one interaction and the next, we can also remove the  $E$  dependency. Finally, if we are not interested in the type of interaction, we can sum over all  $j$ 's to obtain the probability for the neutron to have an interaction of any kind in  $dx$ :

$$dP_I = \Sigma_t dx \quad 3.5$$

Where  $\Sigma_t = \sum_j \Sigma_j$  is called *total cross section*. Let us now define  $P_{NI}(x)$  as the *probability of non interaction*, i.e. the probability for the neutron to travel to point  $x$  without undergoing an interaction. When the neutron covers distance  $dx$ , the infinitesimal change in  $P_{NI}(x)$  is related to the probability of interaction defined earlier by:

$$dP_{NI} = -P_{NI}(x) dP_I = -P_{NI}(x) \Sigma_t dx \quad 3.6$$

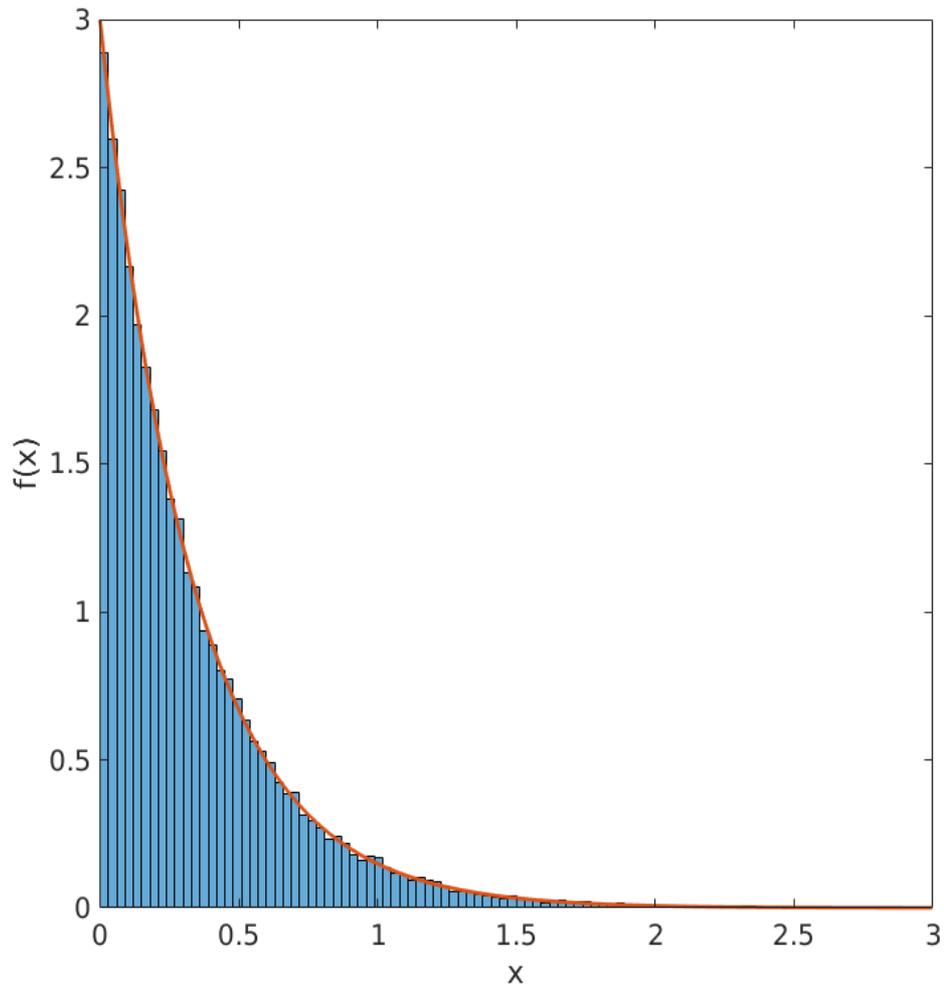
This is a simple differential equation, that we can solve, remembering that  $\Sigma_t$  is constant due to previous assumptions, to obtain a formula for  $P_{NI}$ :

$$P_{NI}(x) = e^{(-\Sigma_t x)} \quad 3.7$$

Finally, by multiplying the two previous quantities, we can obtain the probability for a neutron to travel to point  $x$  with no interaction and then to have an interaction between  $x$  and  $x+dx$ :

$$P(x) dx = P_{NI}(x) dP_I = \Sigma_t e^{(-\Sigma_t x)} dx \quad 3.8$$

By looking at Eq. 3.8 we can see that  $f(x) = \Sigma_t e^{(-\Sigma_t x)}$  is actually a PDF: therefore, if we sample random numbers according to it, we are actually able to correctly reproduce the stochastic process of a neutron traveling between one interaction and the next. **Figure 3.2** shows the distribution of 100,000 random numbers sampled based on said PDF, with  $\Sigma_t=3$ .



**Figure 3.2.** Distribution of 100,000 random numbers sampled according to the PDF of equation 3.8, with  $\Sigma_t=3$ . The PDF is reported in red.

This is the basic concept of how Serpent works. Every particle is described by a set of quantities, namely its position in space, its energy and its direction of motion. Based on atomic composition of the medium, the code evaluates the total cross section  $\Sigma_t$  of the material the particle is in, then it samples a path length according to Eq. 3.8. The particle is moved to the new position and with a series of random samplings the code selects the nucleus involved in the collision, the type of collision (capture, fission, or one of the many types of scattering) and the characteristics of any new particle born from the collision.

Whenever a particle crosses the simulation boundary or is captured, its history is terminated; when all particles histories reach their end, a new cycle starts, with a new source distribution based on the previous cycle fission reaction distribution. Finally, after a number of cycles sufficiently high to allow the statistical error, evaluated on suitable quantities, to decrease to an acceptable level, the simulation is stopped.

#### 3.3.2 Collecting results

After having outlined how the simulation works, it is now worth discussing the way Serpent collects results [10,11]. As said before, a MC code does not solve the transport equation, therefore it doesn't output a continuous function for the flux distribution; instead, the whole process is based on discrete quantities, and the way of obtaining information from the simulation is actually similar to experimental physical measurements.

Typically, quantities of interest in a neutronic simulation are expressed in the form of flux integrals:

$$R = \int_V \int_E f(\mathbf{r}, E) \Phi(\mathbf{r}, E) dE d^3 r \quad 3.9$$

Where  $\Phi(\mathbf{r}, E)$  is the neutron flux and  $f(\mathbf{r}, E)$  is the desired response function. Depending on the choice of the response function,  $R$  obviously takes different meanings. Usually, the response function is just a reaction cross section of some type, in which case the integration yields the corresponding reaction rate for that specific type of interaction.

Since in a MC simulation neutron interactions are reproduced one by one, there is a very simple way to calculate a quantity of this type, which is to actually count the interactions of the chosen type happening inside the domain of integration. This is called the *analog estimate* of the quantity.

For example, if the user is interested in evaluating the total fission reaction rate in the fuel pins, the analog estimate simply consists in counting all the fissions happening in the fuel pins. This method is straightforward, and to change the domain of integration

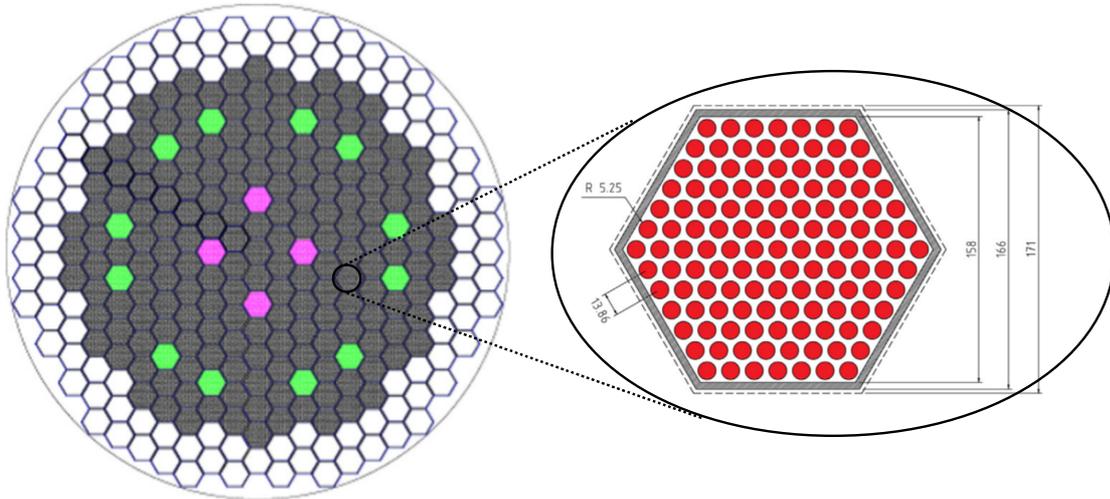
one simply needs to define a geometrical boundary for counting interactions, or an energy threshold. However, the analog estimate's efficiency becomes poor when dealing with small volumes located in areas with low collision density, simply because it becomes less and less likely to have a collision inside the domain of integration, and therefore the statistic sample is drastically reduced. That's why alongside the analog estimate Serpent uses an implicit method, the so-called *collision estimate* [12], to calculate the neutron flux and generally the reaction rates.

Implicit estimate methods are derived from the analog estimate, but instead of scoring events that actually occur during the simulation, the estimates are based on the expected occurrence of the events, evaluated through the use of fractional reaction probabilities. In short, the reason why the collision estimator is more efficient than the analog reaction rate estimate (i.e. counting the number of sampled interactions) is that the score is recorded even if the particular interaction does not occur.

## 3.4 NE Simulation setup

**Figure 3.3** shows a layout of the ALFRED core, with the detail of a FA; the core includes:

- 171 fuel assemblies (FAs, dark grey in the figure), with two differently enriched types of fuel (57 in the “inner zone”, with MOX Pu 21.7% enr., and 114 in the “outer” zone, with MOX Pu 27.8% enr.); each FA has 127 fuel pins.
- 12 control assemblies (green in the figure) for reactivity control during normal operations, each comprising a circular bundle of 19 control rods filled with Boron carbide. Control rods are inserted from the bottom of the core.
- 4 safety assemblies (pink in the figure) for emergency shutdown, always extracted during normal operations; they present a circular bundle of 12 safety rods in Boron carbide. Safety rods are inserted from the top of the core.
- 114 dummy assemblies (white in the figure), i.e. empty assemblies with the liquid lead running through them acting as a neutron reflector.



**Figure 3.3.** Detail of ALFRED core and FA composition

The material composition of the core was obtained from [13]. The detailed isotopic composition of the fuel [14] is reported in **Table 2.1**.

Uranium vector			Plutonium vector		
Isotope	at. %	wt. %	Isotope	at. %	wt. %
$^{234}\text{U}$	0.003	0.003	$^{238}\text{Pu}$	2.348	2.332
$^{235}\text{U}$	0.409	0.404	$^{239}\text{Pu}$	57.016	56.874
$^{236}\text{U}$	0.010	0.010	$^{240}\text{Pu}$	26.951	26.997
$^{238}\text{U}$	99.578	99.583	$^{241}\text{Pu}$	6.069	6.104
			$^{242}\text{Pu}$	7.616	7.693

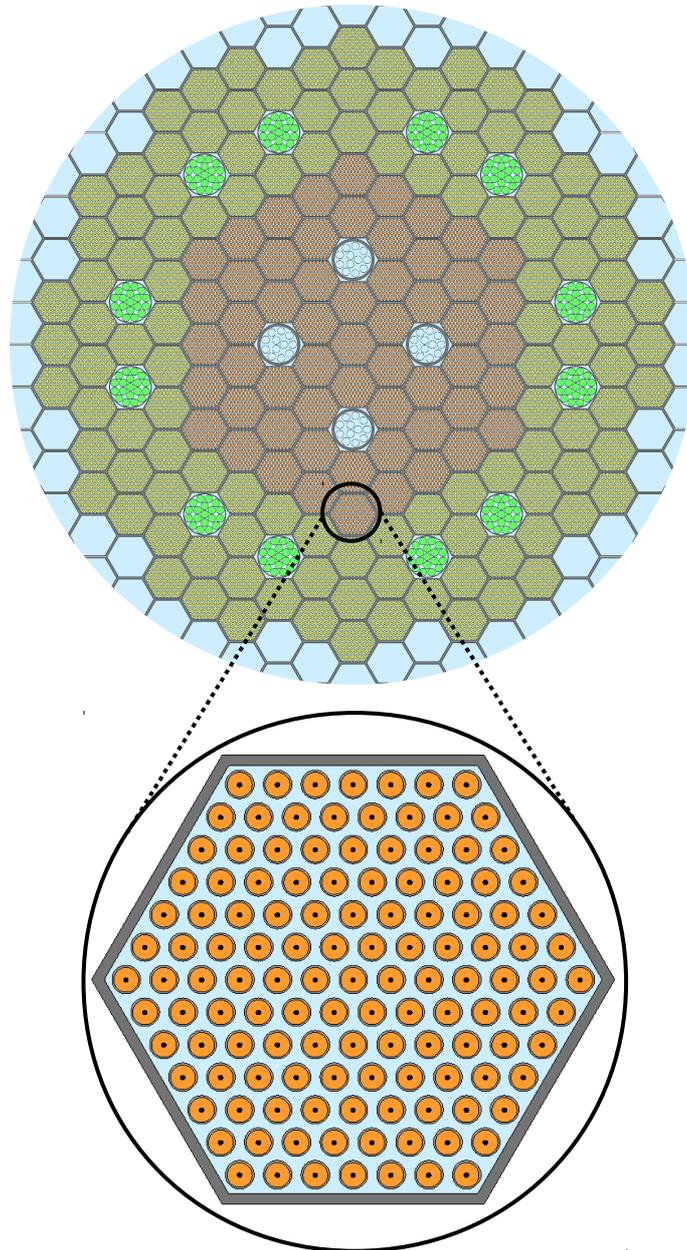
Inner fuel composition			Outer fuel composition		
Element	at. %	wt. %	Element	at. %	wt. %
Pu	21.422	21.534	Pu	27.443	27.576
Am	0.278	0.282	Am	0.357	0.361
U	78.300	78.184	U	72.200	72.063

**Table 3.1.** Fuel vectors and isotopic compositions for inner and outer fuel zones.

Since the desired output of a simulation for the purpose of this work is the power distribution inside the fuel pins, the core was reproduced with accuracy up to the single pin level. Serpent features a nested geometry pre-processor that allows to build

### 3.4 NE simulation setup

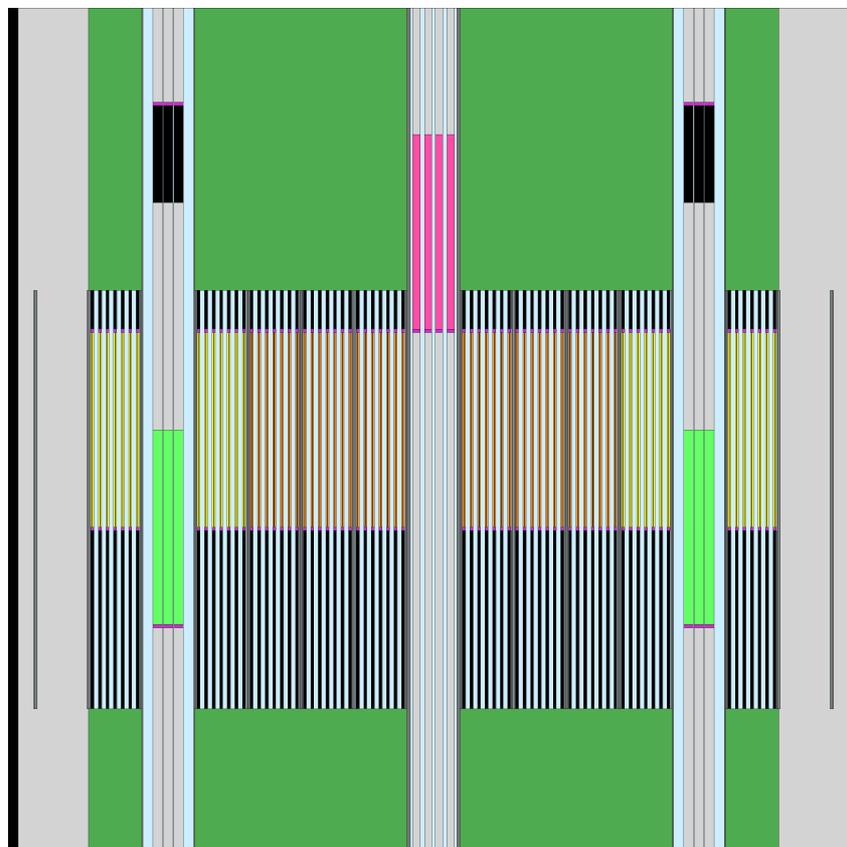
complex arrangements by starting from the basic components. In this case, first the single fuel pins (inner and outer) were modeled, alongside the control and safety rods. Then the four type of assemblies (inner FA, outer FA, CA and SA) were defined and arranged in the configuration shown in **Figures 3.4** (radial view) and **3.5** (axial view).



