Master's Thesis
Sustainable Nuclear Energy Engineering

Development of a neutronic calculation scheme in support to innovative reactor configurations for advanced fuel cycle modelling and multi-physics analysis

Tutor:
Dr. Barbara, VEZZONI

Author:
Angela, MASSA

University tutor:
Prof. Sandra, DULLA

A.Y. 2024/2025

Abstract

In the framework of advanced nuclear fuel cycle and long-term nuclear energy sustainability analysis, major attention is devoted to resource optimization and radioactive waste reduction. In this context, fast reactors are considered one of the most promising technologies.

Since the 1960s, several countries have been developing fast reactor technologies, with France being a major contributor to R&D, experiments, and the operation of fast reactors (FRs). In this framework, the Commissariat à l'Energie Atomique et Aux Énergies Alternatives (CEA) has developed several codes for modeling FR behavior under nominal and transient conditions. Concerning neutronics, the ECCO/ERANOS codes have been extensively used in the past. Recently, new-generation codes, such as APOLLO3®, have been progressively implemented and tested, particularly for sodium-cooled fast reactors.

This work aims to analyze the neutronic calculation scheme based on the APOLLO3® code for a lead-cooled fast reactor. Specifically, the 300 MWth ALFRED lead-cooled reactor, as proposed in the OECD/NEA LFR benchmark, has been analyzed. All the steps necessary for calculations with APOLLO3® have been reviewed, starting from geometry generation via the ALAMOS tool. Pre-processing and post-processing Python scripts have been developed to automate part of the workflow. Additionally, several parametric investigations have been conducted to explore safety considerations, multi-physics analysis, and fuel cycle strategies.

To consolidate the results obtained, a verification step was included at each phase of the study, leveraging the APOLLO3® TRIAGE module for the automatic generation of TRIPOLI-4® inputs. Furthermore, comparisons against available OpenMC results have also been performed.

List of Figures

List of Tables

List of Acronyms

API Application Programming Interface

CEA Commissariat à l'Energie Atomique et aux énergies alternatives

CPM Collision Probability Method

EDF Electricité de France

EFR European Fast Reactors

EGPRS Expert Group on Physics of Reactor Systems

FOM Figure of Merits

FR Fast Reactors

GUI Graphical User Interface

HDF Hierarchical Data Format

LFR Lead Fast Reactors

LWR Light Water Reactors

MOC Method of Characteristics

MPO Multi Parameters Output

NEA Nuclear Energy Agency

OECD Organization for Economic Cooperation and Development

SFR Sodium Fast Reactors

SSERMA (Service d' Études des Réacteurs et de Mathématiques Appliquées

OLFR Open-literature pool-type Lead-cooled Fast Reactor

TDT Two- and Three-Dimensional transport

MC Monte Carlo

FA Fuel Assembly

EOC End Of Cycle

Contents

1 Introduction

In recent years, there is a renewed interest in developing innovative reactor concepts, particularly in the area of Fast Reactors (FR) and co-generation systems. Fast reactors are considered one of the most promising solutions for closing the nuclear fuel cycle [1], [2], [3], [4]. Their favorable neutron balance facilitates multi-recycling of plutonium-based fuels, resulting in improved utilization of resources, including uranium. In addition, FR and related fuel cycle strategies may help address challenges related to waste management [5], [2].

In the past, numerous countries have developed significant research programs in the area of FRs. In this context, France emerged as a leading nation in the field, initially with the *Rapsodie* test reactor, and subsequently with the commercial size prototype reactors *Phénix* and *Superphénix* [6]. One of the key organizations driving research and development (R&D) on FRs in France was, and continues to be, the Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA). In particular, between 2010 and 2019, CEA initiated the ASTRID project (Advanced Sodium Technological Reactor for Industrial Demonstration) and through multiple countries initiative reinforced R&D actions on sodium-cooled FR technologies. The ASTRID prototype was conceived to include significant improvements especially in terms of safety demonstration (e.g. improve core design with negative sodium void coefficient, passive and active safety systems, etc.) and to demonstrate the full fuel cycle closing at the industrial scale [7], [8]. In addition, the experience and knowledge gained, provided also during the EU projects (from EFR [9] to more recent CP-ESFR [10] up to ESFR-SMART [11] projects) led to substantial advancements in terms of design, technology development and safety analysis including codes, tools and methods development.

The Generation-IV International Forum (GIF) [12] recognizes other concepts to explore potential alternatives for long term nuclear energy sustainability. Among the six concepts proposed, the lead-cooled fast reactors (LFRs) technology is considered as a promising options for the next generation of nuclear energy systems due to its potential for enhanced safety, sustainability, and efficiency thanks to the high thermodynamic properties, such as the high boiling temperature, high thermal conductivity and high heat capacity [13], [14].

However, the advancement of LFR technology faces significant challenges, concerning materials but also due to the limited number of operational examples[1] that can be used to validate and verify the theoretical physics and the performances of the reactor core. In order to cope with this gap, several activities have been proposed during the years in the different projects. These activities include also neutronics and thermal-hydraulics comprehensive benchmarks. Such benchmarks are crucial to provide a standardized framework to evaluate reactor designs, facilitate the comparison of different calculation methods, support the development of reliable simulation codes and in general fostering the physical understanding. This will allow also to share techniques among the international community and to support any possible future practical application of this technology.

1.1 Thesis' objectives and structure

It is in this framework, that the Master's Thesis work is inserted, focusing on the neutronics analysis of a specific LFRs reactor concept, the 300 MWth ALFRED (Advanced Lead Fast Reactor European Demonstrator) reactor, proposed for benchmark within the OECD/NEA Expert Group on Reactor Physics [16].

The present work, indeed, focuses on the application of the new generation deterministic code

---

[1] Such as the Russian experience within the BREST reactors under construction[15]

APOLLO3® to a lead-cooled fast reactor concept supporting code multi-purpose further applications development [17]. APOLLO3® is currently under development at the CEA and it has been extensively applied to sodium-cooled FRs concepts replacing previous generation codes ECCO/ERANOS (e.g. [18], [19], [13]). In the study, education and training activities based on the unitary codes have been prioritized[2] focusing on the analysis of all the steps necessary for calculations with APOLLO3® including the generation of unstructured geometries via the ALAMOS tool. The last versions of the codes have been applied for providing the majority of the results required by the OECD/NEA benchmark. During the work, few feedback have been provided to code's developers for improving code robustness.

In addition to the required benchmark results, several parametric studies have been carried out to explore safety considerations, multi-physics analysis, and fuel cycle strategies. Therefore, temperatures, coolant density and adopted materials variations have been considered with respect to heir impact to some reactivity coefficients, essential for potential multi-physics studies. The impact of neutrons data libraries, self-shielding models and solver options has been evaluated as well.

To consolidate the results obtained, a verification step against stochastic simulations, such as TRIPOLI-4® , was included at each phase of the study, leveraging the APOLLO3® TRIAGE module for the automatic generation of TRIPOLI-4® inputs. In addition to the reference TRIPOLI-4® calculations, OpenMC results have also been used for comparison[3].

The codes and tools adopted are described in Chapter 3 and the LFR case study in Chapter 4.

The results required by the LFR benchmark [16] are discussed at Chapter 5 following the different phases composing the benchmark: cell, assembly and super cell.

The complementary parametric studies performed are collected in Chapter 6. In particular, given the experience of CEA on SFR analysis, the same calculation performed for the LFR case have been carried out also for a specific type of sodium-cooled fast reactor [21]. To complete the analysis, considerations about advanced fuel cycles have been briefly introduced via the study of a simplified Open-literature pool-type Lead-cooled Fast Reactor (OLFR) used as a test case: a theoretical core design that allows the introduction of depleted uranium pins in the assembly configuration contributing to the fuel mass balance of the system (Section 6.3).

Finally, conclusions and perspectives are recalled at Chapter 7.

1.2 CEA-SERMA department at Saclay

The Master's Thesis work has been carried at SERMA (Service d'Études des Réacteurs et de Mathématiques Appliquées) a research department within the French Alternative Energies and Atomic Energy Commission (CEA) in Saclay (Paris).

CEA is a major research organisation working in the best interests of the French State, its economy and citizens. Thanks to its strong roots in fundamental research, it is able to provide tangible solutions in four key fields: Low-carbon energy (nuclear and renewable), Digital technology, Technology for medicine of the future and Defence and national security. The SERMA department is mainly involved in the study and development of nuclear reactors, with

---

[2] The INCA [20] platform may be applied to this type of study for facilitate the work. The choice of using unitary codes has been driven by the needs of education and training for showing all, or the major, required actions that the users need to consider if an domain-specific interface is not available.

[3] OpenMC has been used as internal CEA comparison for supporting the analysis of the results obtained and the cross-checking with the ones provided by the OECD/NEA benchmark participants.

a particular focus on reactor physics, numerical simulations, and applied mathematics. The department's work includes: developing advanced simulation tools and methodologies for reactor physics. APOLLO3® and TRIPOLI-4® and others widely used codes for reactor design and safety assessments has been developed at SERMA.

2 Background concepts

The design and analysis of nuclear reactors is strictly related to the precise prediction of neutrons behavior, in terms of space, energy, angle and time dependence. Neutrons are, in fact, responsible of energy release and chain reaction maintenance thanks to fission, scattering and absorption interaction; moreover they lead to a thermo-mechanical response, inducing degradation in the different components of the reactor. For this reason, an interdisciplinary study is necessary to ensure operational safety and economic advantage throughout the lifetime of a nuclear reactor.

The neutron transport process through matter is described by the transport equation: a linear version of Boltzmann's equation originally developed within the framework of the kinetic theory of gases. [22]

2.1 The neutron transport equation

Neutron transport is the mechanism by which neutrons evolve in a physical system.

This process draws its origin from the classical Boltzmann equation, which corresponds to a non-linear integro-differential equation describing thermodynamic systems, through a statistical approach.

The neutron transport equation is essential for understanding the neutrons' behavior in a medium and for implementing the reactor physics analysis. As previously mentioned, it corresponds to a simplified version of the original Boltzmann's equation, based on the seven following assumptions [23]:

- neutrons are treated as point particles;
- neutrons travel in a straight line since they have no charge;
- interactions between neutron particles are neglected;
- collisions are instantaneous;
- the medium is assumed to have isotropic properties;
- the material composition and its nuclear properties are considered as known;
- the mean value of neutron density is considered.

The neutron transport analysis is based on the basic concept of the conservation of neutron number. Thus, considering the fundamental mechanisms involved in the neutron transport equation, the variation in time of neutrons number in the element $d\boldsymbol{r}$, having an energy E and a direction of motion Ω, can be defined as:

variation in neutron number over time = - Leakage rate (L) - Removal rate (R) + Production rate (Q)

Where:

- Q includes the scattering source referring to neutrons coming from energy E' and direction Ω' to energy E and direction Ω, the fission source and the generic external source;

- L accounts for the loss of neutrons due to their movement out of a given region in space, as the difference between neutrons exiting the elemental volume dV and the ones entering the elemental volume dV per unit of time;

- R considers the neutrons which are removed from the system due to absorption or scattering. This term introduce the concept of total macroscopic cross-section as the sum of the absorption and scattering one ($\Sigma_t = \Sigma_a + \Sigma_s$), which corresponds to the probability per unit path for a neutron of energy E and at position $\mathbf{r}$ of experiencing a collision.