**Figure 3.4.** *x-y* section view of the full ALFRED core, as modeled in Serpent, with detail of a single FA

The simulation was ran with 1 million simulated particles and 2000 total cycles, to obtain an acceptably low statistical error on the power detectors. The statistical error

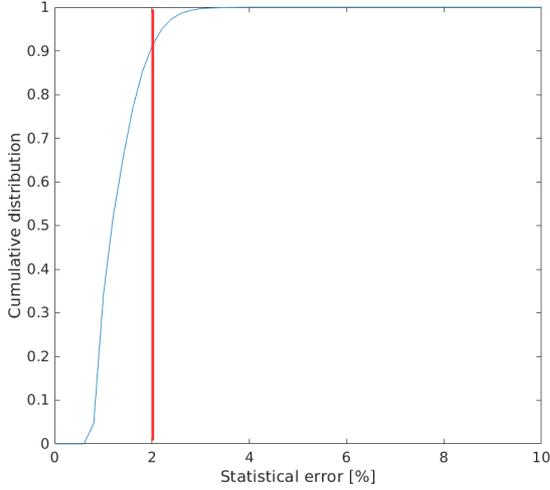
cumulative distribution on the detectors is shown in **Figure 3.6**. Notably, all the detectors have a statistical error below 4%, and more than 90% of the detectors are below 2% error. Moreover, 50 initial inactive cycles were added to allow the source distribution to converge before starting to collect data. These cycles are needed because, as previously explained, Serpent uses the fission distribution of one cycle as a source for the next cycle. The first cycle must use a random guess for the source distribution, so a number of initial cycles must be discarded in order to have physically meaningful results. The correct number of cycles is assessed by looking at both the convergence of  $k_{eff}$  and the Shannon entropy [15], as seen in **Figure 3.7**.



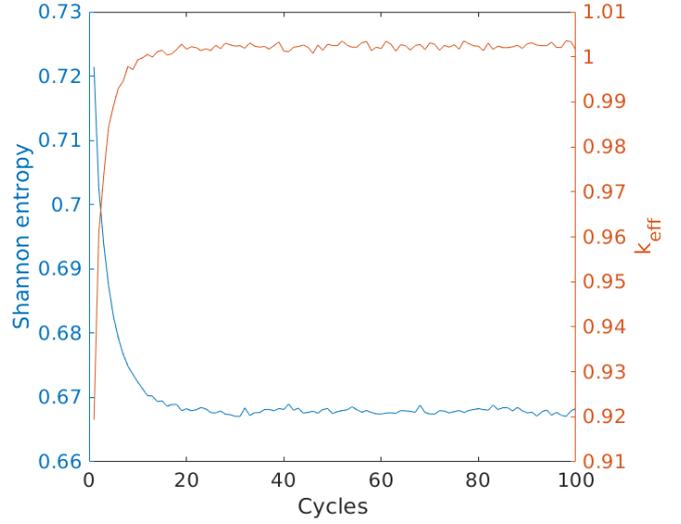
**Figure 3.5.** ALFRED core, y-z section view, in Serpent geometry

To retrieve the power distribution in the fuel (the quantity that serves as input for the TH code) axial detectors were placed in every fuel rod. Specifically, every fuel rod was divided in 100 axial zones, with no radial subdivision.

### 3.4 NE simulation setup



**Figure 3.6.** Cumulative distribution of statistical error on the detectors. The function  $f(x)$  indicates, for every  $x$ , the percentage of detectors with statistical error equal or less than  $x$ .



**Figure 3.7.** Plot showing the convergence of  $k_{eff}$  and Shannon entropy after 100 inactive cycles.

The detectors use the flux collision estimate to evaluate the total fission rate in the fuel, as explained in the previous chapter, thus calculating the integral of Eq. 3.9 with the following specifics:

$$V = \pi r_{pin}^2 \frac{h_{pin}}{100} \quad 3.10$$

$$f(\mathbf{r}, E) = \Sigma_{fission} \quad 3.11$$

The result is then normalized to obtain the discretized axial power distribution for every fuel pin in the core.

#### 3.4.1 Approximation of the core temperature distribution

As previously noted in section 2.1, the detailed CFD model which represents the TH module is limited to a single assembly to avoid excessively high computational burden. This means that the information from the TH analysis (notably, the fuel and coolant temperature and density distribution) is only available for the simulated assembly. Therefore, the problem of estimating the temperature distribution in the rest of the core arises.

### 3. The Serpent Monte Carlo code

---

The strategy proposed in this work to circumvent this difficulty takes advantage of the simulation capabilities of the FRENETIC code, which is able to provide the assembly-averaged temperatures of coolant and fuel for the whole core.

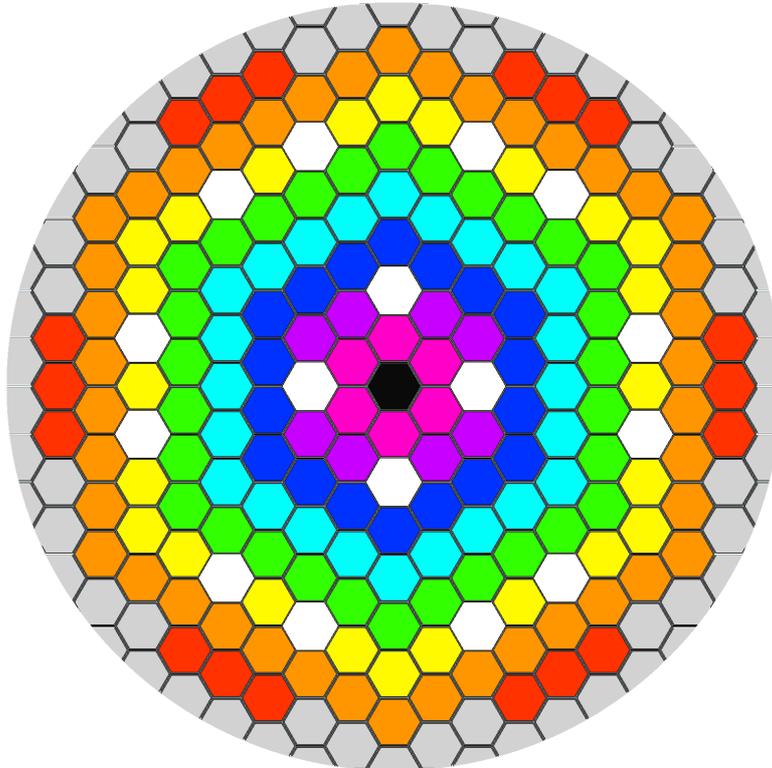
As a first approach, each assembly was divided axially in 10 zones of equal size. This required Serpent to evaluate cross sections for a number of sectors equal to:

$$N = N_{ass} \cdot N_{axial\ zones} \cdot 2 = 171 \cdot 10 \cdot 2 = 3420 \quad 3.12$$

Where the two is due to the fact that each zone included a fuel and a coolant temperature. The memory required for the calculation was excessive (over 100 GB), so another strategy was adopted. Taking advantage of the fact that the problem is almost symmetric, the assemblies were divided radially into nine rings with homogenized temperatures. This allowed to reduce the total number of sectors to:

$$N = N_{rings} \cdot N_{axial\ zones} \cdot 2 = 9 \cdot 10 \cdot 2 = 180 \quad 3.13$$

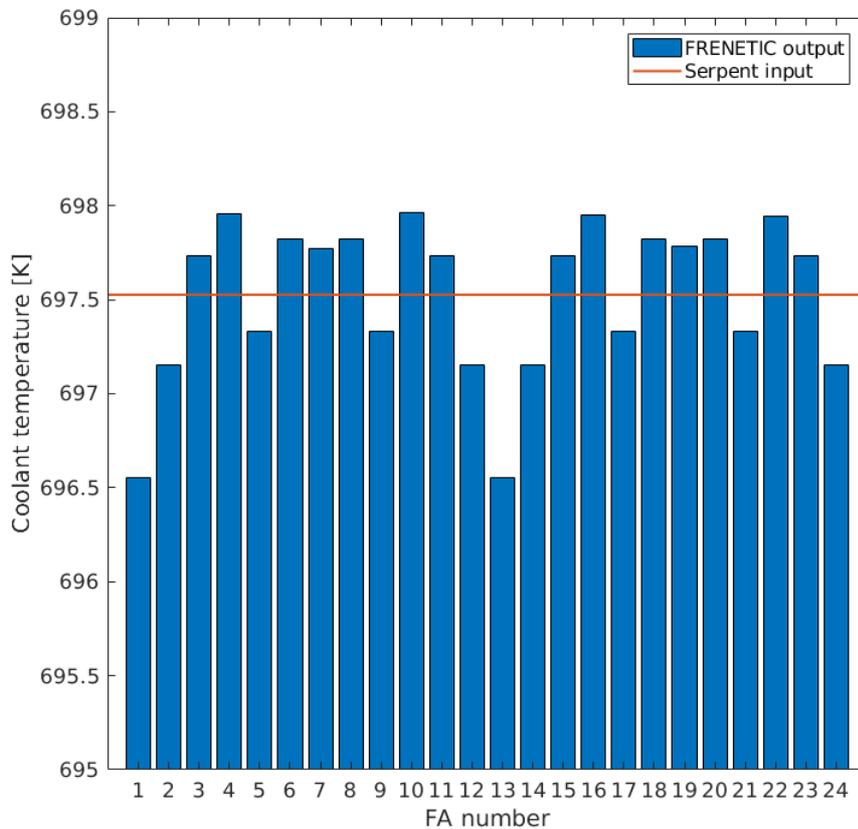
**Figure 3.8** shows the radial ring arrangement.



**Figure 3.8.** Radial temperature distribution implemented in Serpent. Each color represents a different temperature zone.

### 3.4 NE simulation setup

This radial homogenization is actually justified by the fact that, due to the shape of the reactor, temperature differences between assemblies belonging to the same radial ring are relatively small. This is well shown in **Figure 3.9**, which reports the coolant temperatures evaluated by FRENETIC at the middle of the active zone for the FAs in the fifth ring (the light blue one in Figure 3.8). It is easy to see that the maximum error is in the order of 1 K, well below 1%. This error is believed to have a small influence on the resulting power distribution.



**Figure 3.9.** Coolant temperatures at the middle of the active zone evaluated by FRENETIC for the 24 FAs of the fifth ring from the outside. FAs are numerated clockwise starting from the upper middle one in Figure 3.8. The red line shows the mean value implemented in Serpent.

## 4. OpenFOAM

### 4.1 Introduction

OpenFOAM (Open Field Operation And Manipulation) [16,17] is a free, open source, C++ based CFD software developed by OpenCFD Ltd. It was first released in 2004 and has since been regularly updated with contributions from both OpenCFD and OpenFOAM users around the globe, due to its open source nature.

The software offers a large choice of numerical solvers optimized to solve a plethora of problems in fields ranging from fluid flows to acoustics, solid mechanics, electromagnetics, and heat transfer problems. The main feature of the software is its intuitive and highly hierarchical global architecture that allows the users to develop their own solvers at high level avoiding the need to code the low-level routines which perform numerical discretizations and implement well-established models.

For the purpose of this work, the custom solver “*pwrThermoFoam*” developed by Paolo Bianchini [18] was used. It is based on a solver called “*chtMultiRegionFoam*”, contained in the official OpenFOAM release, the purpose of which is to evaluate conjugate heat transfer between materials in different states. The choice of this solver for the case in exam is straightforward, since heat is generated inside the solid fuel pins, it diffuses outwards and is removed by the coolant (liquid lead). *pwrThermoFoam* implements the iterative solution of the energy, momentum and pressure equations for the fluid – including appropriate treatment for turbulent phenomena – and of the heat diffusion equation for the solid. The solution of each equation is achieved by calling well-established low-level routines, letting the user control the accuracy of the solution and the specific methods adopted by adding appropriate keywords and values to relatively intuitive input files. This custom solver also contains some features to facilitate the coupling with a neutronic code. Indeed, *pwrThermoFoam* implements an externally provided volumetric power generation inside the solid regions (fuel pins). The volumetric power distribution is read before

solving the energy equation, for each outer iteration. A detailed explanation of the solution procedure is presented in the next chapter. Moreover, this custom solver adds some features to facilitate the application to nuclear reactor physics and the coupling with a neutronic code; namely, *pwrThermoFoam* accounts for volumetric power generation inside solid regions (fuel pins), by automatically reading the volumetric power field before solving the energy equation at each iteration. A detailed explanation of the solving procedure is presented in the next chapter.

## 4.2 CFD solution procedure

The solver *pwrThermoFoam* is specifically designed for nuclear application (as the name suggests: the PWR, or pressurized water reactor, is currently one of the most widely used fission reactors) and it solves the conjugate heat transfer problem in a solid medium, the fuel pin, and in a circulating adjacent liquid, the coolant.

While the solver is originally structured for time-dependent calculations, it also includes a steady-state version called *pwrThermoSimpleFoam*, which is the one used in the present work. As the name suggests, it implements the SIMPLE algorithm (Semi-Implicit Method of Pressure-Linked Equations) to iteratively solve the balance equations for the coolant. An extensive explanation of the solution process would involve a thorough review of the basics of computational fluid dynamics [19], which would be out of the scope of this work. Instead, in this section, a brief explanation of the SIMPLE solving procedure is offered.

The three conservation equations of mass, momentum and energy that need to be solved in order to fully determine the flow of a fluid are known as the Navier-Stokes equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{v}] = 0 \quad 4.1$$

$$\frac{\partial}{\partial t} [\rho \mathbf{v}] + \nabla \cdot [\rho \mathbf{v} \mathbf{v}] = \mathbf{f} \quad 4.2$$

$$\frac{\partial}{\partial t} (\rho \hat{u}) + \nabla \cdot [\rho \mathbf{v} \hat{u}] = -\nabla \cdot \mathbf{q}_s - p \nabla \cdot \mathbf{v} + (\boldsymbol{\tau} : \nabla \mathbf{v}) + q_v \quad 4.3$$

Where  $\rho$  is the fluid density,  $\mathbf{v}$  is its velocity,  $\mathbf{f}$  is the specific external force acting on the material volume,  $\hat{u}$  is the specific internal energy,  $\dot{q}_s$  is the specific rate of heat transfer across the surface area of the material volume,  $p$  is the pressure,  $\dot{q}_v$  is the volumetric heat source, and  $\boldsymbol{\tau}$  is the viscous stress tensor. These equations are at the basis of any fluid dynamic problem, and to be solved they need suitable closure relationships that vary in their formulation depending on the problem under consideration.

In particular, the main issue is the unavailability of a closure equation for the evaluation of the pressure field appearing in the momentum equation. Eq. 4.1 can in fact be rewritten by splitting the generic force vector  $\mathbf{f}$  into its components  $\mathbf{f}_s$  (surface forces) and  $\mathbf{f}_b$  (body force), and after some manipulation, the presence of the pressure becomes explicit:

$$\frac{\partial}{\partial t}[\rho \mathbf{v}] + \nabla \cdot [\rho \mathbf{v} \mathbf{v}] = -\nabla p + [\nabla \cdot \boldsymbol{\tau}] + \mathbf{f}_b \quad 4.4$$

It is evident that equations 4.1 and 4.4 are strongly coupled and nonlinear, but as mentioned before the real problem is that, with this formulation, the pressure field appearing in 4.4 cannot be directly computed. The SIMPLE algorithm approaches this problem by:

- Discretizing the mass and momentum conservation equations
- Constructing a pressure equation by combining them
- Reformulating the Navier-Stokes equations in terms of the discretized momentum and pressure equations
- Solving iteratively

## 4.3 Simulation details

### 3.3.1 Mesh generation and grid independence study

The first step in every CFD calculation is the generation of the mesh, i.e. the discretization of the domain into small volumes (or surfaces, in a 2D case) which is required to allow for the numerical solution to be evaluated.

For the purpose of ensuring that the discretization error is sufficiently small, the mesh should be as refined as possible. At the same time, a finer mesh implies a larger computational effort. The systematic procedure leading to the choice of the appropriate mesh for a given problem is called *grid independence study*. Finally, a mesh is chosen based on a trade-off between accuracy (directly proportional to mesh refinement) and computational effort (inversely proportional to mesh refinement). This procedure can also provide an estimate of the discretization error, which should be kept below a tolerance which is problem-dependent.

To perform a grid independence study, three or more meshes are usually produced and then a sample calculation is performed for each one of them. Then the results are plotted against some quantity related to mesh refinement – usually, the total number of cells or a characteristic mesh size – and a convergence is expected.

Although OpenFOAM has a mesh generating utility (called *blockMesh*), another software has been chosen for this purpose, due to its greater versatility. The meshes have been created with Star CCM+, a commercial CFD software, and then imported in OpenFOAM with the help of a freely available routine called “*ccm26ToFoam*” custom MATLAB script. This has been done since it is much easier to obtain high quality grids in Star-CCM+ than it is in OpenFOAM and licenses were available at Politecnico di Torino. Nevertheless, in the future the objective is to employ procedures internal to OpenFOAM in order to go towards a fully *open-source* framework. For the same purpose, the MATLAB scripts which have been written during this thesis in order to perform complex input generation and data exchange

operations in an automatic fashion shall be translated in python® or OCTAVE® language.

Three meshes were generated, both for the coolant part and for each of the fuel pins. Due to the strong directionality of the problem – the coolant flow is mainly oriented along the z direction, the FA axis – a so called *extruded mesh* was chosen. This means that the meshes are created in 2D and then extruded along the z axis, resulting in triangular prismatic cells instead of the usual tetrahedral cells. Prism layers were added next to the walls of the assembly and the fuel pins to treat the boundary layer according to the turbulence model employed.

The choice of the turbulence model will be better discussed in Section 4.3.2; for now, it is anticipated that this choice imposes some conditions on the dimensions of the prism layers, i.e. the cells directly adjacent to the solid walls of the domain. In particular, a relevant quantity in this stage of the mesh generation is the normalized or dimensionless wall distance, generally referred to as  $y^+$  and defined as:

$$y^+ = \frac{u_* y}{\nu} \quad 4.5$$

Where  $u_*$  is the friction velocity at the wall (related to the wall shear stress and the fluid density at the wall),  $y$  is the distance to the wall and  $\nu$  is the local kinematic viscosity of the fluid. The turbulence model chosen for this work, the standard k- $\epsilon$  model, prescribes a value of  $y^+$  of at least 30 at every point on the walls. Therefore, since  $u_*$  and  $\nu$  are quantities that depend on the flow field and on the fluid, respectively, this implies a requirement on the minimum value of  $y$ , i.e. the dimension of the first prism layer.

At the same time, the maximum number of layers is constrained by the fact that the distance between neighboring pins is small. Moreover, care should be taken to the ratio between the volume of the near-core boundary layer cell and the first cell after the boundary layer, which should be of the order of 1/3. The latter feature can only be verified after the mesh has been generated. Even worse, verifying the condition on  $y^+$  even requires a preliminary calculation (conventionally, purely hydraulic, i.e. without

### 4.3 Simulation details

---

solving the energy equation) since the local friction velocity  $u_*$  which is contained in the definition of  $y^+$  cannot be estimated a priori.

For the fuel pins, the physics of the problem has been considered in the mesh creation. Since heat transfer in the pins is mostly radial, the problem is again strongly directional. This has been taken advantage of with a mesh consisting of concentric rings azimuthally subdivided.

To avoid the singularity that would occur by keeping this layout up to the pin center, the central part has been automatically meshed with tetrahedral cells. It should be noticed that, should the design foresee a central hole in the fuel pin, the mesh generation strategy for the fuel pins would not change, and possibly the “structured” layout of the grid could be retained for the entire radial extent of the pin.

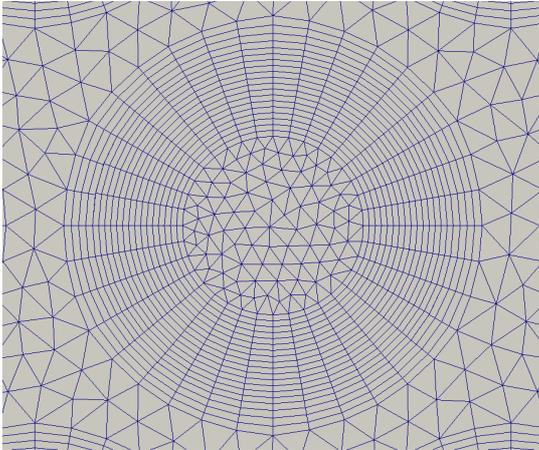
As will be better explained in the following section, the mesh makes no distinction between the fuel pin, the gap between pin and cladding, and the cladding itself. However, should this distinction be added in future works, the structured mesh could also be effectively applied to the cladding.

The aforementioned discussion conveys the difficulties associated to the generation of the first mesh. Then, once the main details are fixed (prism layers near wall thickness, structured pin mesh) the process continues with the generation of two other meshes, progressively refined and compliant with the constraints described above.

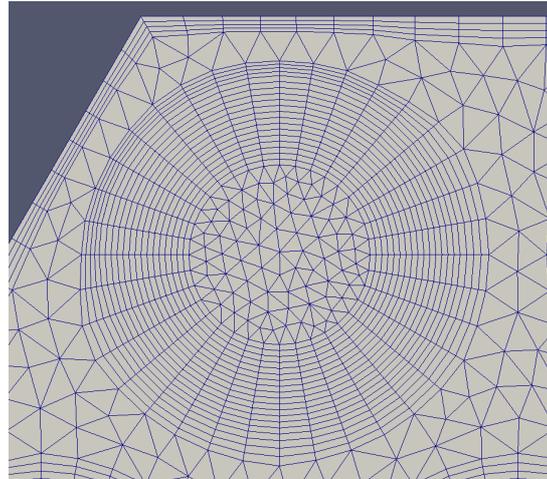
A numeric comparison of the three meshes’ main parameters is offered in **Table 4.1**, while a visual comparison is shown in **Figures 4.1 – 4.6**.

	<b>Coarse mesh</b>	<b>Medium mesh</b>	<b>Fine mesh</b>
<b># of coolant cells</b>	3,378,600	8,048,600	13,274,600
<b># of cells in fuel pin</b>	~72,000	~145,000	~216,800
<b># of prisms layers</b>	4	4	4

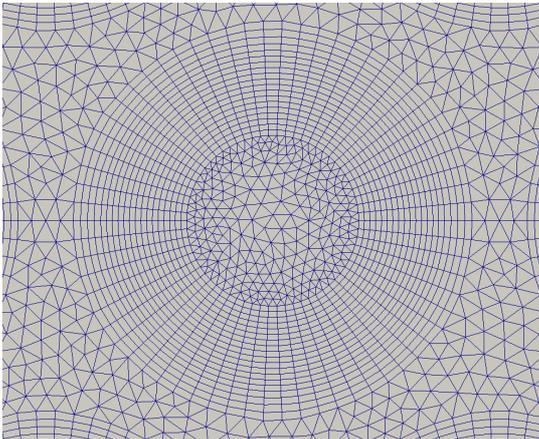
**Table 4.1.** Main mesh parameters for the three meshes. It should be noted that the number of cells per fuel pin is an average number, as each pin has been meshed separately and the actual numbers for each pin may vary.



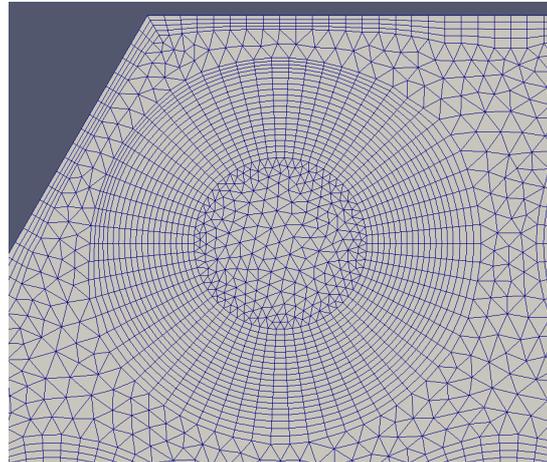
**Figure 4.1.** *Coarse mesh, central pin*



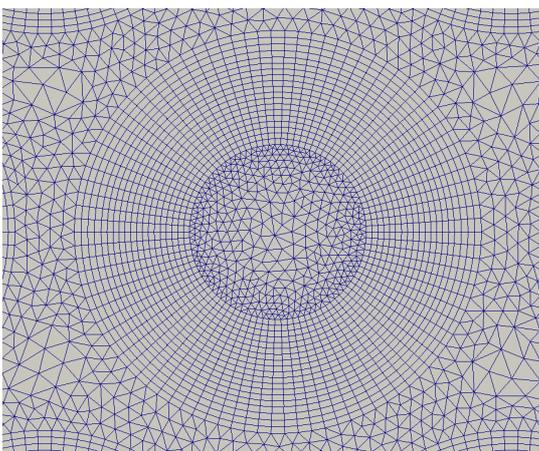
**Figure 4.2.** *Coarse mesh, corner pin*



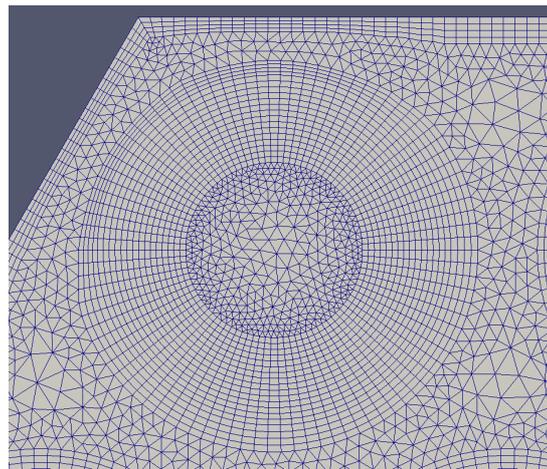
**Figure 4.3.** *Medium mesh, central pin*



**Figure 4.4.** *Medium mesh, corner pin*



**Figure 4.5.** *Fine mesh, central pin*

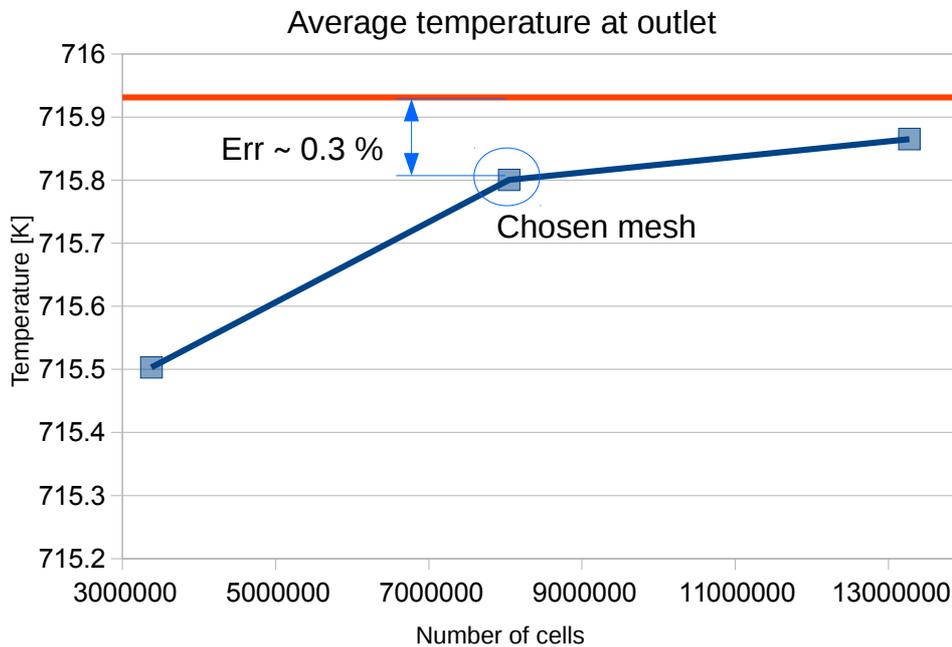


**Figure 4.6.** *Fine mesh, corner pin*

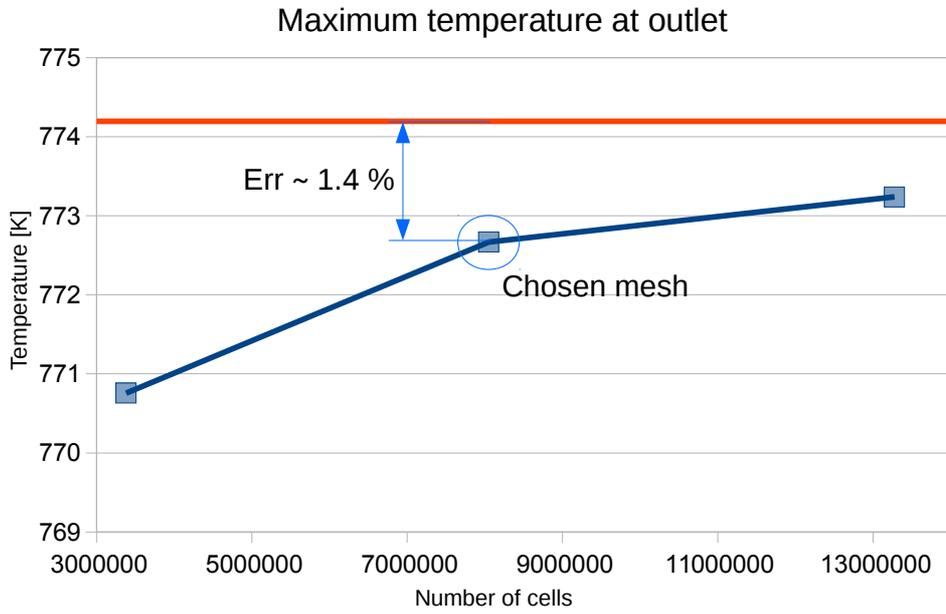
### 4.3 Simulation details

For the grid independence study, an assembly on the external border of the core was selected. A preliminary Serpent simulation was carried out to obtain a relevant power distribution for the FA under consideration. Since the results of this first phase were to be used for the grid independence only, being accurate in the definition of the core temperatures for evaluating the feedbacks on cross sections was not a priority. Therefore, in the neutronic simulation the fuel and coolant temperatures were kept constant throughout the core and equal to the fuel inlet temperature, without the zone subdivision described in section 3.4.1.