Developing the different terms, the mathematical expression of the neutron transport equation is:

$$\begin{aligned} &\frac{1}{v(E)}\frac{\partial\phi(\mathbf{r},E,\mathbf{\Omega},t)}{\partial t} + \mathbf{\Omega}\cdot\nabla\phi(\mathbf{r},E,\mathbf{\Omega},t) + \Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E,\mathbf{\Omega},t) = \\ &= \int dE' \oint d\mathbf{\Omega}'\Sigma_s(\mathbf{r},E')\phi(\mathbf{r},E',\mathbf{\Omega}',t)f_s(\mathbf{r},E'\rightarrow E,\mathbf{\Omega}'\cdot\mathbf{\Omega},t) + S(\mathbf{r},E,\mathbf{\Omega},t) \end{aligned} \tag{2.1}$$

In the equation above, the unknown variable is the angular flux $\phi(\mathbf{r}, E, \Omega, t)$, which describes the distribution of neutron paths across a specific region of the phase space at a given time t. On the left-hand side of the equation, lies the integral nature of the equation itself, in particular $f_s(\mathbf{r}, E' \rightarrow E, \Omega\text{'}{\cdot}\Omega, t)$ represents the so-called *scattering probability function*, which corresponds to the probability of a neutron to change energy (E') and direction (Ω') to an energy (E) and a direction (Ω) after a scattering event [23].

By applying this general case to the specific one of a nuclear reactor composed of fissionable material, an additional term accounting for fission reactions becomes necessary. This term has a similar structure as the scattering one; it is built starting from the fission cross section (Σ_f) and the fission probability function ($\chi(\mathbf{r}, E)$) which represents the probability of obtaining a neutron of energy E at position $\mathbf{r}$ from a fission reaction. This last is normally approximated as isotropic.

Thus, the equation 2.1 may be rewritten as follow:

$$\begin{aligned} &\frac{1}{v(E)}\frac{\partial\phi(\mathbf{r},E,\mathbf{\Omega},t)}{\partial t} + \mathbf{\Omega}\cdot\nabla\phi(\mathbf{r},E,\mathbf{\Omega},t) + \Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E,\mathbf{\Omega},t) = \\ &= \int dE' \oint d\mathbf{\Omega}'\Sigma_s(\mathbf{r},E')\phi(\mathbf{r},E',\mathbf{\Omega}',t)f_s(\mathbf{r},E'\rightarrow E,\mathbf{\Omega}'\cdot\mathbf{\Omega}) + \\ &+ \frac{\chi(\mathbf{r},E)}{4\pi}\int dE' \oint d\mathbf{\Omega}'\nu(E')\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E',\mathbf{\Omega}',t) + S(\mathbf{r},E,\mathbf{\Omega},t) \end{aligned} \tag{2.2}$$

Where $\nu(E')$ corresponds to the average number of neutrons produced by a fission interaction [24].

However, the control of a nuclear reactor is made feasible thanks to the existence of delayed neutrons, which are not emitted immediately after a fission event. These neutrons originate from the radioactive decay of certain fission products, known as delayed neutron precursors. Unlike prompt neutrons, which are released almost instantaneously during fission, delayed neutrons are emitted with a time lag that can range from milliseconds to nearly a minute, depending on the half-lives of the precursors involved. This significant difference in emission times between prompt and delayed neutrons is what fundamentally allows for effective control of the reactor. Thus, the complete set of equations becomes [24]:

$$
\begin{cases}
\dfrac{1}{v(E)}\dfrac{\partial \phi(\mathbf{r},E,\mathbf{\Omega},t)}{\partial t} + \mathbf{\Omega}\cdot\nabla\phi(\mathbf{r},E,\mathbf{\Omega},t) + \Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E,\mathbf{\Omega},t) = \\
\qquad = \displaystyle\int dE' \oint d\mathbf{\Omega}'\Sigma_s(\mathbf{r},E')\phi(\mathbf{r},E',\mathbf{\Omega}',t) f_s(\mathbf{r},E'\rightarrow E,\mathbf{\Omega}'\cdot\mathbf{\Omega}) + \\
\qquad\qquad + \dfrac{\chi_P(\mathbf{r},E)}{4\pi}\displaystyle\int dE' \oint d\mathbf{\Omega}'\nu(1-\beta)\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E',\mathbf{\Omega}',t) + \\
\qquad\qquad\qquad + \displaystyle\sum_i^R \dfrac{\chi_D(\mathbf{r},E)}{4\pi}\lambda_i C_i(\mathbf{r},t) + S(\mathbf{r},E,\mathbf{\Omega},t) \\
\frac{\partial C_i(\mathbf{r},t)}{\partial t} = -\lambda_i C_i(\mathbf{r},t) + \int dE\nu\beta\Sigma_f(\mathbf{r},E)\phi(\mathbf{r},E,\mathbf{\Omega}',t)
\end{cases}
\tag{2.3}
$$

Where:

- β is the fraction of delayed neutrons emitted by one fission reaction;
- $C_i(\mathbf{r},t)$ represents the concentration of delayed neutron precursors of the i-th group at position $\mathbf{r}$ and time t;
- $\chi_D(\mathbf{r},E)$ is the energy spectrum of delayed neutrons;
- λ_i is the decay constant of the i-th group of delayed neutron precursors. It represents the rate at which the precursors decay, emitting delayed neutrons.

In the present analysis, delayed neutrons are excluded from the mathematical and physical formulation of neutron transport within the reactor core. Although this approximation is generally justified for steady-state analyses and criticality evaluations. Nonetheless, it is important to acknowledge the essential role delayed neutrons play in reactor kinetics [22].

Finally, considering a steady-state problem, the time dependent term can be neglected such as the time variable, and so the delayed neutrons contribution can be neglected. Then, removing the external source term and introducing the multiplication factor k, the eigenvalue of the system. This last term represents the neutron multiplication factor, quantifying the change in the neutron population from one generation to the next under steady conditions.The equation becomes:

$$
\begin{aligned}
\mathbf{\Omega}\cdot\nabla\phi(\mathbf{r},E,\mathbf{\Omega}) + \Sigma_t(\mathbf{r},E)\phi(\mathbf{r},E,\mathbf{\Omega}) &= \\
= \int dE' \oint d\mathbf{\Omega}'\Sigma_s(\mathbf{r},E')\phi(\mathbf{r},E',\mathbf{\Omega}') f_s(\mathbf{r},E'\rightarrow E,\mathbf{\Omega}'\cdot\mathbf{\Omega}) &+ \\
+\frac{1}{k}\frac{\chi(\mathbf{r},E)}{4\pi}\int dE' \oint d\mathbf{\Omega}'\nu(E')\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E',\mathbf{\Omega}')&
\end{aligned}
\tag{2.4}
$$

When the neutron transport equation accounts for the actual geometry and boundary conditions of the reactor, this eigenvalue k corresponds to the effective multiplication factor k_{eff}, which represents the ratio between the number of neutrons in one generation and the number of neutrons in the generation before. If its value is equal to 1, it means that the number of neutrons does not change from one generation to another, the correspondent reactor is so called reactor. If $k_{eff} > 1$ the reactor constitutes a supercritical system, instead $k_{eff} < 1$ refers to a subcritical system, not able to sustain the chain reaction.

2.2 Boundary conditions

Regarding the boundary conditions, they can be of different types: *vacuum*, *periodic* and *reflecting* or *albedo*.

In this context, the last type of boundary condition has been considered, since it aligns with the approach used for the simulations and evaluations conducted in this thesis.

The known outgoing flux is imposed equal to the incoming one, according to Robin type condition:

$$\phi(\mathbf{r}_s, E, \mathbf{\Omega}) = \beta\phi(\mathbf{r}_s, E, \mathbf{\Omega}') \qquad \Omega \cdot n_s < 0 \tag{2.5}$$

In particular, n_s is the unit normal vector directed outward from the surface at the boundary position r_s. The term β describes the type of neutron reflection, if $\beta = 0$ the outgoing particles will not come back into the system, instead, $\beta = 1$ corresponds to a reflection of the particle.

2.3 Multigroup method

Typically, in a reactor, neutrons have energy ranging from 0.02 eV to 2 MeV, and their cross-sections are strongly dependent on the energy. In particular, this range is divided into a thermal region, up to 1 eV, which concerns slow or thermal neutrons, and a fast region above 1 eV, typical for neutrons having higher speeds. Another region can be also identified: the epithermal region, also known as the resonance zone. Neutrons in this range typically have energies between 1 eV and 0.7 MeV [25].

In order to express the energy dependence of the transport equation, the energy interval is divided into a certain number of discrete energy groups. In this so-called *multigroup method*, the continuous typical energy range of a thermal reactor is split into G energy groups, and each group is assigned an index $g \in$ *1, 2, 3,...,g-1, g, g+1,...,G-1,* G where $E_{g-1} > E_g$.

Thus, considering an infinitesimal volume dV of a homogeneous fissile material and integrating over the g^{th} energy group, defined within the range $E_g < E < E_{g-1}$ and over the entire neutrons directions, $[0, 4\pi)$, the different terms of the equation 2.4 can be rewritten as follow [26].

The **streaming term** can be simplified knowing the definition of angular current density and the relation between the flux and the current density, given by the Fick's law as:

$$\begin{aligned} \int_{4\pi}\int_{E_g}^{E_{g-1}} \mathbf{\Omega}\cdot\nabla\phi(\mathbf{r}, E, \mathbf{\Omega})dEd\mathbf{\Omega} &= \int_{4\pi}\int_{E_g}^{E_{g-1}} \nabla\cdot\mathbf{j}(\mathbf{r}, E, \mathbf{\Omega})dEd\mathbf{\Omega} = \\ = \int_{E_g}^{E_{g-1}} \nabla\cdot\mathbf{J}(\mathbf{r}, E)dE &= -\nabla\cdot\int_{E_g}^{E-1} D(\mathbf{r}, E)\nabla\phi(\mathbf{r}, E)dE = \\ &= -\nabla\cdot D_g(\mathbf{r})\phi_g(\mathbf{r}) \end{aligned} \tag{2.6}$$

Where $D_g(\mathbf{r})$ is the *group diffusion coefficient* and ϕ_g is the neutron flux integrated over the energy range defined by the group *g*.

Then, the **group disappearance term** represents the rate at which neutrons within the group g interact with the fissile material and disappear can be rewritten as:

$$\int_{4\pi}\int_{E_g}^{E_{g-1}} \Sigma_t(\mathbf{r}, E)\phi(\mathbf{r}, E, \mathbf{\Omega})dEd\mathbf{\Omega} = \Sigma_t^g(\mathbf{r})\phi_g(\mathbf{r}) \tag{2.7}$$

Knowing that Σ_t^g is the *macroscopic group cross-section.*

The **total in-scattering rate** is simplified recalling the definition of *macroscopic transfer group cross section*, and it has been rewritten as:

$$\int_{4\pi}\int_{E_g}^{E_{g-1}} \int dE' \oint d\mathbf{\Omega}'\Sigma_s(\mathbf{r}, E')\phi(\mathbf{r}, E', \mathbf{\Omega}')f_s(\mathbf{r}, E' \to E, \mathbf{\Omega}' \cdot \mathbf{\Omega})dEd\mathbf{\Omega} = \sum_{g'=1}^{G} \Sigma_s^{g' \to g}(\mathbf{r})\phi_{g'}(\mathbf{r}) \tag{2.8}$$

This term represents the probability that neutrons, initially belonging to other energy groups and undergoing scattering events, contribute to group g.

Considering an isotropic emission, the integration with respect to $d\mathbf{\Omega}'$ over the entire angle is equal to 4π. Finally, the fission term can be simplified as:

$$\begin{aligned} &\int_{4\pi} d\mathbf{\Omega} \int_{E_g}^{E_{g-1}} dE \frac{1}{k}\frac{\chi(\mathbf{r}, E)}{4\pi} \int dE' \oint d\mathbf{\Omega}'\nu(E')\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E', \mathbf{\Omega}') = \\ &= \frac{1}{k}\int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE'\chi(\mathbf{r}, E)\nu(E')\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E') = \frac{1}{k}\chi_g(\mathbf{r})\nu_{g'}\Sigma_f^{g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) \end{aligned} \tag{2.9}$$

Thus, the **total group fission term** is given by the contribution from all the groups into the group *g*, and so represented by the sum over the entire incident neutron energy groups *g'*.

$$... \;=\; \frac{1}{k}\chi_g(\mathbf{r}) \sum_{g'=1}^{G} \nu_{g'}\Sigma_f^{g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) \tag{2.10}$$

2.4 Computational methods for neutron transport

The neutron transport equation is a complex integro-differential equation that depends on multiple variables, including space, angle, energy, and time. Due to its high dimensionality and the presence of coupled terms, finding an exact analytical solution is, in most practical cases, either extremely difficult or entirely infeasible.

As a result, a wide range of numerical methods and computational techniques have been developed to approximate its solution with sufficient accuracy for reactor physics and shielding analysis [27].

2.4.1 Deterministic methods

The deterministic approach is based on the discretization of the continuous variables typical of the transport equation. In particular:

- the **energy** variable is discretized with the multi-group energy approximation (see 2.3);

- the **angle** discretization is achieved through different approaches such as the *discrete ordinates method*, *the spherical harmonics method*, *the collision probability method*;
- the **space** discretization is based on *the Method Of Characteristics (MOC)*, *the finite difference method* and others;
- the **time** dependency can be solved using, for example, a direct discrete method.

Normally, this type of approach has a simpler numerical calculation process and a faster convergence rate, though with a trade-off in accuracy when compared to stochastic (Monte Carlo) methods.

For the purpose of this work, this section will introduce only the Method of Characteristics, as it is the approach employed by the code used to solve the neutron transport equation.

2.4.1.1 Method of Characteristics

The Method Of Characteristics consists of a deterministic approach to solve the neutron transport equation. In particular, it is adopted in bi-dimensional calculations where a detailed representation of the core is required. One of the main advantages of this method is its ability to compromise the precision of the solution with the computational calculation time, especially when compared with the stochastic method, such as the Monte Carlo method, which, even if it guarantees a high level of accuracy of the solution, turns out to be heavier from a computational point of view [28], [29].

The base idea of the MOC is to transform the mono-energetic integro-differential neutron transport equation, which depends on the spatial position and the moving direction of neutrons, into an integral form along some defined trajectories, called characteristic lines. These lines represent the straight paths that neutrons follow within the medium, consistent with the assumption that they do not undergo deviations between two successive interactions.

Thus, starting from the stationary and mono-energetic version of the transport equation:

$$\mathbf{\Omega} \cdot \nabla\phi(\mathbf{r}, \mathbf{\Omega}) + \Sigma_t(\mathbf{r})\phi(\mathbf{r}, \mathbf{\Omega}) = S(\mathbf{r}, \mathbf{\Omega}) \tag{2.11}$$

A change of variables is introduced to parametrize the position along a characteristic line as:

$\mathbf{r}(s) = \mathbf{r}_0 + s\mathbf{\Omega}$

Where s corresponds to the traveled length along the direction $\mathbf{\Omega}$. The equation is simplified into a first order differential equation on the variable s:

$$\frac{d}{ds}\phi(s) + \Sigma_t(s)\phi(s) = S(s) \tag{2.12}$$

This formulation is more suitable for numerical treatment and enables an analytical solution to be obtained in each homogeneous subregion of the domain, under the assumption of a constant total cross section.

$$\phi(s) = \phi(0)e^{-\Sigma_t s} + \int_0^s S(s')e^{-\Sigma_t(s-s')}ds' \tag{2.13}$$

The first term in the right-hand side of the solution corresponds to the attenuated incident flux, where the exponential represents the survival probability that a neutron can travel a distance s without undergoing any interaction. Instead, the second term consists of the contribution from internal sources along the characteristic line, and the exponential term again accounts for the attenuation, considering that only a portion of neutrons produced at point s' will survive and contribute to the flux at point s.

The solution 2.13 describes the exponential attenuation of the neutron flux along the path caused by the interactions with the matter and by the source [30].

The domain has been discretized both spatially and angularly; so, for each transport direction considered, characteristic lines are drawn through the reactor geometry and the flow contribution in each traversed section is calculated. Then, once the angular flux has been found along each direction, the scalar flux can be calculated through an angular integration. This quantity represents the physical quantity typically used in the engineering application [31].

As mentioned before, this method is widely used in new generation deterministic codes, such as APOLLO3® developed at the CEA. Depending on the chosen calculation scheme, the MOC can be preferred for lattice calculations because of its ability to deal accurately with geometric inhomogeneities and present materials, especially in complex nuclear fuels such as MOX or advanced reactors, offering the possibility of integration with other calculation modules, such as those for thermal hydraulics or burn-up.

2.4.2 Monte Carlo methods

The core principle of the Monte Carlo method lies in simulating random processes to estimate quantities of interest. Specifically, when the solution to a problem involves evaluating the probability of an event or the expected value of a random variable, these can be approximated through repeated random sampling. By performing a large number of simulations, the event frequency or the average of observed values provides a numerical estimation of the desired quantity. This process guarantees a more accurate solution with respect to deterministic codes, paying with a slower convergence rate [32].

Specifically, this method can be applied to neutron transport simulations, starting from the estimation of a single neutron history in a specific geometry. Then a larger number of experimental values, also known as sampled values, may be obtained by tracking more neutron histories. In particular, three processes are required in order to solve the neutron transport problem:

- Source sampling;

- Tracking of the neutrons' energies, locations and directions;

- Result analysis.

Before starting the transport simulation of a neutron using the Monte Carlo method, it is necessary to sample its initial state from a predefined probability distribution. This initial state is defined by seven variables: three spatial coordinates, two directional angles, the neutron's energy, and the time of emission (if time dependence is considered). These variables together form the neutron phase space. Accurate sampling from this space ensures that the simulated source accurately represents the physical characteristics of the problem and allows the subsequent tracking of neutron behavior through the reactor geometry to yield meaningful results [33].

It is now necessary to simulate the neutrons' interactions. Knowing that they interact with matter after covering a certain distance, this flight distance can be described as:

$$l = -\frac{1}{\Sigma_t} ln(\xi) \tag{2.14}$$

Where Σ_t is the total cross-section and ξ represents a uniform random variable in the interval [0,1]. Based on this random sampling, both the nuclide involved in the interaction and the reaction type can be determined (e.g., elastic scattering, inelastic scattering, fission, capture).

After the reaction type is established, the post-collision properties of the neutron, specifically its energy and direction, must be determined. The outgoing energy is sampled according to the physics of the selected reaction, often based on energy probability distributions or models specific to that reaction type. The new direction of the neutron is typically sampled using the angular differential cross-section, which describes the likelihood of scattering at various angles. Additionally, conservation laws, particularly those of momentum and kinetic energy, are applied to ensure that the physical behavior of the interaction is faithfully reproduced in the simulation.

This process is repeated for each neutron and its secondary particles, allowing the Monte Carlo method to build a statistical picture of neutron behavior across the geometry and materials of the reactor system [27].

As a result, various physical quantities, such as neutron flux, dose rates, and reaction rates, can be estimated by simulating a large number of neutron histories.

2.5 Self-shielding treatment

Another important aspect to take into account in order to assure the accuracy of the results is the self-shielding phenomenon [34], [35].

As mentioned before, in the context of the multi-group formulation, the continuous energy interval can be divided into a finite number of groups, and each of them is associated with a mean and approximated value of the different cross-sections. However, this approximation may lead to some non-physical results, especially in the presence of materials with strong absorption resonances, such as ^{238}U and Thorium. In correspondence with strong resonance peaks, the absorption cross-section can vary significantly even within very narrow energy intervals. As a result, the local neutron flux in these energy regions tends to decrease due to the increased likelihood of neutron absorption, as shown in Figure 2.1. This phenomenon is known as the *energy self-shielding effect*: resonant nuclei preferentially absorb neutrons at specific energies, leading to a local depression of the energy-dependent flux in the vicinity of the resonances. Consequently, the average absorption per nucleus is reduced, since fewer neutrons are available at those energies where the cross-section is highest.

A similar concept can be spatially applied, indeed the *spatial self-shielding effect* may also occur in heterogeneous systems, particularly in reactor lattices or fuel assemblies composed of small but highly absorbing regions embedded in a moderating environment. In these configurations, the outer layers of absorbing material can attenuate the incoming neutron flux, effectively shielding the inner volume. This spatial shielding effect reduces the probability that neutrons reach the core of the material, again resulting in a lower effective absorption rate per nucleus.

If this effect is not taken into account, the approximated cross-section can cause significant errors during reactivity calculations, flux calculations, and production rates. For this reason,

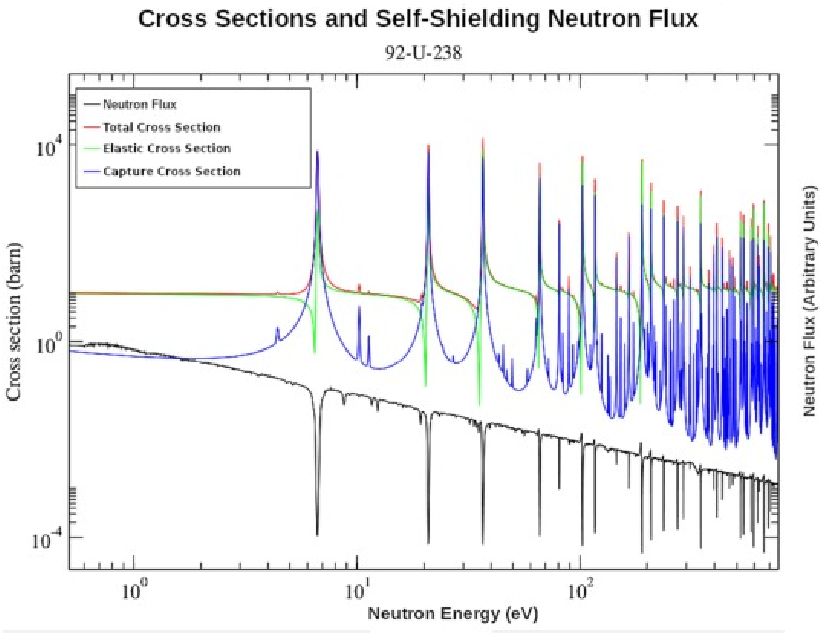


Figure 2.1: *Neutron flux in correspondence of* ^{238}U *cross sections [36]*

neutron transport deterministic codes present the possibility to implement specific models for the self-shielded cross-sections calculations, in order to obtain more realistic results. Once the self-shielded cross-sections have been computed using simplified geometries, they can be mapped onto the actual reactor configuration and employed in transport calculations, which are then solved using the Method of Characteristics (MOC) or other suitable deterministic methods ([34], [35]).

2.6 Fuel depletion and burnup calculations

After introducing the neutron transport equation, its energy discretization through the multigroup theory, and self-shielding effects, it is now appropriate to discuss one of the primary practical applications of these tools in reactor physics: the calculation of burnup.