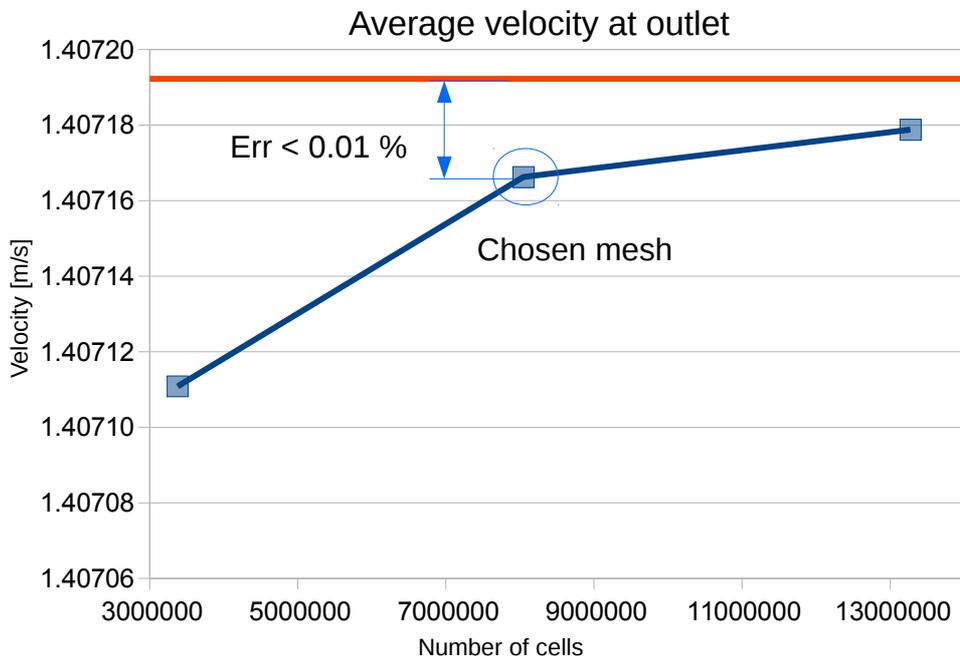
To assess grid independence, some quantities of interest have been considered, namely the coolant temperature and axial velocity over different sections of the FA; for both quantities, the average and the maximum value over the sections were selected. Some results of the analysis are presented in **Figures 4.7 – 4.10**.



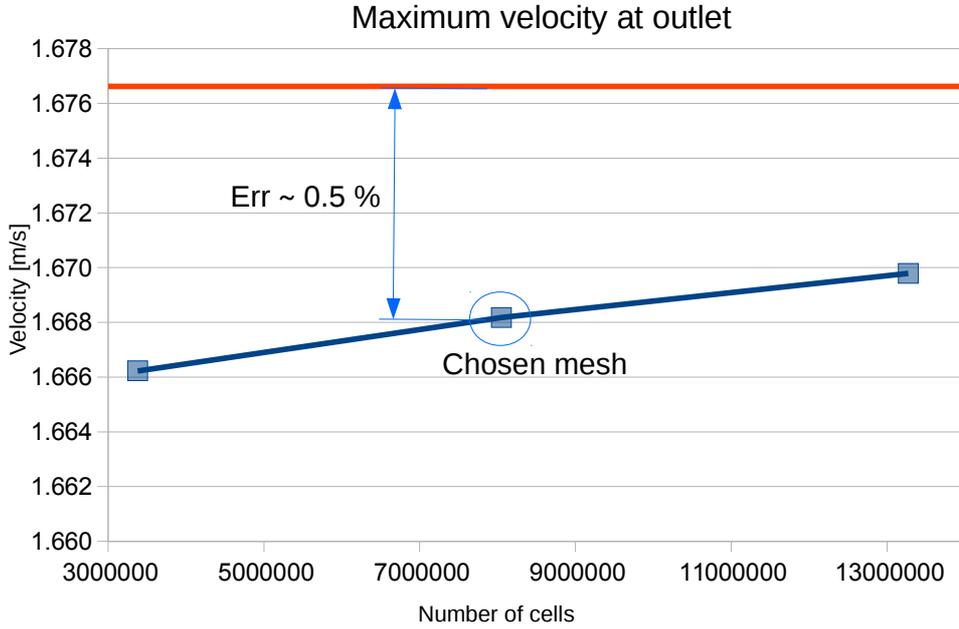
**Figure 4.7.** Average coolant temperature over a horizontal section positioned at 1 cm from the outlet



**Figure 4.8.** Maximum coolant temperature over a horizontal section positioned at 1 cm from the outlet



**Figure 4.9.** Average coolant axial velocity over a horizontal section positioned at 1 cm from the outlet



**Figure 4.10.** Maximum coolant axial velocity over a horizontal section positioned at 1 cm from the outlet

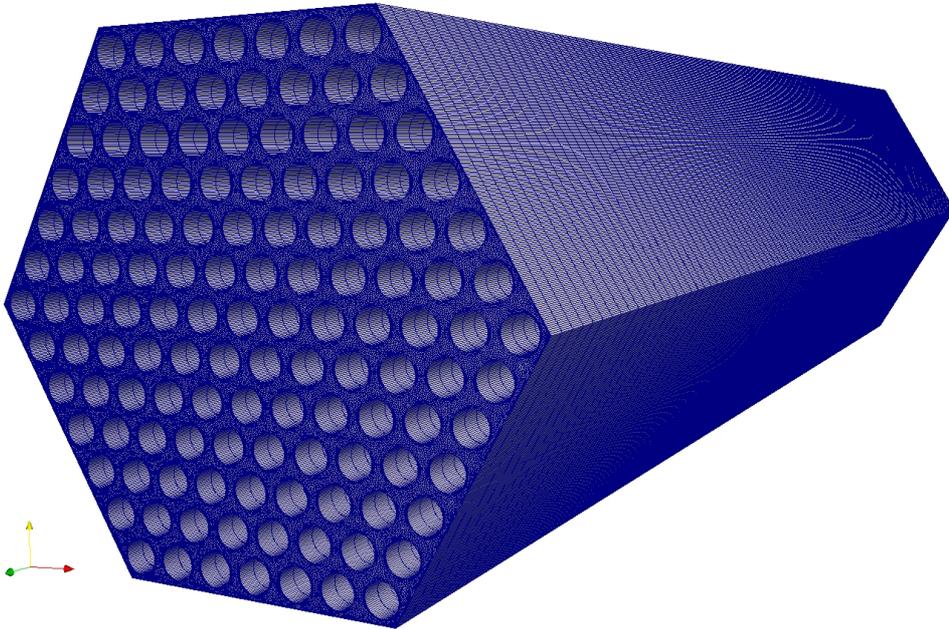
The graphs of **Figures 4.7 – 4.10** report the solution as a function of the three meshes total cell number. The quantities shown refer to an  $x$ - $y$  section of the FA taken at a distance of 1 cm from the outlet, to avoid every kind of forced boundary conditions. The red line in the graphs is the expected “real value” predicted by means of a Richardson extrapolation [20]. The errors shown in the graph are calculated with respect to this predicted value; for the velocity, the error is calculated as:

$$err = \frac{v_{predicted} - v_{calc}}{v_{predicted}} \quad 4.6$$

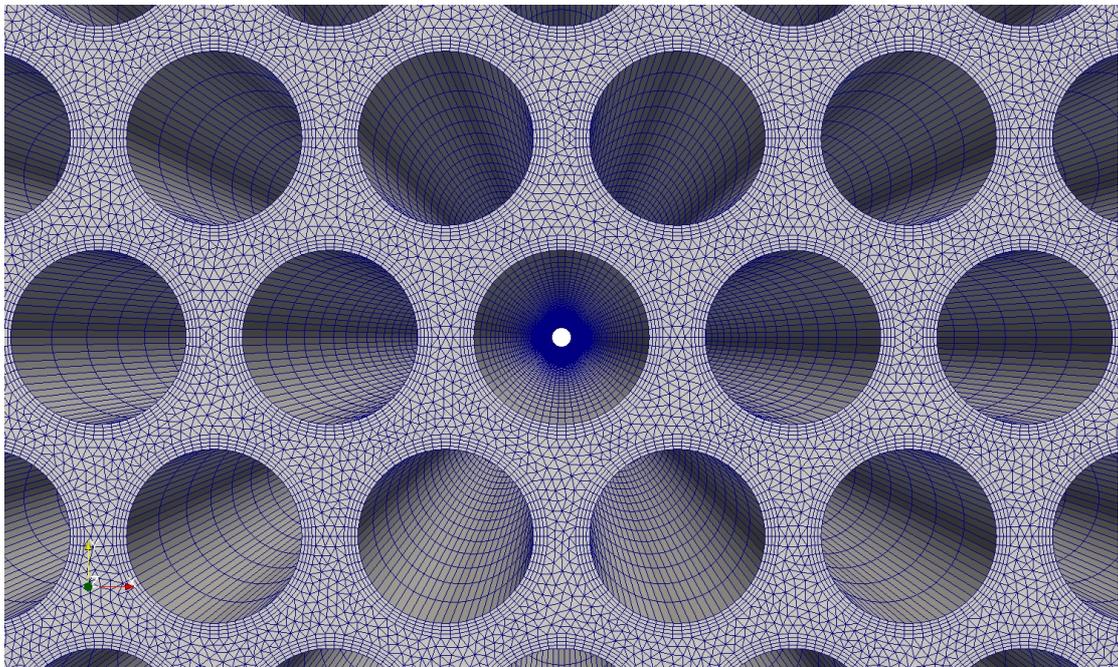
While for the temperatures, the error refers to the temperature jump between inlet and outlet:

$$err = \frac{\Delta T_{predicted} - \Delta T_{calc}}{\Delta T_{predicted}} = \frac{T_{predicted} - T_{calc}}{T_{predicted} - T_{inlet}} \quad 4.7$$

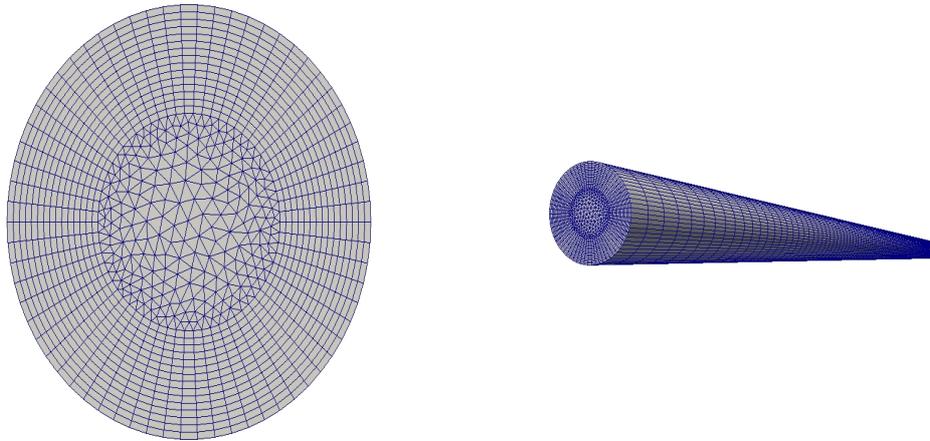
**Figures 4.11, 4.12** and **4.13** show the chosen mesh, with 8,048,600 cells in the coolant and ~145,000 cells in every fuel pin.



**Figure 4.11.** *Coolant mesh*



**Figure 4.12.** *Coolant mesh, inlet detail*



**Figure 4.13.** Fuel pin mesh, frontal (left) and side (right) view (Figure might be distorted in the electronic version of the file)

#### 4.3.2 Simulation setup

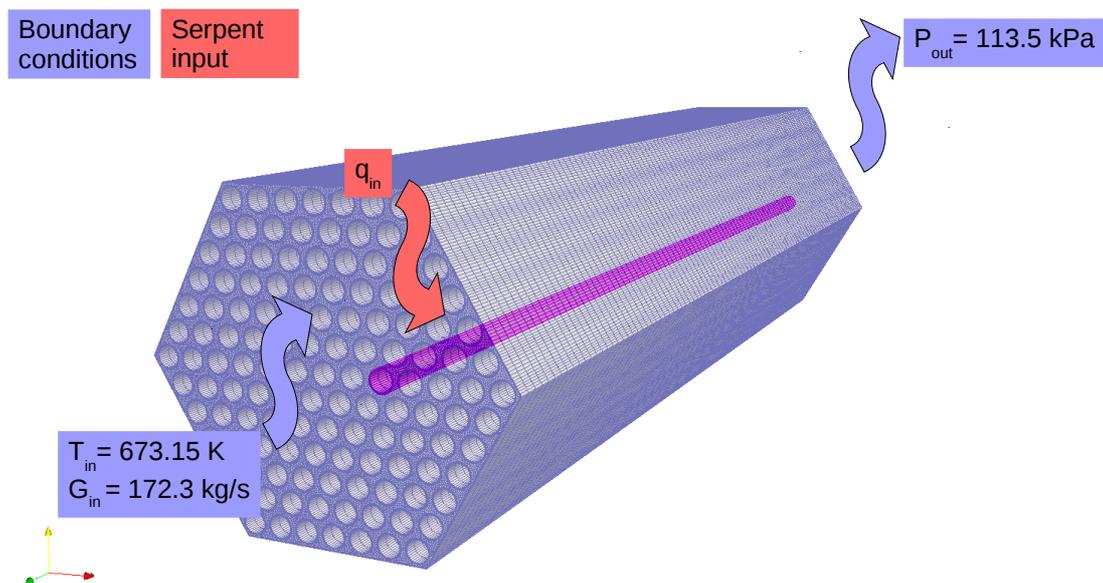
After having selected an appropriate mesh, the CFD simulation must be set up according to the physical situation to be evaluated.