Burnup represents the time evolution of the isotopic composition of the nuclear fuel due to neutron-induced reactions. It is commonly expressed in units of MWd/tU (megawatt days per metric ton of uranium), indicating the amount of energy extracted per unit mass of fuel. From a physical and computational point of view, burnup reflects a complex process where fissionable isotopes such as ^{235}U or ^{239}Pu undergo fission, while fertile isotopes like ^{238}U transmute into heavier actinides through neutron capture [37].

Accurate burnup calculations require knowledge of the neutron flux distribution across space and energy, which is obtained by solving the neutron transport equation using computational methods such as the Method of Characteristics (MOC). Within a multigroup formalism, the energy domain is discretized, and groupwise reaction rates are computed. In this context, *self-shielding effects* plays a fundamental role by correcting the effective cross-sections to account for the depression of the neutron flux in resonance regions, especially in heterogeneous configurations.

Once the neutron flux and reaction rates have been determined, the isotopic evolution over time is modeled using a system of coupled differential equations based on the *Bateman equations*. These equations describe the time-dependent behavior of nuclide densities by accounting for radioactive decay, neutron-induced reactions, and production terms. The transmutation and decay of nuclides within an irradiated medium are governed by the following equation [38]:

$$\frac{dN_i(t)}{dt} = \sum_j [f_{j\to i} \int_0^\infty dE\sigma_j(E)\phi(E,t) + \lambda_{j\to i}]N_j(t) - [\int_0^\infty dE\sigma_i(E)\phi(E,t) + \sum_j \lambda_{i\to j}]N_i(t) \tag{2.15}$$

where:

- $N_i(t)$ is the number density of isotope i at time t;
- $\lambda_{j\to i}$ is decay constant for decay modes in nuclide j that produce nuclide i;
- σ_i is the transmutation microscopic cross-section at energy E;
- $\phi(t)$ is the neutron flux;
- $f_{j\to j}$ is the fraction of transmutation reactions in nuclide j which produce nuclide i.

To complete the initial value problem, an initial condition is also required:

$$N_i(0) = N_{i,0} \tag{2.16}$$

These equations can be rewritten in a more compact way in matrix notation as:

$$\frac{d\mathbf{n}}{dt} = \mathbf{A}(\mathbf{n},t)\mathbf{n}, \qquad \mathbf{n}(0) = \mathbf{n}_0 \tag{2.17}$$

Where $\mathbf{n} \in \mathrm{R}^n$ corresponds to the nuclide density, $\mathbf{A(n,t)} \in \mathrm{R}^{n\times n}$ is the burn-up matrix including the decay and transmutation coefficients, and $\mathbf{n}_0$ is the initial density vector.

In conclusion, it can be noticed that the rate of change of N_i is equal to the production rate minus the loss rate.

These calculations are crucial for assessing reactor performance, optimizing fuel utilization, predicting the buildup of actinides and fission products, and evaluating long-term waste management strategies [39].

3 Codes and methods adopted in the study

3.1 Calculation method adopted

In order to carry out the different phases of the study, several codes and tools under development at CEA have been adopted. A global vision is provided in Figure 3.1[4] and more details concerning APOLLO3® , ALAMOS and TRIPOLI-4® codes are provided in the next paragraphs. As indicated in Figure 3.1, the different tools and codes are used in a sequential manner and applied to geometries of increasing complexity (from 2D fuel cell up to super cell and 3D models).

These new generation codes (mainly APOLLO3® and ALAMOS) have already been used at CEA for Fast Reactors (FR) applications [13], [8]. To simplify the workflow of neutron fast reactor conception and studies by providing user-friendly tools to technical design teams, they have been encapsulated in the INCA platform [40],[20]. However, for the present work, it has been decided to skip the use of the INCA platform privileging to work directly with the unitary codes. This choice allowed to test some new APOLLO3® features (TRIAGE, MOC3D, etc.) and to provide feedback to code's developers on the latest code versions for improving code robustness. This choice has also given the occasion to analyse/show the main options that the users need to consider when setting up a lattice neutronic calculation scheme if an domain-specific interface is not available (e.g. INCA [20], NEMESI [41], others) prioritizing education and training aspects with respect to results production.

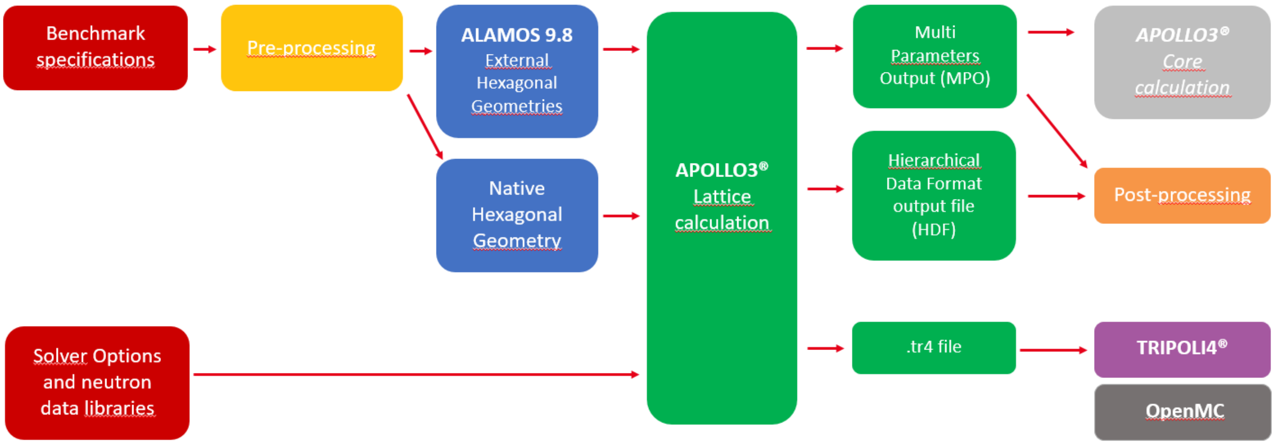


Figure 3.1: *Work path adopted during the calculations*

The work has started by collecting all necessary data for the lattice calculation. This includes data related to geometric configurations (cell, assembly, core) and their dimensions, all materials compositions and operating conditions (temperatures and densities). The data used in this work are the ones shared within the OECD/NEA community concerning Lead Fast Reactors (LFR) and Sodium Fast Reactors (SFR) neutronic benchmarks [16], [21].

In order to automatize the work, several Python scripts (pre-processing) have been prepared for converting the data provided by the benchmark (e.g. material composition) into the format accepted by the codes. An example of the Python scripts is available in Annex A.

---

[4] In the scheme: the light gray box (APOLLO3® core calculation) refers to a possible future work; the dark gray one (OpenMC) refers to results not obtained during this work.

The ALAMOS tool [42], described at Paragraph 3.2.4, has been used to prepare geometries and meshes for the different configurations investigated. When possible, the geometries were cut to take advantages of the symmetry (one-sixth and one-twelfth symmetries were adopted respectively for assembly and super cell cases). In the work, the ALAMOS Python Application Programming Interface (API) has been used. Python scripts, initially development for VVER hexagonal configurations, have been adapted to the cases considered during the internship profiting of previous experience gained during the H2020 CAMIVVER project [43], [44], [41]. Depends on the solvers used for the calculation, different mesh types may be needed for the same technological configuration (e.g. a fuel cell) [45]. The ALAMOS scripts allow this option as shown also in Annex B. In the study, Collision Probability Method (CPM) and Method of Characteristics (MOC) based on TDT (Two- and Three-Dimensional Transport) solver are used. The required unstructured meshes are provided on the basis of the ALAMOS capabilities. For the study, according to [46], the same mesh was adopted for both the self-shielding and flux calculations. On the contrary, for Light Water Reactors (LWR) cases different approaches are used (e.g. [47] as shown in presented Annex B).

In addition to the mesh required for the self-shielding and the flux calculations, output meshes need to be created. In the study, systematically two types of output meshes were created: the assembly *homogeneous* one and the *pin-by-pin* one. The two output meshes may be used for different purposes, for instance the homogeneous one is useful to collect data for core calculation (i.e. under a *coarse mesh* approach) and the pin-by-pin may be suitable for comparison against Monte Carlo fine calculations and in preparation to more refined core resolutions (i.e. based on transport solvers). The Figure 3.2 shows the meshes used for the fuel assembly case, reported as example.

Once the geometry and meshes are defined, the ALAMOS tool provides various files containing the information in the format suitable for APOLLO3® : the *.ap3* describes the unstructured geometry, the *.za* provides the Zone Assignment, i.e. allocating all materials, temperatures and properties to the each zones and the *.eqg* gives the information for building EquiGeom between different geometries, e.g. between flux mesh to output mesh. All of these files may be given as input to APOLLO3® (e.g. [41]) for creating specific APOLLO3® objects. It is also possible to provide only some of these files (e.g. .ap3) and construct the required objects adopting other functionalities of the code directly in the input file (e.g. for the EquiGeom). This second option has been used extensively in this work.

To facilitate standard calculations, APOLLO3® has been equipped of native geometries (e.g. for Cartesian lattices) which may be used without the support of ALAMOS tool. For the hexagonal shape, only the hexagonal cell can correctly represented using native geometries[5]. Therefore, for the single cell case, the two options for describing the geometry (native and ALAMOS-based) have been compared.

Once the geometry is defined, either based on external files or by native objectives, the lattice calculation steps need to be properly set up. At first, self-shielding calculation is done (several self-shielding methods have been tested, subgroup ECCO and TONE [17]) and then flux calculation is performed (in the study TDT-MOC solver was used testing different energy groups structures). Homogenization in space and energy (this last typically called also condensation) are performed and all output data (homogenized cross sections, multi-group flux, isotopic concentrations, etc.) are collected in dedicated files (*.hdf*, i.e. Hierarchical Data Format (HDF). APOLLO3® allows two types of output files: the Output, containing a limited

---

[5] Regular Hexagonal lattice are available in native geometries. However, to provide a FR fuel assembly with wrapper and inter-wrapper zones without insertion of other approximations, the use of ALAMOS tool remains necessary.

number of information but easy to manipulate for post-processing, and the Multi Parameters Output (MPO), more complex structure containing all information needed by the APOLLO3® core solvers, MINOS and MINARET [17]. The calculation sequences (included the depletion one) are presented in Annex C.

In order to consolidate the results, a verification step has been performed including comparisons against stochastic calculations. In particular, the CEA Monte Carlo code TRIPOLI-4® [48] has been used by profiting of the *TRIAGE* module of APOLLO3® available in the lattice calculation so far, that generates automatically TRIPOLI-4® input files (in format *.tr4*), simplifying the verification process. Comparisons against OpenMC calculations [38] are provided as well.

Dedicated post-processing Python scripts have been established for facilitate the results comparisons, extracting data from the different output files and automatically generate plots for a more detailed examination. Examples of the Python scripts is available in Annex A.

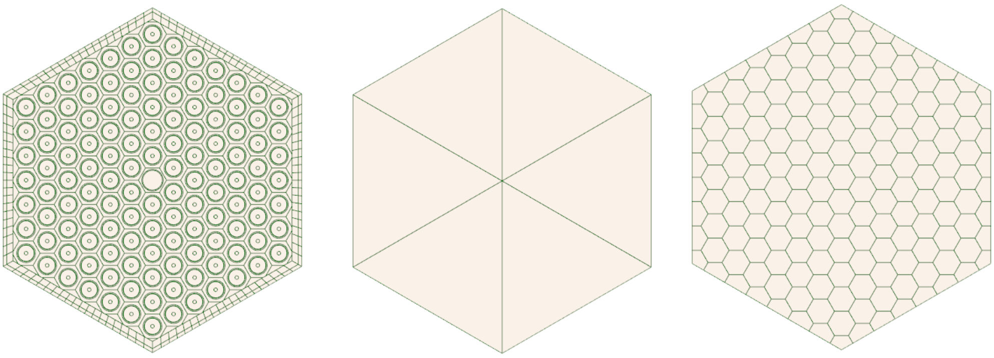


Figure 3.2: *From left to right: self-shielding and flux calculation mesh, output homogeneous and pin-by-pin meshes*

3.2 Description of the codes used

In the context of reactor physics, the analysis of the multiple interacting physical processes within a nuclear reactor has driven the development of several codes and couplings. The traditional modeling approaches based on separate physics models, where fuel performance, thermal–hydraulics, and neutronics calculations are decoupled by relying on user-imposed boundary conditions, are complemented by new solutions that take into account interdependent phenomena among the different physics [49], [50], [51], [52]. The improvement of multi-physics calculations are important for in-operation reactors but become fundamental in the development of new concepts, such as FR.

For reason of time, the present work is restricted to the analysis of a single physics, the neutronics as a first step towards coupled calculations. In order to get an idea of the interaction with other physics, typically feedback coefficients [53], [54] are evaluated under some approximations as described in Section 6.

3.2.1 The APOLLO3® code

APOLLO3® is the new French deterministic reactor code for lattice and core calculations. It is developed since 2007 by the CEA with the financial and technical support of EDF and Framatome [17].

APOLLO3® includes multi-scale (lattice/full-core) capacities and address multi-purpose uses going from for thermal reactor (with squared and hexagonal arrangements) to fast reactors, including experimental ones. It is developed for replacing previous generation codes, such as APOLLO2® /CRONOS2 for the thermal reactors and ECCO/ERANOS [**ECCO/ERANOS**] for the fast reactors [17]. In the same executable APOLLO3® includes the access to several solvers used for the typical two-step calculation approach. New features, including parallelization and domain decomposition, are developed to allow full three-dimensional pin-by-pin transport calculations at the assembly/core level for reference studies. In the work, the application of MOC3D to a LFR cell configuration [19] has been investigated (see Paragraph 5.4) for opening to new calculation approaches.

The main components of the APOLLO3® code are shown in the Figure 3.3 [17]. The code includes Neutron Boltzmann transport equation solvers at lattice, such as the Method Of Characteristic method (MOC), the collision probability method (PIJ), the Integro-Differential Transport one (IDT) and the NYMO method (also available at core level) and at core level, such as MINOS and MINARET solvers. It includes also self-shielding models for cross-sections correction. Via the MENDEL [55] component, the Bateman equations [56], equations governing the isotopic evolution of materials, are solved. The lattice and core calculations are linked by the Multi-Parameter homogenized cross-sections Output libraries (MPO). Only the features used in the work are here recalled. Other features available in APOLLO3® are detailed in [17].

In the study, the latest APOLLO3® version (V3.0) [6] has been used profiting of Python API and the keyword/value-text based interface (UDF). Both interfaces have been used at the beginning of the activity. However, the Python interface has been used for the majority of the results obtained. Via the the *TRIAGE* module [57], APOLLO3® (lattice part) has been equipped of the capability to support and facilitate the comparison with the stochastic code TRIPOLI-4® through the generation of specific files in order to automatize the verification and validation process [58].

The APOLLO3® code is still under development. Several applications have been setup in both thermal and fast reactor domains demonstrating the versatility and it contributions to accelerate advancements in nuclear research and development [17], [58], [59], [41], [60].

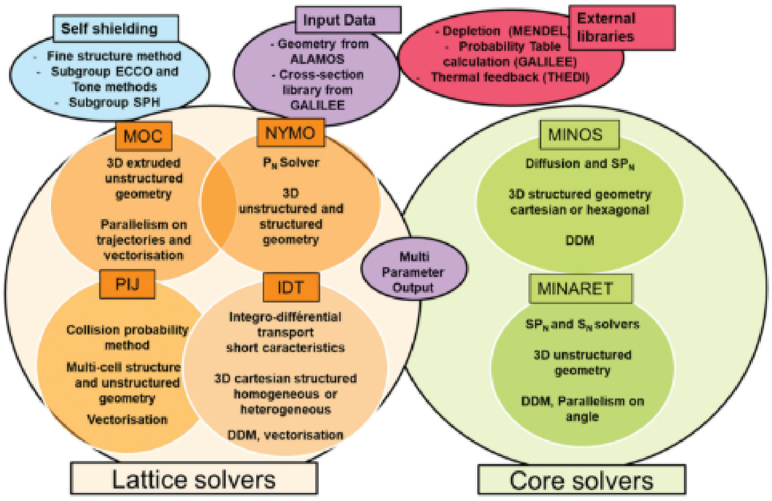


Figure 3.3: *APOLLO3® main components scheme* [17]

3.2.2 The TRIPOLI-4® code

TRIPOLI4®, developed by the CEA since the mid-1960s, utilizes the Monte Carlo stochastic method to simulate the transport of neutrons, photons, electrons, and positrons. Based on C

---

[6] During the internship period February 2024 - August 2025

and C++, TRIPOLI-4® includes its own geometry package with predefined shapes and Boolean operators. The Monte Carlo code is also compatible with various geometry software, enabling the import of more complex structures. It operates by simulating the propagation of individual particles using point-energy nuclear data, randomly evaluating their histories and interactions with matter, within a specific energy range. This approach is considered continuous in energy and exact, as it avoids the discretization approximations typical of deterministic solutions. The accuracy of the results is directly related to the number of simulated particles, which also impacts the calculation time [49].

Particles are generally organized into batches, which are simulated according to the chosen options by solving the stationary Boltzmann equation with a user-imposed source or under sterile capture conditions. The convergence criterion can be set by the user or automatically determined. The chosen options and the number of simulated batches significantly affect both the accuracy of the results and the calculation time, necessitating a balance between these aspects.

3.2.3 The OpenMC code

OpenMC is a community-developed Monte Carlo neutron and photon transport simulation code. It is capable to perform fixed source, k-eigenvalue, and sub-critical multiplication calculations. It supports both continuous-energy and multi-groups transport [61],[62].

OpenMC was originally developed by members of the Computational Reactor Physics Group at the Massachusetts Institute of Technology starting in 2011. The aim was to cope with legacy code providing from scratch a Monte Carlo code with a focus on high performance scalable algorithms as well as modern software design practices [62].

In the work, it has been used as internal CEA comparison for supporting the analysis of the results obtained and the cross-checking with the ones provided by the other NEA benchmark participants.

3.2.4 Tool for geometries and meshes generation: ALAMOS

The deterministic and the stochastic codes often require the support of specialized tools to construct complex geometries. Dedicated tools to assist the user with the preparation of input data have been developed at CEA since CPM and MOC capabilities by the solver TDT have been made available [42].

In particular, for the purposes of this work ALAMOS tool (version 9.8) has been used. ALAMOS has been developed as a module of SALOME platform [63], profiting the access to numerous Python libraries, which enhance its functionality and flexibility. It takes advantage of MEDCoupling module, a library specialized in operations with fields defined upon unstructured geometries, representing scalar, vector and tensor quantities (among other rich functionalities) [64] .

In general, the geometry is built starting from points and lines and can be either structured or unstructured. Structured geometry uses basic shapes defined by equations with user-editable coefficients. In contrast, unstructured geometry is represented by a collection of points and lines, therefore giving access to general geometry descriptions.

ALAMOS offers a graphical user interface Graphical User Interface (GUI) and a Python API; it works with object layer consisting of a geometry to which regions specific properties and references are associated. Additionally, ALAMOS introduces the concept of *metacell* that denotes the different regions of the geometry delimited by lines and points. At the end, it is possible to insert a specific layer inside an existing one through a process called dereferencing.

In the work, the ALAMOS Python API has been used. Python scripts, initially development for VVER hexagonal configurations, have been adapted to case study profiting of previous experience gained during the H2020 CAMIVVER project [43], [44], [41]. These scripts already allow the definition of three different types of meshes at assembly level: one for the self-shielding calculation, one for the flux calculation, and one for output homogenized data. The scripts have been updated for representing all the geometries (from cell to super cell) described in the study in order to make possible the calculations performed in this work. The used Python script may be generalized and standardized as already demonstrated by the work carried out at [44], [65]. However, this standardization process has been considered out of the scope of the current work.

4 The case study: OECD/NEA LFR Benchmark

In order to apply the codes described in Chapter 3, the OECD/NEA LFR benchmark has been selected giving the possibility at the same time of accessing to comparison against other solutions provided by the participants [16]. Other test cases have been added for completing the analysis (see details at Paragraph 6.2 and 6.3). The cases considered are an industrial size Sodium Fast Reactor (SFR) adopted in a different OECD/NEA benchmark [21] and a numerical case, OLFR, proposed for carrying-out out several investigations concerning multi-physics and fuel cycle analysis [66], [60].

4.1 LFR benchmark description and objectives

Within the OECD/NEA Working Party on Scientific Issues and Uncertainty Analysis of Reactor Systems (WPRS) the Expert Group on Physics of Reactor Systems (EGPRS) [67] is in charge of providing expert opinion to the nuclear community regarding the development needed in the specific domain and beyond.

The EGPRS group is proposing and following several benchmark exercises covering different types of reactor concepts: light-water cooled, molten-salt, sodium-cooled, gas-cooled and lead-cooled fast reactor [68]. This latest case has been selected as reference case for this work.

The LFR configuration corresponds to the ALFRED reactor [69], [70]. The thesis work has been organized following the phases proposed for the benchmark: 1) the pin-cell simulation, 2) the assembly simulation, and 3) the whole core simulation. Each step requires specific results, which are detailed in the coming sections and the majority of them have been accomplished. Additional parameters have been studied to complement the study. For sake of time, the phase three (3D full core) has not been considered.

4.2 Reactor description

ALFRED is a pool-type fast spectrum reactor of 300 MWth, cooled with pure lead within a temperature range between 400 °C and 520 °C, referring to the inlet and the outlet regions, respectively in nominal conditions.

Figure 4.1 presents the core flow path and the core loading structure. The full core is composed by 253 Sub-Assemblies (S/As) arranged in an hexagonal lattice. All S/As include a wrapper made of AIM1 steel. Table 4.1 gives detail for each type [16].

4.2.1 Fuel Assemblies

The fuel assembly is composed by pellets of mixed uranium and plutonium oxide (MOX). Two different plutonium contents are considered in order to achieve a better homogenization of the power distribution within the core: 20.5 wt% for the inner fuel and 26.2 wt% for outer fuel. The fuel assembly is composed by 126 fuel pins arranged in a triangular lattice around a central guide filled by helium at 1 atm, meant for special instrumentation aimed to post-irradiation examination [16]. In the Figure 4.2 are shown the axial and radial sections of the fuel assembly.

Focusing on the single fuel rod, the corresponding cuts are shown in the Figure 4.3. The central hole of the pin and the gap are both filled by helium at 1 atm; the austenitic stainless steel AIM1 is adopted for the cladding such as for the other structural components.

Above and below the active zone of the fuel rod a plenum region and an insulator are located. This last is made of Yttria-Stabilized Zirconia (YSZ), a mixed ceramic material of Y_2O_3 and

ZrO_2 [16].