An *a priori* estimate of the Reynolds number yields a value of  $\sim 60.000$ , therefore the flow of Pb along the fuel bundle can be considered turbulent. In order to deal with turbulence, the RANS approach (Reynolds-Averaged Navier Stokes) is selected, being by far more computationally affordable than LES (Large Eddy Simulation) and DNS (Direct Numerical Simulation). Among the many models which fall under the RANS framework, the standard  $k-\varepsilon$  model with conventional high  $y^+$  wall treatment is employed. This is the oldest and most widespread among the two-equation turbulence models. It is suitable for flows which are not characterized by an adverse pressure gradient, such as the one at hand [21]. It should be noticed that, if a wire-wrapped rod bundle was to be simulated, this would not hold true anymore and a different model should be selected. [22]. The standard  $k-\varepsilon$  model does not aim at resolving the boundary layer at the wall. Indeed, as mentioned in the previous section, it requires a dimensionless wall distance  $y^+ \geq 30$ , and it uses wall functions for reproducing the normalized velocity profile in the viscous sublayer [23]. The aforementioned cautions are believed to be sufficient for reliably describing the turbulent momentum transport

and would therefore be enough for a purely hydraulic simulation. Additional care should be taken in the present case, where heat transfer is extremely important and therefore turbulent heat transport should be properly described. Indeed, one of the characteristics of LMs is the extremely low Prandtl number ( $Pr \sim 0.016-0.019$  for Pb in the temperature range of the case in exam). The fact that  $Pr \ll 1$  implies that the conventional approach for relating turbulent momentum transport and turbulent heat transport are not strictly applicable. Indeed, recently many works have been performed for deriving turbulence models which are suitable for computational simulation of liquid metal flow and heat transfer in both bare and wire-wrapped rod bundles [24]. In the future, one or more of these models could be included in the solver. However, for the present work the approach suggested by Cheng et al. [25] has been followed, which suggests to modify the turbulent Prandtl number from the default 0.9 value (suitable for fluids with  $Pr \sim 1$ ) to 1.5. This value has also been employed in recent studies concerning the design of experimental facilities for heavy LM thermal-hydraulics in reactor-relevant configurations [26].

Concerning the fuel, no distinction is made between pins and cladding. Indeed, homogenized thermophysical properties are used and the volumetric power generation is assumed to occur in the entire fuel rod (even though, obviously, the total power is the same of the input SERPENT simulation). This choice does not affect the power transferred to the coolant per unit axial length – since the global energy balance of the fuel pin is unchanged – nor the calculated surface temperature of the pin. However, should the model be employed in the future to provide an evaluation of the temperature distribution *inside* the pin, the distinction between fuel and cladding materials and power generation should be implemented, together with a means for taking into account a finite thermal resistance between fuel and cladding (e. g. evaluated by another tool such as URGAP® or estimated according to correlations which can be embedded in *pwrThermoFoam*). This extension would be relatively straightforward, but it has not been possible to perform it during the present work.

All the thermophysical properties for the coolant are taken from [27]. The boundary conditions applied are summarized in **Figure 4.14**.



**Figure 4.14.** Boundary conditions for an assembly representative of cooling group number 1 (see Figure 4.15)

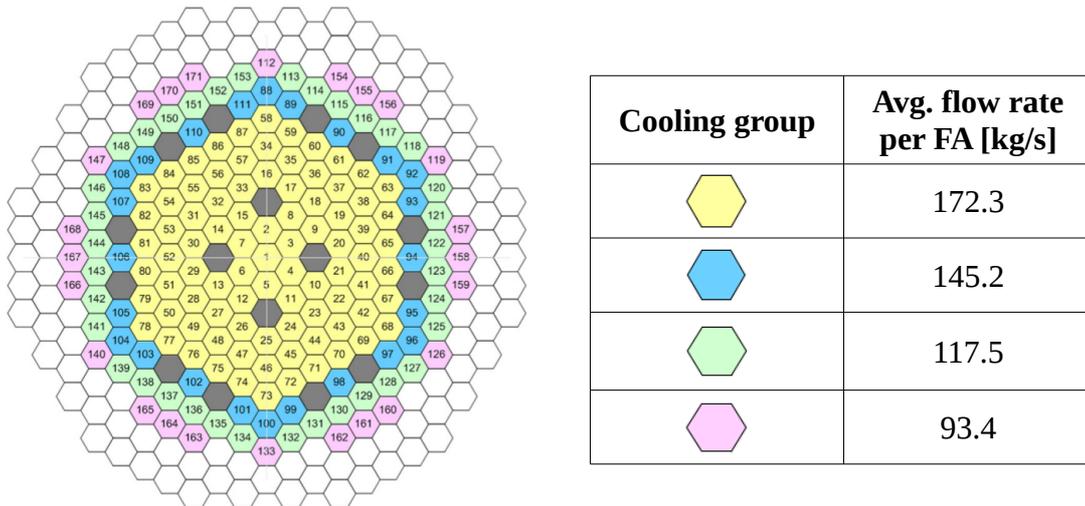
Coolant inlet temperature and mass flow rate are fixed, as well as outlet pressure. Concerning the inlet velocity, it should be noted that the OpenFOAM domain starts at the beginning of the core active zone; therefore, when the coolant enters this region its velocity profile is expected to be already fully developed.

At the assembly walls, a no-slip condition is imposed for the velocity and a fixed heat flux (i.e. a fixed gradient, Neumann BC) which results from the inter-assembly heat transfer is imposed. Of course, since the single-FA model is not capable of evaluating this heat flux, it is taken from the FRENETIC code. In some sense, we can therefore say that FRENETIC, in this work, has been employed to provide a full-core “environment” for the single-FA calculation.

Finally, at the pin interface, a no-slip condition is again selected for the velocity and a coupling BC between coolant and pin is chosen for the temperature. For the solid regions (i. e. the fuel pins) the lower and upper surfaces of the fuel pins are considered adiabatic, whereas the lateral surfaces are allowed to exchange energy with the neighboring fluid. As already stated, the volumetric power generation is taken from the NE code.

As far as BCs are concerned, it is worth noting how the inlet mass flow rate – and therefore, the inlet velocity – actually depends on the assembly position inside the core. In fact, to achieve a flatter outlet temperature distribution for the coolant, the ALFRED core was divided in four different cooling groups with mass flow rate decreasing radially from the center to the borders of the core. A scheme of the four cooling groups, with the corresponding average mass flow rate values, is reported in **Figure 4.15** [13].

Since the picture is very similar to Figure 3.4, it should be remarked that this is a “thermal-hydraulic” subdivision of the core, while the “neutronic” subdivision in inner and outer fuel zones has already been described above.



**Figure 4.15.** Cooling groups scheme

After having set the appropriate initial and boundary conditions related to the chosen assembly, the simulation is run until convergence is reached, i.e. until the solver stops iterating to evaluate the solution at each successive timestep.



## 5. The FRENETIC multi-physics code

### 5.1 Introduction

FRENETIC (Fast REactor Neutronic/Thermal-hydraulICs) [28] is a full-core multi-physics reactor analysis code for the dynamic analysis of liquid metal cooled fast nuclear reactors. The objective behind the development of the code is to have a coupled neutronics/thermal-hydraulics tool that can simulate the full-core of fast reactors during accidental transient of medium severity, while remaining computationally effective. Of course, some simplifications are required since a detailed description of the full-core distribution of temperature and neutron fields is not reasonably achievable, especially if a transient is involved. Some distinctive features of the code are the presence of advanced time-marching schemes for the neutronic calculations [29] and the thermal coupling between neighboring assemblies [30]. The code also features a steady state module, which is the one employed in the present work.

The neutronic module solves the multigroup neutron diffusion equations with delayed neutron precursors to evaluate the neutron flux in the core and the consequent volumetric power distribution. The problem is represented for each node by a system of  $N + M$  equations, where  $N$  is the number of energy groups into which neutrons are divided and  $M$  is the number of neutron precursor families. The equations are discretized in space through a nodal approach and in time with a quasi-static method.

The thermal-hydraulic module solves a set of four conservation equations; for the coolant, the full set of mass, momentum and energy equations is considered, while for the fuel the code adopts the assumption of immobile, non-deformable pins, therefore only the heat conduction equation needs to be solved. The fluid and solid equations are coupled at the material interface with the inclusion of appropriate heat source terms. The equations are discretized in space with the finite element method and in time with the theta method.

The full 3D thermal-hydraulic solution is actually achieved by splitting the problem in 1D (axial) +2D (radial). In fact, at each time step the code first solves the system of equations in 1D form for each single FA – the only direction being, in this case, the axial one – thereby obtaining a single, average value of the fuel temperature, the coolant temperature, the coolant velocity and the coolant pressure at each axial node of the FAs. Then, every FA is radially coupled with its surrounding adjacent neighbors, to obtain a full-core 3D solution.

Although the two modules can be run in stand-alone mode (NE or TH), the true objective of FRENETIC is the coupled simulation. This is achieved by using the spatial distribution of the power density evaluated by the NE module as input for the TH module, and in turn by using the fuel and coolant temperature distributions evaluated by the TH module as input for the NE module. In particular, while the power density explicitly appears in the TH energy conservation equations as a volumetric heat source term, the fuel and coolant temperatures do not seem to be found in the NE system of equations. Actually, as already noted in section 2.1, the temperature effects are implicitly accounted for in the macroscopic cross sections. The code therefore needs an additional auxiliary model accounting for the variation of macroscopic cross sections as a function of fuel and coolant temperature.

The chosen model consists of a simple linear interpolation between some reference cross section values evaluated at suitable temperatures. The interpolation is described by the equation:

$$\Sigma(T_f, T_c) = \Sigma(T_{f,0}, T_{c,0}) + \left( \frac{\partial \Sigma}{\partial T_f} \right)_{T_c} (T_f - T_{f,0}) + \left( \frac{\partial \Sigma}{\partial T_c} \right)_{T_f} (T_c - T_{c,0}) \quad 5.1$$

In FRENETIC, each macroscopic cross section for each reaction  $x$  in each energy group  $g$  for each material  $m$  is written in the form  $\Sigma_{x,g}^m(T_f, T_c)$ , where  $T_f$  is the fuel temperature and  $T_c$  is the coolant temperature. Since only two variables are needed to determine the cross sections values, and the minimum number of points needed to perform a linear interpolation is two, a minimum of four values would be needed for each cross section. Specifically, as explained in **Table 5.1**, these are calculated by considering for the two variables  $T_f$  and  $T_c$  a “cold” and a “hot” value; in this way, one

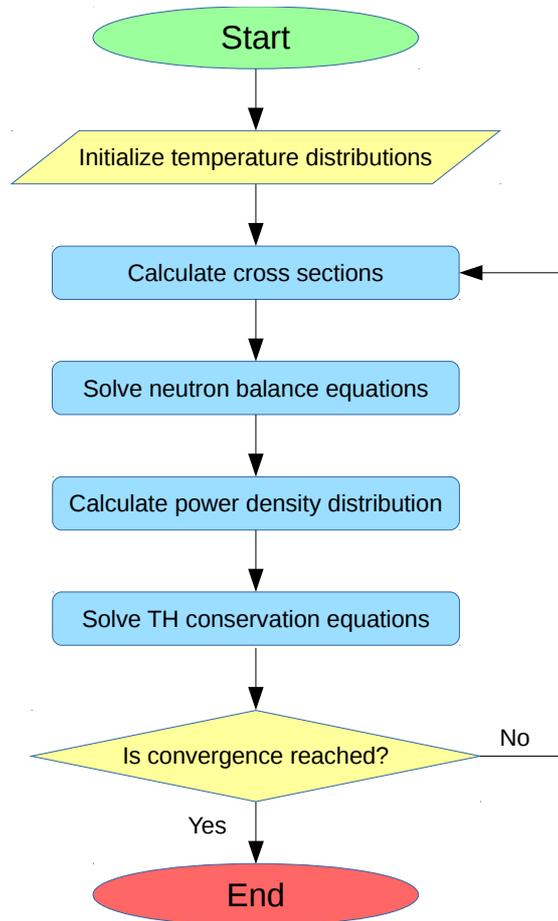
of the cases under consideration represents an unphysical condition in which the coolant temperature is higher than the fuel. For this case, the cross section values are just assumed to be zero, therefore only the other three Serpent simulation need to be executed.

	$T_{c,0} = 673$	$T_{c,1} = 1073$
$T_{f,0} = 673$	(673,673)	(673,1073)
$T_{f,1} = 1073$	(1073,673)	(1073,1073)

**Table 5.1.** Temperatures selected for cross section generation (values reported in Kelvin). The upper-right cell, in which the fuel temperature is smaller than the coolant one, actually represents an unphysical case, so all cross sections are simply assumed to be 0 in this case.

Once the reference cross section values are provided by Serpent, the interpolation can be performed by FRENETIC as shown in Eq. 5.1.

In this way, the feedback effects for both modules are taken care of and the coupling is completed. The full solving procedure for steady-state problems, in which the two modules are iteratively called until convergence is reached (i.e., until the variation in the power distribution and the temperature distributions between two successive iterations is lower than a certain predefined tolerance) is described in **Figure 5.1**.



**Figure 5.1.** FRENETIC steady-state solving algorithm



## 6. Simulation results

Results are presented following the flowchart of Figure 2.1. For the last part, i.e. the FRENETIC comparison with the CFD solution provided by OpenFOAM, complete results are still not available at the time of this thesis writing: instead, preliminary simulation results are shown to prove the functioning of the model.

### 6.1 NE preliminary simulations and benchmark

As already noted in previous sections, the goal of the three preliminary Serpent simulations was to evaluate nuclear data to be provided to FRENETIC for its first run. Therefore, for these simulations the detailed core power distribution was not evaluated, to improve code performance.