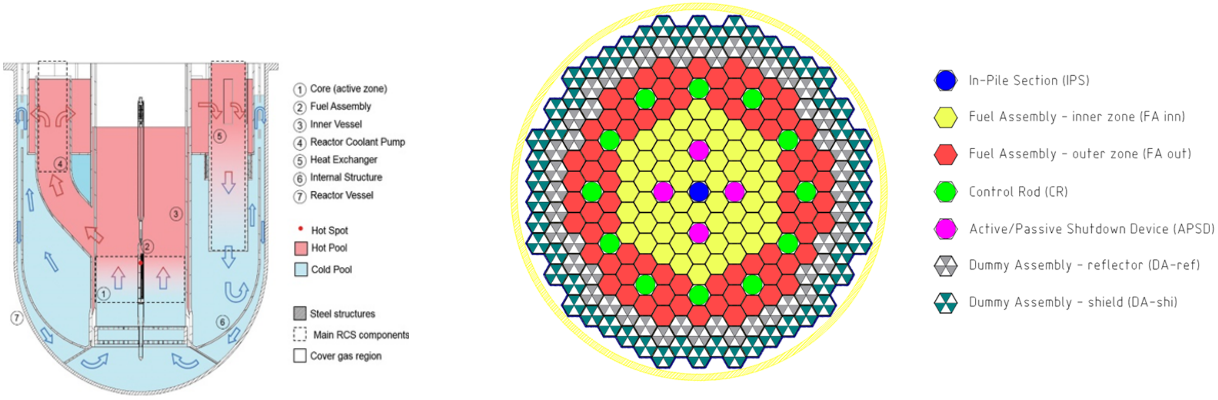


Figure 4.1: *Lead flow illustration on the left* [69] *and ALFRED core arrangement* [71]

Table 4.1: *Number of Sub-Assemblies in the reactor*

Type	Quantity
Inner Fuel Assemblies (FAs)	56
Outer Fuel Assemblies (FAs)	78
Control Rods (CRs)	12
Safety Devices (SDs)	4
Test Assembly (TA)	1
Reflector Assemblies (RAs)	48
Shield Assemblies (SAs)	54

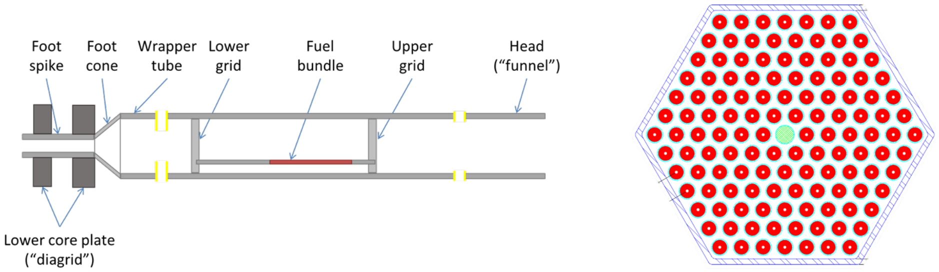


Figure 4.2: *Axial and radial view of the fuel assembly* [16]

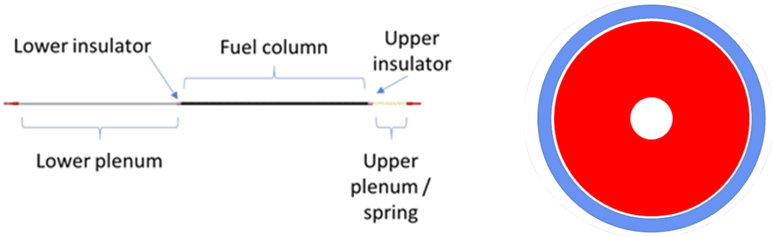


Figure 4.3: *Axial and radial view of the fuel rod* [16]

4.2.2 Control Rods and Safety Devices

The Control Rods and Safety Devices are meant to be inserted in the core from the bottom.

In particular, the Control Rods are manufactured as shown in Figure 4.4, with 31 pins arranged in a triangular lattice. The pins are made of a boron carbide absorber material with two different 10^B enrichment for the lower (90% 10^B enr.) and the upper (42% 10^B enr.) parts. During normal operation, only the upper part is inserted in the core and progressively removed from the core until complete extraction is obtained at the End of Cycle (EoC). The bottom part with an higher 10^B enrichment is normally out of the core and it is inserted only for shutdown purposes.

The Safety Devices have the same geometry as the Control Rods and they are composed by boron carbide at 90% 10^B enr. The Safety Devices are completely extracted during normal operations [16].

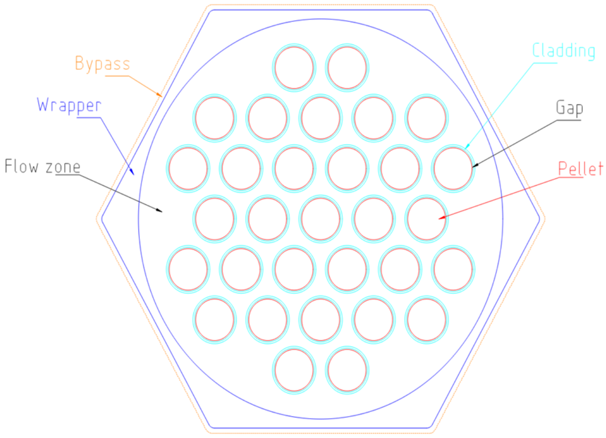


Figure 4.4: *Control Rods Sub-Assembly arrangement* [16]

4.2.3 Reflector Assemblies, Shield Assemblies and Test Assembly

The core is composed by other type of S/As. Test Assembly is placed in the center of the reactor aiming to accommodate in-pile irradiation experiments. The Reflector Assemblies and the Shield ones are located at the core periphery and they are composed by YSZ pellets [16].

4.3 Assumptions and requested results

In the benchrmark, in order to facilitate the comparison among participants, several assumptions have been defined as well as a list of required results.

All the calculation are carried out with the densities referring to *cold conditions* (@ 20 °C) with exception of lead density that is considered at 460 °C (density of 10.503 g/cm3), corresponding to the temperature at middle core height. The temperatures are fixed to 1200 K for the fuel and 600 K for all the other materials, including lead.

Concerning the neutron data library, the benchmark does not request a specific evaluation. The calculation in this work have been performed mainly using JEFF3.1.1 library [72]. In addition, for facilitating comparisons with other codes also JEFF3.3 library has been adopted

[73]. Concerning the outputs, the data are provided condensed at 1, 33 and 172 energy group structures.

Reflective boundary conditions are used for the 2D cases[7].

Between the results proposed by the benchmark, the following ones have been evaluated:

- k_∞;
- 1-group microscopic fission and capture cross sections for ^{239}Pu, ^{241}Pu, ^{235}U and ^{238}U and capture cross sections for ^{56}Fe, ^{208}Pb and ^{10}B[8];
- number of neutrons emitted by the fission for ^{239}Pu, ^{241}Pu, ^{235}U and ^{238}U;
- neutron spectrum at different energy groups structures;
- the utilization factor (Eq. 4.1) and the reproduction factor (Eq. 4.2).

$$f = \frac{\Sigma_{a,fuel}}{\Sigma_{a,tot}} \tag{4.1}$$

$$\eta = \frac{\nu\Sigma_f}{\Sigma_{a,fuel}} \tag{4.2}$$

In the analysis of the fuel pin and fuel assembly, an alternative homogenized model has been also investigated. In this model, the gap and the central hole are smeared with the fuel adjusting correctly the fuel density to maintain the total mass. All the configurations investigated are illustrated in the Figures 4.5, 4.6 and 4.7 respectively for the cell, the assembly and the super cell cases.

Figure 4.5: *Heterogeneous fuel cell model on the left and the homogeneous one right*

---

[7] Concerning the boundary conditions in the pin-cell and the Sub-Assembly cases, these have been established according to the symmetry of each configuration. For simulations requiring the complete geometry, the appropriate boundary conditions to be used in APOLLO3®is translation, instead in case of cut geometry is specular type.

[8] ^{10}B only in case of the presence of the absorber.

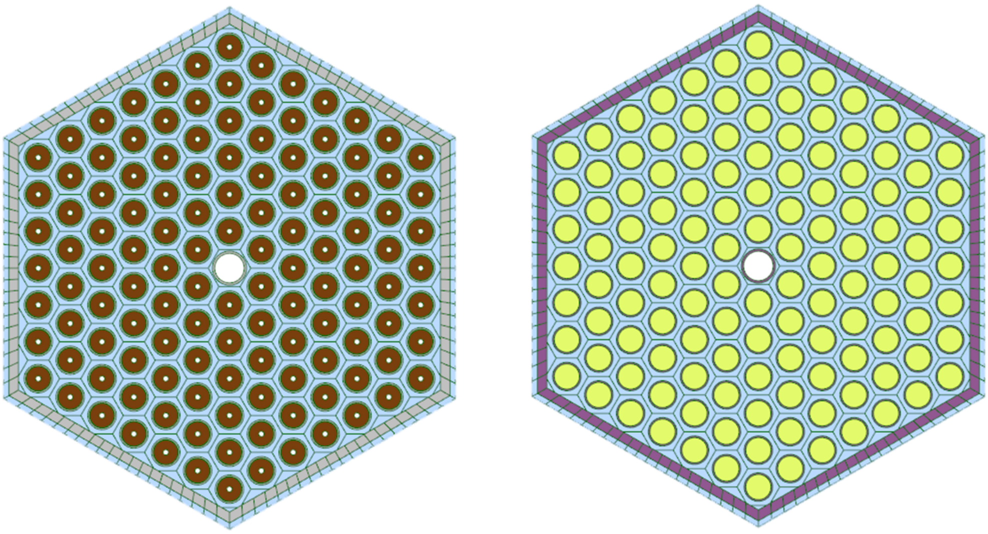


Figure 4.6: *Heterogeneous fuel assembly model on the left and the homogeneous one on the right*

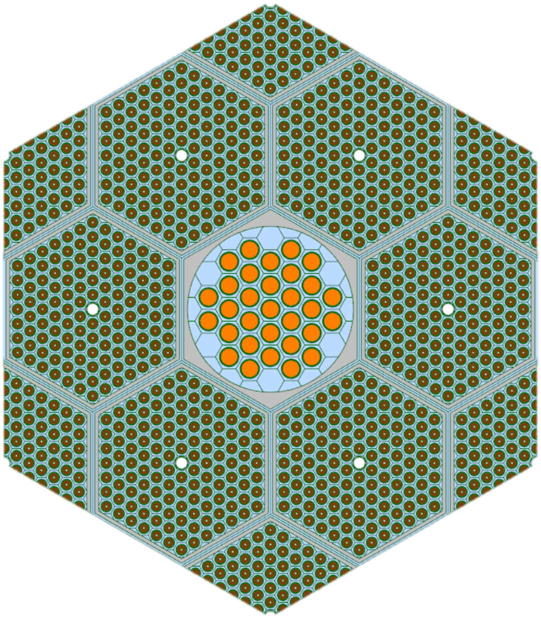


Figure 4.7: *Control rods super cell configuration*

5 Main results obtained

In this section, the main results obtained are presented following the phases proposed by the OECD/NEA benchmark. Complementary results are available in D.

When possible, the results obtained through the deterministic code were compared either with Monte Carlo solutions provided by TRIPOLI4®, either with other solutions available at CEA[9] or provided by the benchmark participants[10].

5.1 Analysis of the cell configuration

In this section the main relevant results concerning the single fuel cell are analyzed. In particular, the structure has been studied in an infinite medium by setting the boundary conditions as *SPECULAR* and using a fuel plutonium content equal to the inner one (20.5 wt%).

Two approaches for geometry definition has been tested: native geometry and ALAMOS-based geometry. The modifications required in the APOLLO3® Python input are shown in Annex D.1.1. As expected, the two configurations show a very good agreement (discrepancy in terms of reactivity, of about 2 pcm[11]). This test has allowed to improve the original ALAMOS scripts by adding the ability to obtain a cell geometry with one-twelfth symmetry.