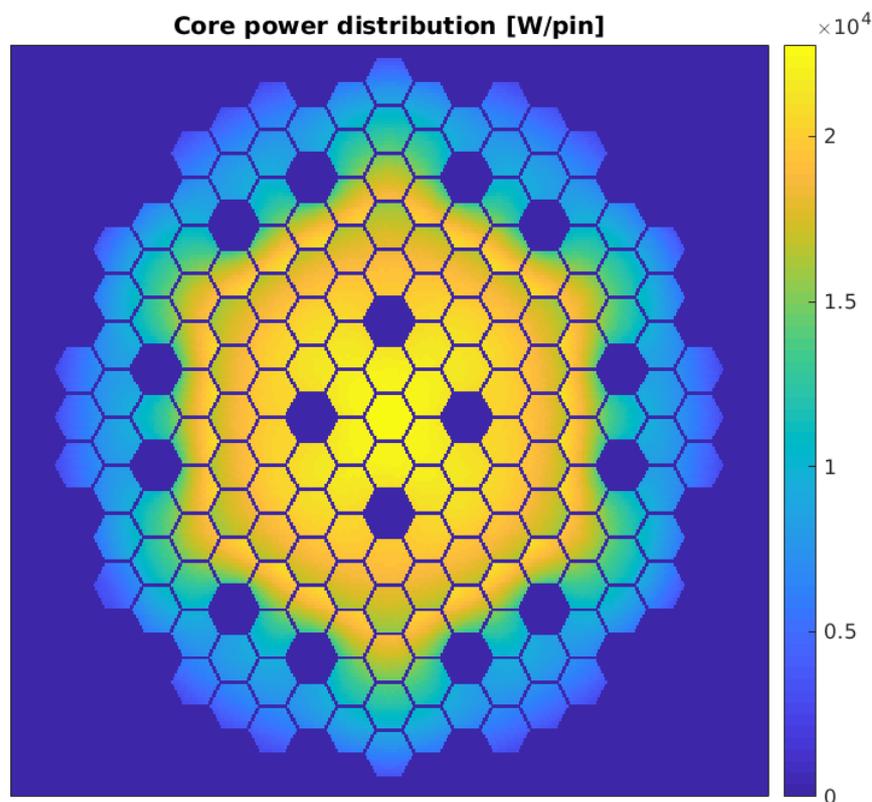
Nevertheless, the information on the reactor's effective multiplication factor in the three cases clearly shows the effect of temperature feedbacks on the simulations. In fact, as reported in **Table 6.1**, the value of k-effective shows a steep decrease from run #1 to run #2, in correspondence to the increase in the fuel temperature. Moreover, another much less critical but still appreciable decrease can be observed from run #2 to run #3, corresponding to the increase of the coolant temperature. This shows the greater importance of the fuel temperature coefficient on reactivity with respect to the coolant temperature coefficient.

	<b>Run #1</b>	<b>Run #2</b>	<b>Run #3</b>
<b>T<sub>f</sub></b>	673 K	1073 K	1073 K
<b>T<sub>c</sub></b>	673 K	673 K	1073 K
<b>k<sub>eff</sub></b>	1.00408 ± 2.4e-05	1.00141 ± 2.4e-05	1.00100 ± 2.4e-05

**Table 6.1.** Effective multiplication factor (with statistic uncertainty) for the three preliminary Monte-Carlo runs

Next, results of the Serpent Monte-Carlo simulation with consistent core temperature distribution evaluated by FRENETIC are presented and compared with the FRENETIC results.

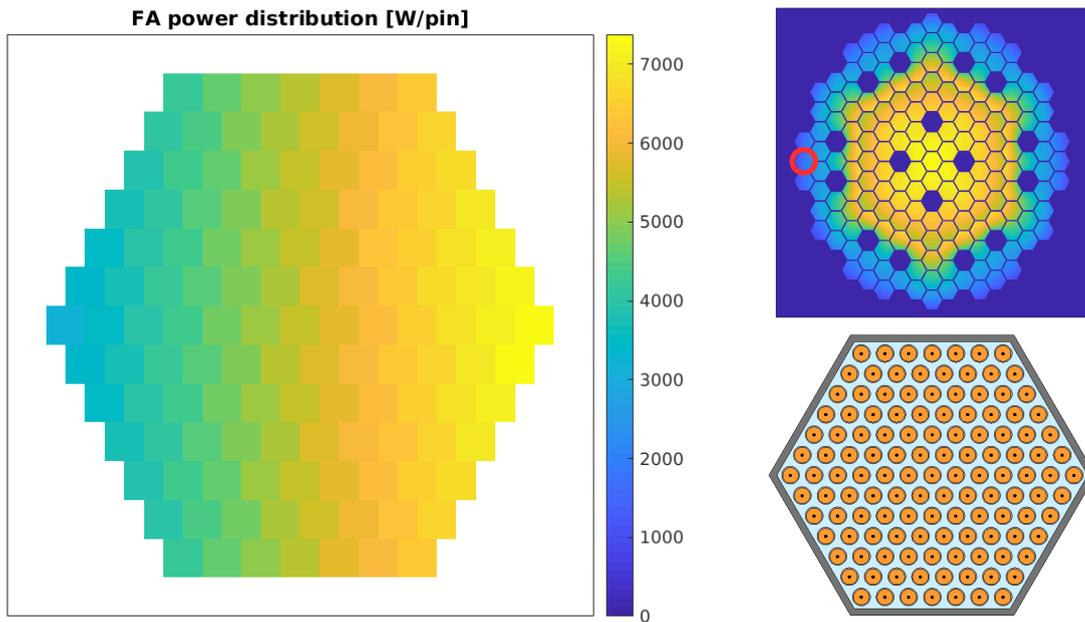
**Figure 6.1** shows the full core, pin-by-pin power distribution. The transition between inner and outer differently enriched fuel zones is clearly visible, as well as the effect of control rods which cause a flux depression in nearby assemblies.



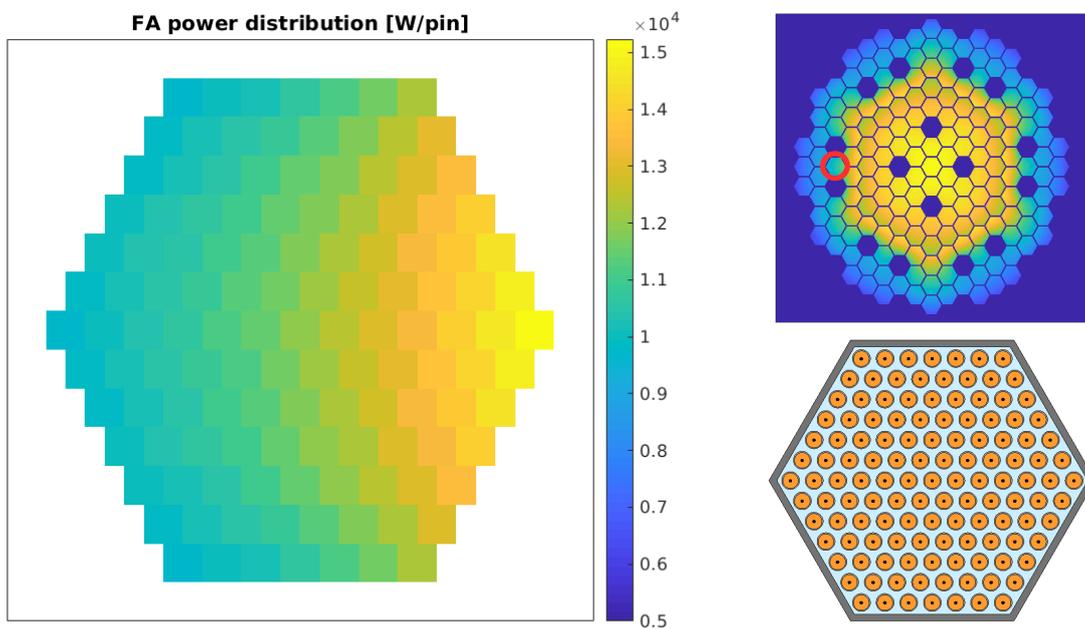
**Figure 6.1.** Pin-by-pin full core power distribution evaluated by Serpent

**Figures 6.2 – 6.4** focus on the power distribution inside the three fuel assemblies selected for the analysis. Depending on their position in the reactor core, the FAs will be intuitively referred to as “Border”, “Middle” and “Central” for the rest of the presentation. The skewness of the power distribution, which justifies the choice of these specific FAs to evaluate the limits of the FRENETIC assembly-averaged approach, is well visible in Figures 6.2 and 6.3. Note that a different scale has been adopted for the three plots due to the large differences in the average power per FA.

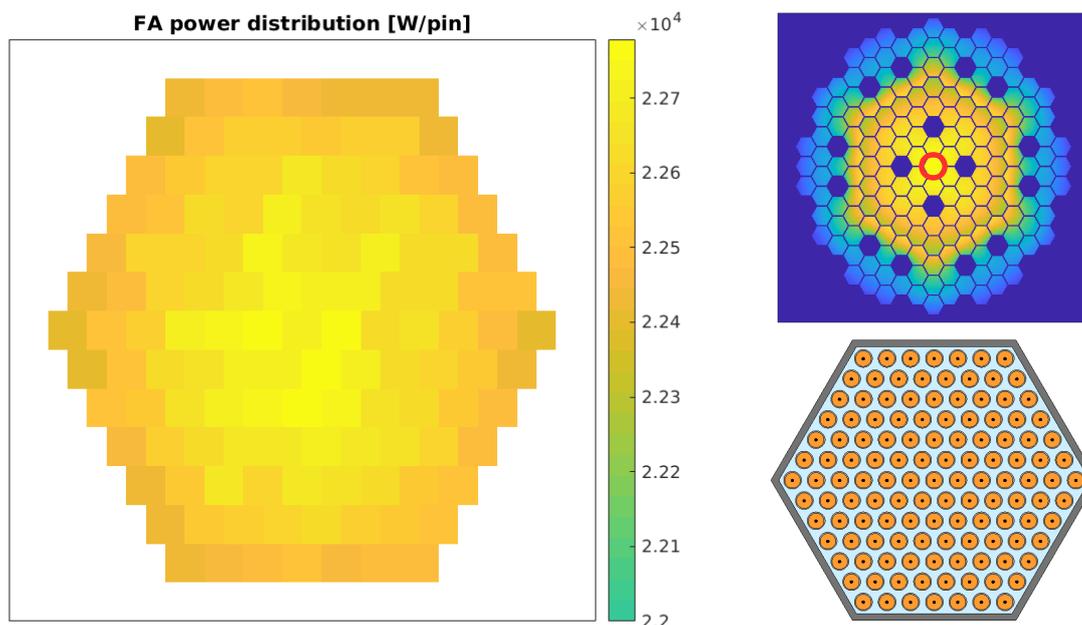
## 6. Simulation results



**Figure 6.2.** Pin-by-pin power distribution and core spatial collocation of the Border assembly



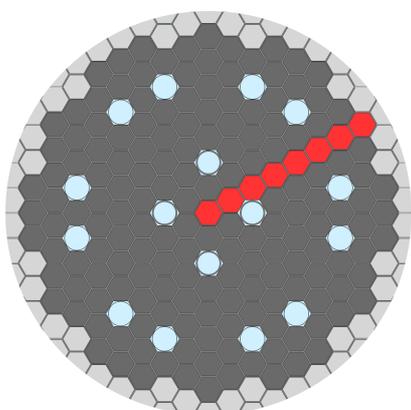
**Figure 6.3.** Pin-by-pin power distribution and core spatial collocation of the Middle assembly



**Figure 6.4.** Pin-by-pin power distribution and core spatial collocation of the Central assembly

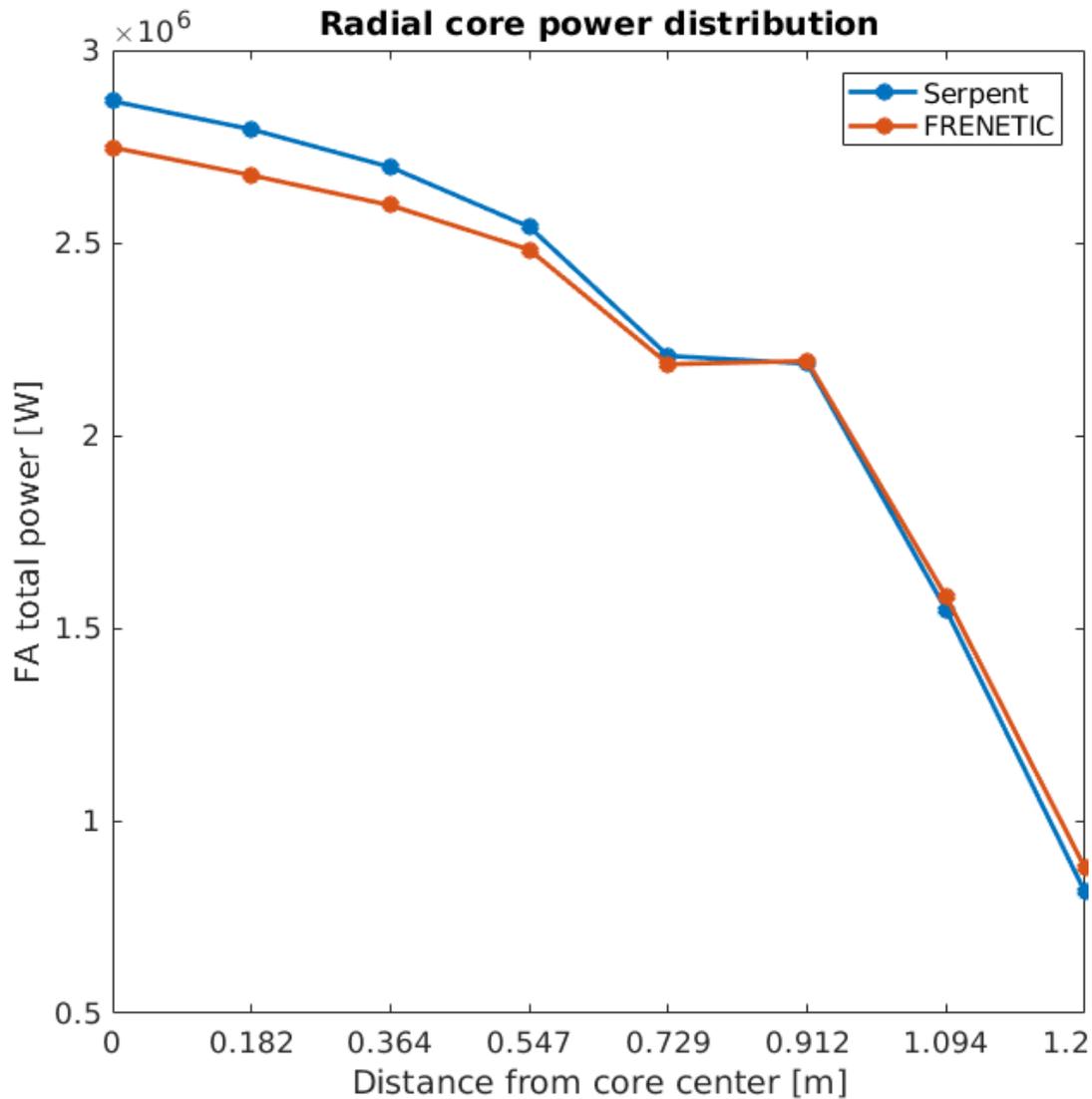
For the NE benchmark to be conducted, the pin-specific data from the Serpent simulation had to be compared to the FA-specific data evaluated by FRENETIC. Therefore, a sum over all the 127 pins per FA at each axial step was performed. Then, both the single assembly axial distribution and the full-core radial distribution were considered for the comparison.

**Figure 6.6** shows the power distribution as a function of radial distance from the core center. The values reported are relative to the total power for each FA. **Figure 6.5** shows the FAs selected for the plot.