Several options have been tested (neutron data library, self-shielding models, mesh refinements) as illustrated in the Table 5.1. The reactivity variation is calculated using the Eq. 5.1, where $k_{\infty,2}$ is assumed to be the case corresponding to JEFF3.1.1 library at 1968g energy structure and SgECCO self-shieldiing method. The cases shown in Table 5.1 are all based on native geometry (one-twelfth symmetry).

Table 5.1: *2D fuel cell: k_∞ comparison*

	APOLLO3®					TR4®	OpenMC
Library	JEFF3.1.1			JEFF3.3		JEFF3.1.1	JEFF3.3
SSH model	SgECCO	TONE			SgECCO	-	-
Flux mesh	-	-	refined	-	-	-	-
E groups	1968			1760		cont.	cont.
k_∞	1.35890	1.35878	1.35878	1.36068	1.36079	1.35614[12]	1.36158[13]
$\Delta\rho$ [pcm]	-	-6	-6	96	102	-150	145
Time[14] (s)	98	54	53	74	113	∼8 h	∼20 h

$$\Delta\rho = \frac{k_{\infty,1} - k_{\infty,2}}{k_{\infty,1}k_{\infty,2}} \tag{5.1}$$

---

[9] OpenMC has been used for internal R&D analyses.

[10] Within the 3rd LFR Benchmark Workshop, Lucca, 20 May 2024, the benchmark participants have provided several results based on OpenMC/Serpent and other codes. The results obtained in this work has been compared with the results presented by the participants but this comparison may not be reported in this document. However, the comparison has been used for comforting in the results obtained.

[11] The precision on the eigenvalue calculation is fixed by default at 1 pcm.

[12] With a standard deviation 23 pcm - all tallies are in agreement with APOLLO3®output required.

[13] With a standard deviation of 3 pcm, using 50000 batches, 800 inactive, 10000 particles/batch.

[14] The calculations times are indicatives, a refined study has not been performed. The calculations have been run on Machine Intel5R° Xenon5R° CPU E5-2630 v4 2.20GHz.

From Table 5.1, it is possible to identify that the self-shielding models SgECCO and TONE are almost equivalent (6 pcm difference). On the other hand, TONE provides a gain in calculation time between 30 and 40%. The effect of nuclear data library on k_∞ is about 100 pcm. In addition, thanks to the OpenMC input, created for facilitating comparisons with the other benchmark participants, also the JEFF3.3 results were confirmed (40 pcm difference)[15]. The results are in agreement with the ones presented by the NEA participants at the 3rd LFR workshop [74].

The Figures 5.1 and 5.2 present, respectively, the whole cell spectrum based on 33 and 172 energy group structures using log-log and lin-log scales. In the two cases, generated automatically by the post-processing scripts shown in Annex A, the y-axis shows the flux normalized on the total flux and the x-axis the energy expressed in MeV. As expected, the tendency is typical of fast reactors with a peak around 100 keV and a negligible amount of neutrons in the thermal zone.

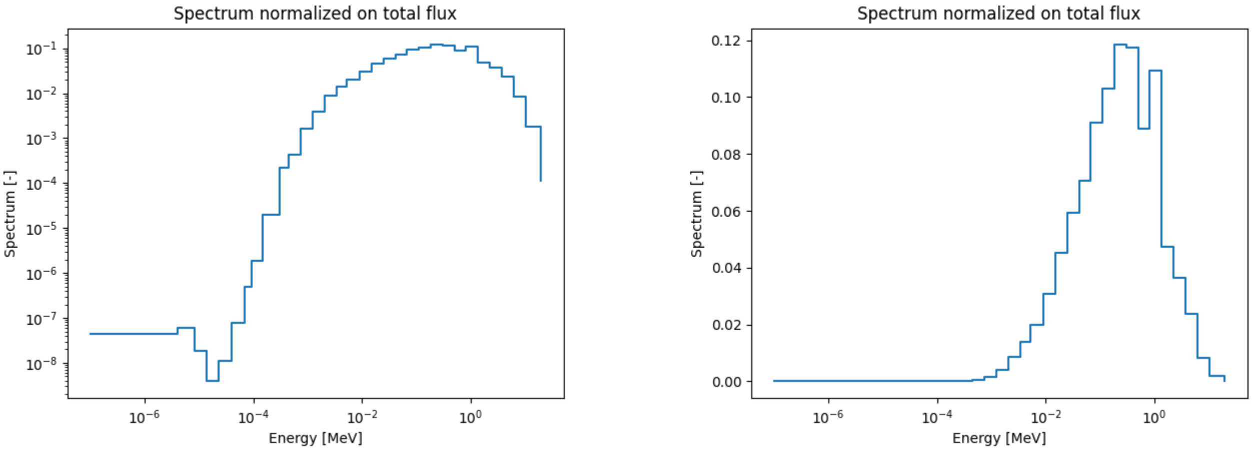


Figure 5.1: *33 energy group structure: whole cell neutron spectrum*

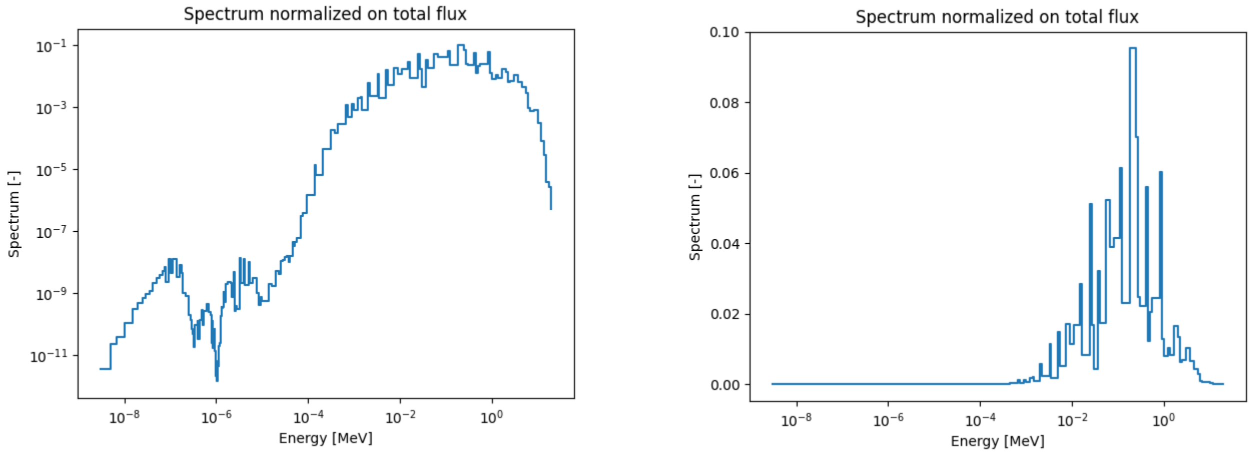


Figure 5.2: *172 energy group structure: whole cell neutron spectrum*

The results requested by the OECD/NEA benchmark (Paragraph 4.3) have been evaluated. In the following tables (Table 5.2 and Table 5.3) the results obtained using JEFF3.1.1 (1968 energy

---

[15] The calculation time for stochastic simulations are quite important probably due to the tallies required for these cases. More investigations have to be done for confirming these analysis.

groups structure) and JEFF3.3 (1760 energy groups structure) libraries are presented. The discrepancy between the two adopted libraries is negligible and the main difference refers to the lead microscopic cross-section (see Annex D.1.2). The obtained results were then compared and confirmed by the ones obtained by the other benchmark participants [74]. The parameters not directly accessible to the calculation are the capture cross-section (evaluated as the difference between the absorption and the fission cross-section) and the number of neutrons emitted by fission ($NuBar$) (evaluated as to the fraction of the $NuFission$ parameter and the fission cross-section).

Table 5.2: *Microscopic cross-sections calculated in barn (b)*

Isotope	Fission [b]		Absorption [b]		Capture [b]	
	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3
239Pu	1.71724	1.73732	2.16383	2.15695	0.44659	0.41963
241Pu	2.41513	2.39394	2.86003	2.83357	0.44491	0.43963
235U	1.81785	1.83180	2.32338	2.38424	0.50553	0.55244
238U	0.03590	0.03638	0.30629	0.30115	0.27039	0.26477
56Fe	-	-	0.00702	0.00709	0.00702	0.00709
208Pb	-	-	0.00069	0.00045	0.00069	0.00045

Table 5.3: *Number of neutrons emitted by fission*

Isotope	NuFission [b]		NuBar [-]	
	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3
239Pu	5.04921	5.06320	2.94031	2.91437
241Pu	7.17936	7.13912	2.97266	2.98216
235U	4.46574	4.49728	2.45660	2.45512
238U	0.09918	0.09979	2.76307	2.74300

In particular, the self-shielding methods have not impact in the neutron spectrum while different data library are slightly affecting the thermal region as indicated in Figure 5.3.

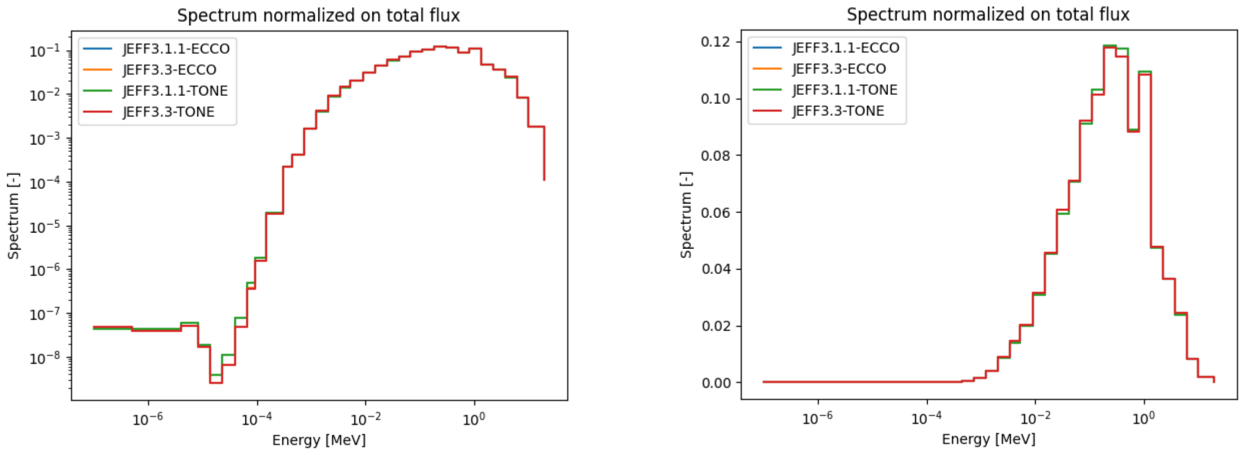


Figure 5.3: *Spectrum comparison at 33 energy group structure*

The criticality factors are given in Table 5.4. They were calculated according to Eqs. 4.1 4.2 using macroscopic cross-sections obtained directly from the output files provided by APOLLO3® . Also in this case the results are in agreement with the benchmark participants[74]. The effect due by the data library is around 0.5%.

Table 5.4: *Cell model criticality factors*

Criticality Factor	Value [-]	
	JEFF3.1.1	JEFF3.3
f	0.91417	0.90919
η	1.40748	1.41215

As mentioned in the Subsection 4.3, a case in which fuel gap and central hole are smeared within the fuel has been investigated. The k_∞ calculated using JEFF3.1.1 library is equal to 1.35869 corresponding to a difference of less that 2 pcm. The smeared case does not show a particular gain in term of computation time with respect to the heterogeneous case (108 s vs. 103 s). The Figure of Merits (FOM) are in agreement with the heterogeneous case; the highest relative error refers to the f parameter, about 1%, due to a discrepancy on the fuel macroscopic cross-section of about 10%. This larger difference comes from the different structure of the fuel pin.