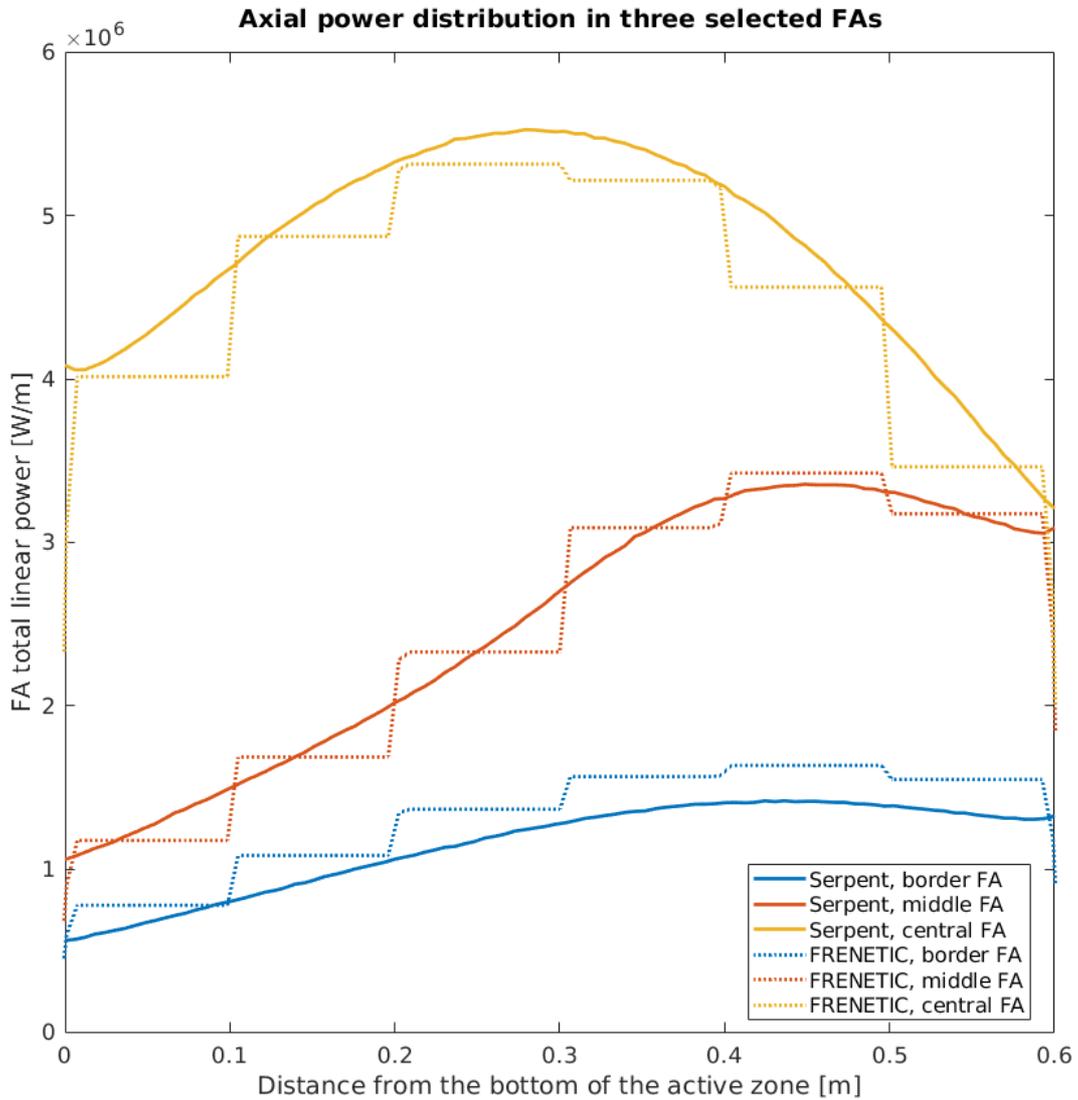
**Figure 6.5.** Map of the FAs selected for the radial plot of Figure 6.6

The radial plot of Figure 6.6 shows good agreement between the outputs of the two codes. The total power produced in the core is fixed as an input quantity and has to be the same, but it can be observed that the distribution evaluated by FRENETIC is somewhat flatter with respect to the Monte-Carlo one.



**Figure 6.6.** Total power produced for each FA as a function of the FA center's distance from the core center

In fact, the power in the central assemblies appears to be underestimated, while the opposite happens in the outermost assemblies, for which the FRENETIC calculation slightly overestimates the results. This same behavior can be observed in the axial plot of **Figure 6.7**. Here, the axial linear power distributions in the Border, Middle and Central FAs are reported. The node averaged FRENETIC power distribution is shown to better approximate the Monte-Carlo solution for the Middle assembly, overestimating the linear power in the Border one and underestimating it in the Central one.



**Figure 6.7.** Comparison of the axial linear power distribution in the three selected FAs

## 6.2 OpenFOAM preliminary simulations results

As already mentioned at the start of this chapter, the CFD simulations relevant for the TH comparison with FRENETIC are not yet available at the time this thesis is being written; nevertheless, the OpenFOAM model is complete. The results of the preliminary analyses conducted to assess its performances are reported in this section, after a brief resume of the boundary conditions, which for sake of simplicity slightly differ from the ones presented in chapter 4.

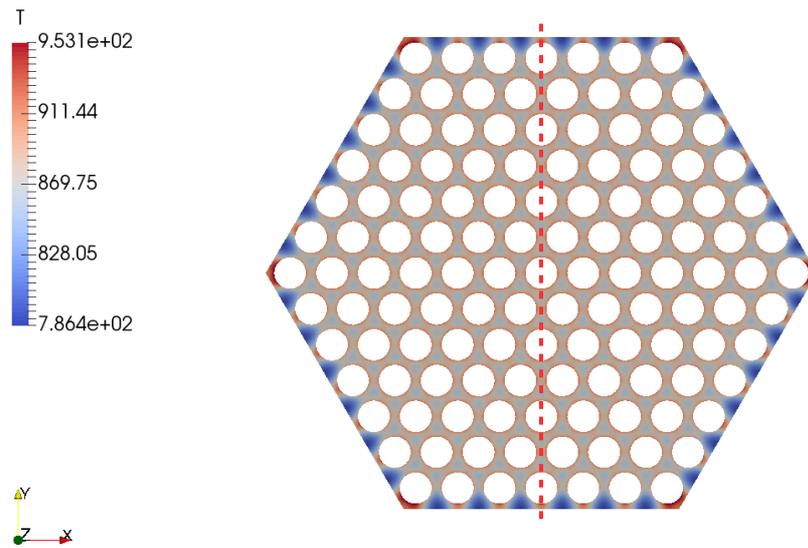
The analysis presented concerns the Central fuel assembly. The power distribution is evaluated by Serpent in a “cold” configuration, i.e. without having implemented the full-core temperature distribution provided by FRENETIC, considering both the fuel and the coolant temperature to be equal to 673 K everywhere in the core. Therefore, due to the absence of feedback effects, the power distribution in the central assembly is expected to be higher than in the real case. Consequently, also the temperature field evaluated by the CFD analysis is expected to have higher values.

The assembly walls are considered to be adiabatic, which is actually a special (homogeneous) case of the Neumann boundary condition presented in chapter 4.

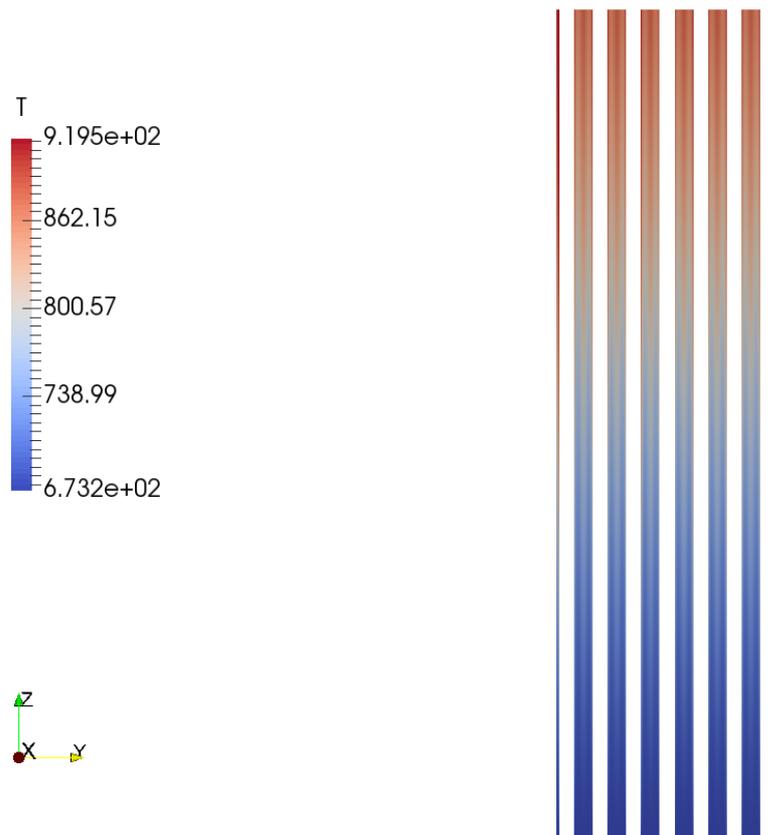
Finally, the coolant velocity in this case is homogeneously fixed on the inlet section. This implies that the first part of the assembly will not have a fully developed flow, as is the case in FRENETIC. Therefore, the results presented here are all evaluated at sections sufficiently far from the inlet (with the exception of plots showing the entire FA length), to avoid effects related to the entrance length.

**Figures 6.8** and **6.9** show the coolant temperature distribution respectively for an  $x$ - $y$  section at 1 cm distance from the outlet ( $z=0.59$  m from the start of the active zone) and for an axial ( $y$ - $z$ ) section, the position of which is indicated by the red line in Figure 6.8. Some notable features are the hot channels at the FA corners, where the coolant temperature peaks, and the particularly high temperature jump between inlet and outlet, due to the high values of the volumetric power in the fuel rods. In the currently ongoing design review of the ALFRED core this issue has been accounted for and corrected by increasing the distance between the corner pins and the FA walls. **Figures 6.10** and **6.11** show the coolant pressure distribution for the same sections described above. **Figures 6.12** and **6.13** show, respectively, the temperature and axial velocity distribution for the coolant in a sub-channel between two adjacent fuel pins, near the center of the FA. Finally, a full 3D plot of the coolant temperature is offered in **Figure 6.14**. **Figures 6.15** and **6.16** show a temperature plot for the FA’s central fuel pin. Again, the extremely high centerline temperature can be explained with the absence of feedback effects in the preceding NE calculation.

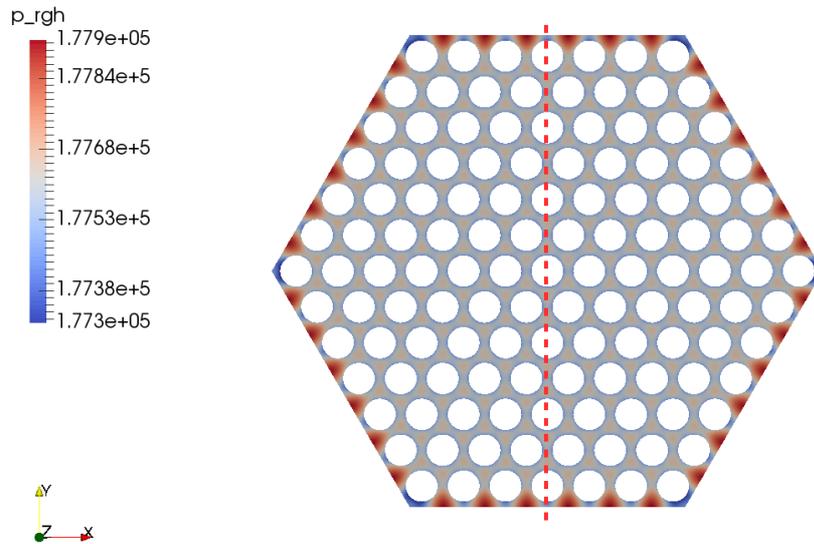
## 6. Simulation results



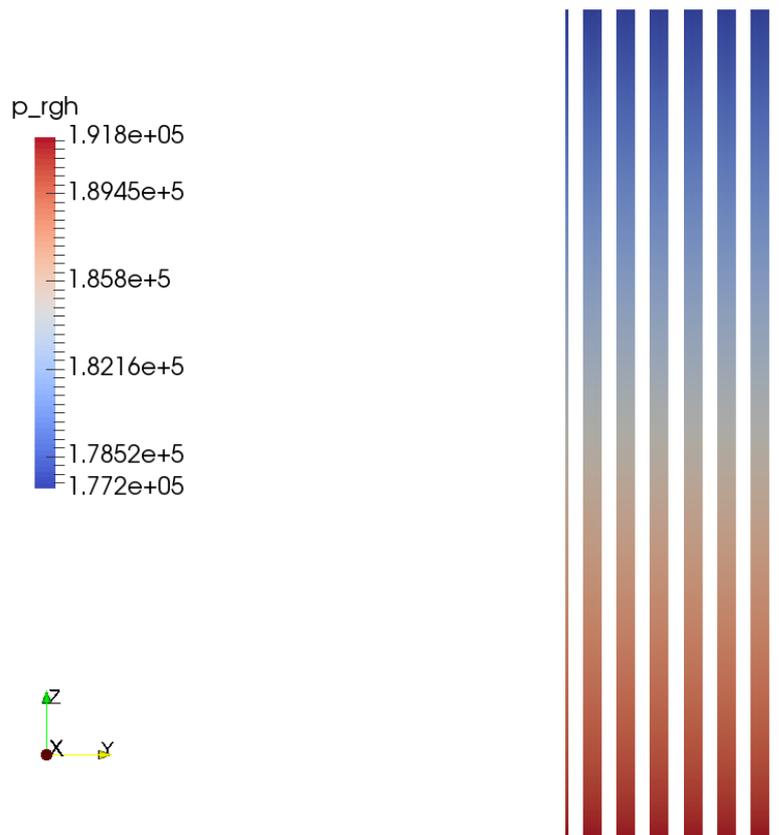
**Figure 6.8.** Temperature distribution for the Central FA at  $z=0.59$  m. The red line shows the position of the section in Figure 6.9



**Figure 6.9.** Axial temperature distribution for the Central FA

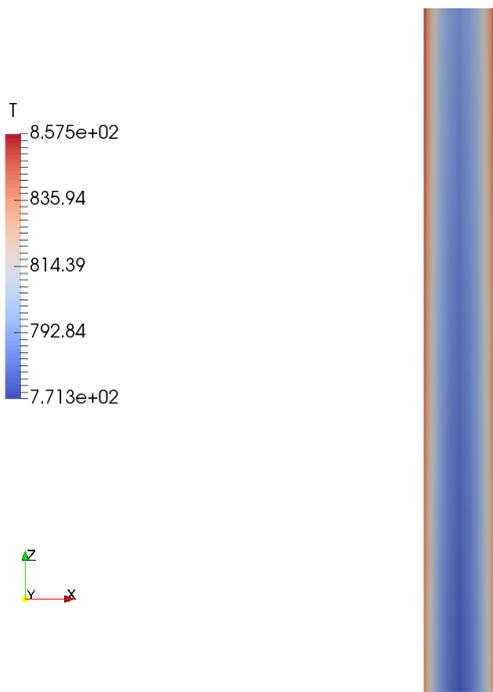


**Figure 6.10.** Pressure distribution for the Central FA at  $z=0.59$  m. The red line shows the position of the section in Figure 6.11

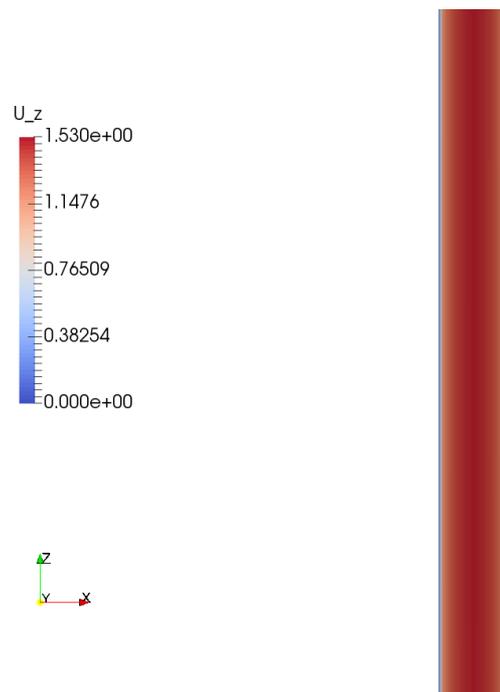


**Figure 6.11.** Axial pressure distribution for the Central FA

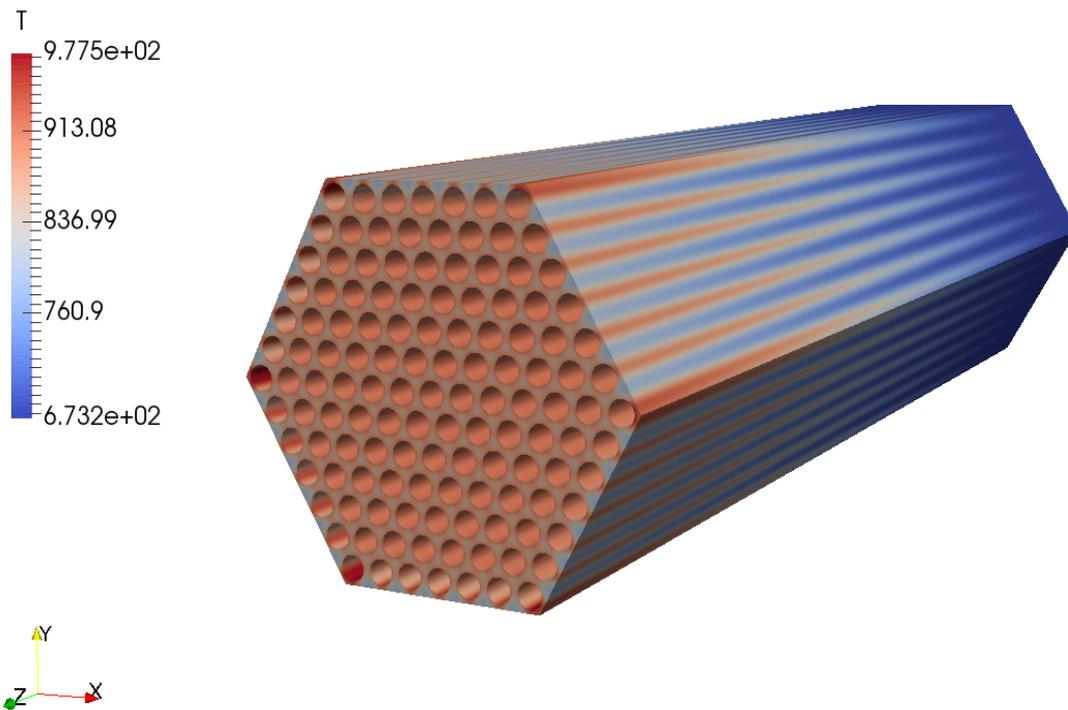
## 6. Simulation results



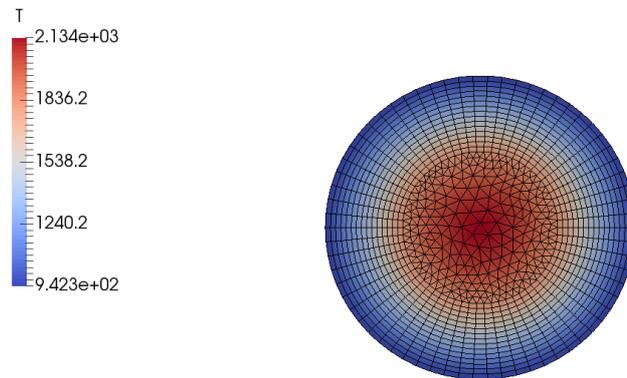
**Figure 6.12.** Central sub-channel section between  $z=0.29$  and  $z=0.31$ , coolant temperature plot



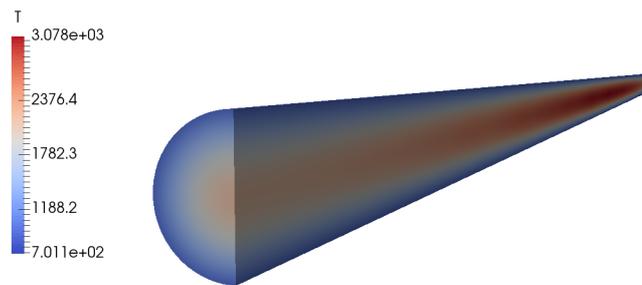
**Figure 6.13.** Central sub-channel section between  $z=0.29$  and  $z=0.31$ , coolant axial velocity plot



**Figure 6.14.** Full 3D coolant temperature plot.



**Figure 6.15.** Central fuel pin temperature at the end of the active zone ( $z=0.6$  m)



**Figure 6.16.** Central fuel pin temperature, 3D section plot

The model results shown in the plots are consistent with expectations based on the physics of the problem. For example, **Figures 6.8** and **6.12** show that the coolant temperature is higher in proximity of the fuel pins and lower near the sub-channels core. The higher coolant temperature shown at the channels centers in **Figure 6.9**, which seems to contradict this statement, is actually explained by the close positioning of fuel pins in the  $x$  direction, perpendicular to the section and therefore not visible in the  $y$ - $z$  section plot. In **Figure 6.13** the no-slip condition applied on the fuel pin surface is well visible. The highly nonuniform temperature profile at the walls shown in **Figure 6.14** reflects the absence of a Dirichlet boundary condition (as

## 6. Simulation results

---

explained above, a Neumann BC is employed in this case). Finally, **Figures 6.15** and **6.16** show a fuel temperature profile in accordance with the volumetric heat source distribution (higher near the center of the active zone).

## **7. Conclusions and perspectives**

In this thesis work, a coupled multi-physics model has been developed which foresees the use of a full-core Monte-Carlo module for neutronics and a CFD module for thermal-hydraulics. To avoid the excessive computational burden of a full-core CFD simulation, the full 3D TH analysis was restricted to a single FA. The temperature field in the rest of the core has been approximated with the 1D+2D approach embedded in the FRENETIC multi-physics code.

Results of both modules are encouraging. The comparison between the NE module and FRENETIC shows a good agreement even though FRENETIC predicts a slightly flatter power distribution throughout the core. Preliminary CFD simulations of suitably selected FAs proved the reasonable behavior of the TH module.

The near future developments foresee the completion of the benchmark of the FRENETIC code in steady state, both for the stand-alone TH and for the coupled NE/TH simulation mode, by means of the iterative NE/TH solving capabilities of the model developed in this work.

This thesis has been carried out in the framework of a fruitful partnership between Politecnico di Torino and Politecnico di Milano. The models originally developed by each of the two institutions have profited from the collaboration, which has provided a good insight on the strengths and weaknesses of the models themselves and on the different approaches adopted therein.



## Acknowledgements

This thesis work marks the conclusion of my Master's Degree studies in Energy and Nuclear engineering; this is the obvious part, but it's more than that. It is, in the long list of life's "endings", the most important I've faced in my existence up until now. It probably won't be the ending of my academic career, but it's the end of an incredible journey that made me deeply change, inside and out, that made me grow, that helped me shape my life and my future, that ultimately made me the person I am today, as I write these crucial lines. A journey through which, luckily for me, I was never truly alone, even if for some short periods it may have felt like it; and the least I can do now is to look back and thank the wonderful people who held my hand along the path. From the very bottom of my heart, I feel the need to say "thank you":

- To my family, most of all my mother, father and sister. It's easy and trivial to say that you've supported me, but not many can imagine how much this is true in my case. You have always given me everything I needed and more, without flinching, even if I never truly deserved all the love, patience and care you donated me. Most importantly, you basically never asked for anything in return. I hope that the goal I am achieving today may make you proud of me, and may repay even a small part of the sacrifices you made for me.

- To my tutors Sandra Dulla and Antonio Cammi, and to Giuseppe Francesco Nallo, for giving me the initial idea for this work and the help I needed to complete it. Your support and your encouragement has been crucial to push me through the harder stages of the work, and to reassure me when things seemed to go as wrong as possible.

- To my Poly2Nuc "brothers in arms": Rosario, Alessio and Francesco. You are probably the ones that know better than anyone else how dark the darkest moments of the last two years were for me, simply because it was the same for you. But those hard times have made us strong and close, and I feel a special bond with you that I hope will not fade in the future. Now that we all have reached the end of this adventure, it is time to look back and remember not only the sufferings, but also the pure moments of joy and fun we shared in the past years, without ever losing our *baricentro*; and to look forward, to the bright future that awaits all of us, each in its own path.

- To my friends: Frenk, Zio, Lu, Ale, Della, Dado, Gis, Ele, Paolo, and all the crew. If I turn to look at the best moments of my life in the last five/six years, you are present in almost each and every one of them. When life got hard, seeing you and spending time with you was always the way to cheer me up, and make me forget my troubles. The adventures I shared with some of you are the craziest of my life, and will be the tales I tell my grandchildren when I'm old and tired... if I ever get there, after all the adventures we still have to face together, because *scafunno will never die*.

- To Francisca, for her wonderful friendship: we started as simple "desk buddies", but we became much more. You have seen me grow and change, and you helped me throughout. You have been a safe haven when I needed someone to talk to, and when I needed counsel you were always there. Even when life started to drift us apart, in the last three years, we didn't lost our bond, which makes me hope that we will manage to do that also in the future, whatever that will be. And by the way, I'm sure your future will be fantastic, because you are Francisca from Turin!

- To Bianca. Compared with all the other people in this list, the amount of time I've known you is pretty ridiculous. You basically entered my metaphorical journey as I was already seeing the finish line standing before me. Yet, in such a short time you have already taken your place here, alongside the most important people in my life, and I can only hope you're here to stay. Here's to the many more journeys we shall take together.





## References

- [1] Intergovernmental Panel on Climate Change, *Climate change 2014: mitigation of climate change*, report, 2014
- [2] <http://resourceirena.irena.org/gateway/dashboard/index.html>, IRENA (International Renewable ENergy Agency) website
- [3] Generation IV International Forum, *2007 Annual Report*, 2007
- [4] A. Alemberti, *Final Report Summary - LEADER (Lead-cooled European Advanced Demonstration Reactor)*, CORDIS (Community Research and Development Information Service), 2013
- [5] R. Bonifetto, S. Dulla, P. Ravetto, L. Savoldi Richard, R. Zanino, *A full-core coupled neutronic/thermal-hydraulic code for the modeling of lead-cooled nuclear fast reactors*, Nuclear Engineering and Design 261 (2013) 85– 94, 2013
- [6] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho, *The Serpent Monte Carlo code: Status, development and applications in 2013*, Annals of Nuclear Energy, 82 (2015) 142-150, 2015
- [7] J. Leppänen, *Development of a new Monte Carlo reactor physics code.*, Doctor of Science Thesis, Helsinki University of Technology, 2007.
- [8] A. Dubi, *Monte Carlo methods in nuclear reactor analysis*, ISPRA course handouts
- [9] B. Montagnini, *Fondamenti di fisica del reattore*, course handouts
- [10] J. Leppänen, *Serpent - a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code: User's Manual*, 2015
- [11] [http://serpent.vtt.fi/mediawiki/index.php/Result\\_estimators](http://serpent.vtt.fi/mediawiki/index.php/Result_estimators), *Serpent online wiki*
- [12] I. Lux, L. Koblinger. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Inc., 1991.
- [13] G. Grasso, C. Petrovich, D. Mattioli, C. Artioli, P. Sciora, D. Gugiu, G. Bandini, E. Bubelis, K. Mikityuk, *The core design of ALFRED, a demonstrator for the*

*European lead-cooled reactors*, Nuclear Engineering and Design 278 (2014) 287–301, 2014

[14] G. Grasso, Private correspondence, Nov. 2017

[15] F. B. Brown, *On the Use of Shannon Entropy of the Fission Distribution for Assessing Convergence of Monte Carlo Criticality Calculations*, PHYSOR-2006 American Nuclear Society's Topical Meeting on Reactor Physics Organized and hosted by the Canadian Nuclear Society, Vancouver, BC, Canada, Sept. 10–14 2006.

[16] C. J. Greenshields, *OpenFOAM User Guide – Version 5.0*, OpenFOAM foundation, Ltd., 2017

[17] <http://www.openfoam.com>, *OpenFOAM official website*

[18] P. Bianchini, *Computational features of neutronics and thermal hydraulics coupled simulations*, Master thesis, Politecnico di Milano, 2017

[19] F. Moukalled L. Mangani M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®*, Springer, 2016

[20] P. J. Roache, *Quantification of uncertainty in fluid dynamics*, Annu. Rev. Fluid Mech., vol. 29, pp. 123-160, 1997

[21] M. Tarantino, P. Gaggini, V. Labanti, G. Venturi, A. Class, K. Liftin, N. Forgione, V. Moreau, *Final Report on Integral Circulation Experiment in CIRCE*, IP EUROTRANS, domain DEMETRA, Task 4.5.3, ENEA ET-F-S003, 2010.

[22] L.M. Brockmeyer, F.S. Sarikurt, Y.A. Hassan, *Cfd investigation of wire-wrapped fuel rod bundles and flow sensitivity to bundle size*, NURETH-16, Chicago, IL, 2015

[23] D. C. Wilcox, *Turbulence modeling for CFD*, DCW Industries, 3rd edition, 2006

[24] S. Manservigi, F. Menghini, *Four Parameter Heat Transfer Turbulence Models for Heavy Liquid Metals*, J. Energy Power Sources Vol. 2, No. 2, pp. 63-74, 2015

[25] X. Cheng, N. Tak, *Investigation on turbulent heat transfer to lead–bismuth eutectic flows in circular tubes for nuclear applications*, Nuclear Engineering and Design 236 (2006) 385–393, 2006

- [26] G. Bandini et al., *CIRCE Experimental setup design and test matrix definition*, Integral test IT-F-S-001, 2011
- [27] V. Sobolev, *Database of thermophysical properties of liquid metal coolants for GEN-IV*, SCK•CEN-BLG-1069, 2011
- [28] D. Caron, *Neutronics methods for the multi-physics analysis of nuclear fission systems*, Doctor of Science Thesis, Politecnico di Torino, 2017
- [29] R. Bonifetto, S. Dulla, P. Ravetto, L. Savoldi Richard, R. Zanino, *Full-core coupled Neutronic-ThermalHydraulic model of innovative lead cooled fast reactors*, Transactions of the American Nuclear society, vol. 106, Chicago, Illinois, June 24-28 2012
- [30] D. Caron, R. Bonifetto, S. Dulla, V. Mascolino, P. Ravetto, L. Savoldi, D. Valerio, R. Zanino, *Full-core coupled neutronic/thermal-hydraulic modelling of the EBR-II SHRT-45R transient*, International Journal of Energy Research, Special issue paper, 2016