5.2 Analysis of the assembly configuration

This section presents the main results obtained for the fuel assembly configuration. For this specific case, the fuel assembly geometry for APOLLO3® has been built with the support of ALAMOS tool (see Paragraph 3.2.4). One-sixth symmetry has been considered for the calculations as shown in the Figure 5.4, since initially the ALAMOS Python scripts do not include a one-twelfth symmetry. In the late phase of the study, those scripts have been updated (see Paragraph 5.3) and calculations with one-twelfth symmetry repeated (see Figure 5.5).

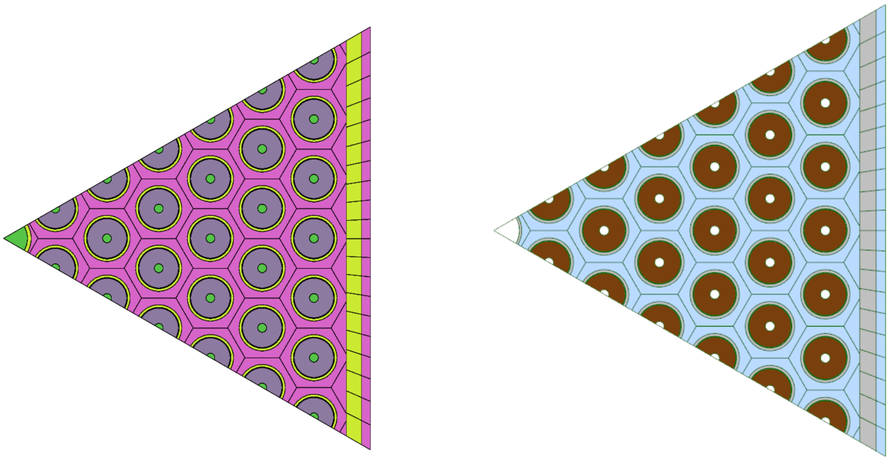


Figure 5.4: *One-sixth fuel assembly symmetry: APOLLO3® and ALAMOS representations*

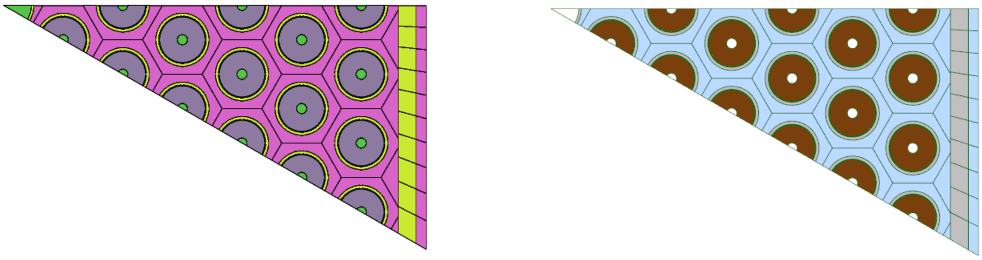


Figure 5.5: *One-twelfth fuel assembly symmetry: APOLLO3® and ALAMOS representations*

As for the cell case, JEFF3.1.1 and JEFF3.3 data libraries have been considered. Table 5.5 collects the results obtained.

Table 5.5: *2D fuel assembly: k_∞ comparison*

	APOLLO3®				TR4®	OpenMC
Library	JEFF3.1.1	JEFF3.3			JEFF3.1.1	JEFF3.3
SSH model	SgECCO		TONE		-	-
Symmetry	1/6th		1/6th	1/12th		full
E groups	1968	1760			cont.	cont.
k_∞	1.29216	1.29193	1.29181	1.29169	1.29015[16]	1.29436[17]
$\Delta\rho$ [pcm]	-	-14	-21	-28	-121	132
Time (s)[18]	1665	1579	169	123	~6 h	3313

As shown in Table 5.5, the self-shielding method does not impact significantly the calculations, with a difference of 12 pcm but it allows to gain in terms of computation time. Indeed, the TONE method is ten times faster than SgECCO method. Also in this case the deterministic results were confirmed by the stochastic simulations and by participants results [74].

Since they do not impact the k_∞ (a difference equal to 13 pcm), the solver option used in the fuel assembly configuration are chosen in order to gain in terms of computation time. Indeed, the use of the cell solver options increases the time of the calculation by a factor of three with SgECCO (4794 s) and by two in case of TONE (226 s). This difference is linked to the tracking options during the TDT calculation (see Annex D.2.1).

Tables 5.6, 5.7 and 5.8 show the results obtained for the fuel assembly case. As expected, the results are in agreement with the ones obtained for the cell case.

Even in terms of spectrum, the behaviour between the cell and fuel assembly configurations is almost the same as shown in Figure 5.6. The main difference is in the range between [0.01 - 1] keV. In this region, the fuel assembly presents a higher spectrum due to the presence of more structural materials, which tends to switch the spectrum toward the thermal zone. Indeed, in this zone, due to the increase of the absorption reactions in the fuel assembly configuration, the reproduction factor (η) is lower as shown in Figure 5.7.

---

[16] With a standard deviation of 23 pcm - all tallies are conformed to APOLLO3®.

[17] With a standard deviation of 13 pcm with 20000 batches, 800 inactive, 20000 particles/batch.

[18] The calculations times are indicatives, a refined study has not been performed. The calculations have been run on Machine Intel(R) Xeon(R) CPU E5-2630 v4 2.20GHz.

Table 5.6: *Microscopic cross-sections calculated in barn (b)*

Isotope	Fission [b]		Absorption [b]		Capture [b]	
	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3
239Pu	1.75643	1.77372	2.24983	2.23759	0.49340	0.46387
241Pu	2.50758	2.48032	2.98204	2.94726	0.47446	0.46694
235U	1.89231	1.90463	2.43651	2.49047	0.54421	0.58584
238U	0.03303	0.03350	0.31803	0.31236	0.28500	0.27885
56Fe	-	-	0.00705	0.00708	0.00705	0.00708
208Pb	-	-	0.00068	0.00044	0.00068	0.00044

Table 5.7: *Number of neutrons emitted by fission*

Isotope	NuFission [b]		NuBar [-]	
	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3
239Pu	5.15713	5.16188	2.93614	2.91019
241Pu	7.44599	7.38990	2.96940	2.97941
235U	4.64453	4.67139	2.45442	2.45265
238U	0.09115	0.09176	2.75928	2.73890

Table 5.8: *Fuel assembly model criticality factors*

Criticality Factor	Value [-]	
	JEFF3.1.1	JEFF3.3
f	0.91611	0.91055
η	1.36782	1.37276

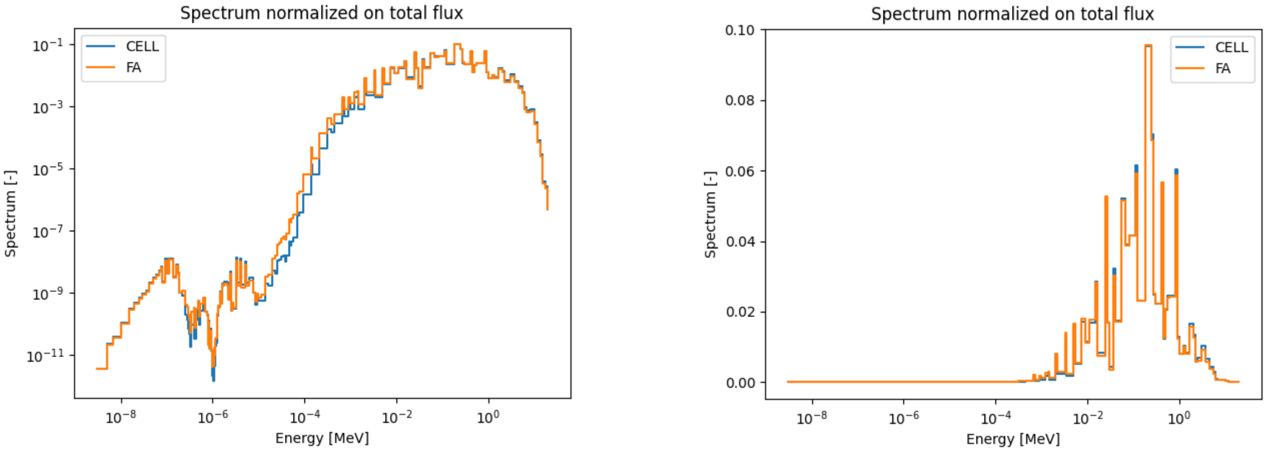


Figure 5.6: *Cell and fuel assembly spectrum at 172 groups comparison*

Also for the assembly case, the fuel gap and central hole have been smeared within the fuel (homogeneous configuration). The corresponding k_∞ calculated using JEFF3.1.1 library is equal to 1.29163 with a difference respect to the heterogeneous case of 15 pcm. The difference is larger than the cell case and it may be due to the different mesh of the internal fuel zone (see

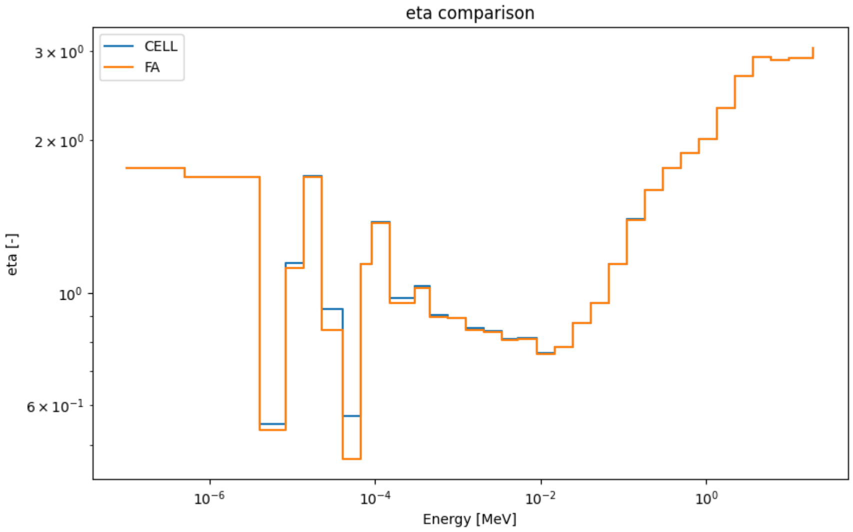


Figure 5.7: *Comparison between η depending on energy*

Annex D.2.2). The FOM required by the benchmark have been evaluated for the homogeneous case as well and they are in agreement with the heterogeneous case; even in this case the larger discrepancy corresponds to the *utilization factor* showing the same order of magnitude as the cell case.

5.3 Analysis of the super cell configuration

This section presents the main results obtained for the super cell configuration. The super cell corresponds to a assembly lattice where the central position is occupied by a control rod assembly, as shown in Figure 4.7. The case involving a central reflector in the super cell was not considered in this study.

A step-by-step approach has been adopted for setting up the model. At first, the super cell configuration has been investigated inserting at the central position a standard fuel assembly (as shown in Figure 5.8). This case has been compared with the fuel assembly stand-alone model 5.2. The obtained k_∞ is equal to 1.29249 (9 pcm different from the case fuel assembly, see Table 5.5). Minor difference may be due to the external surface. Anyway, this check has allowed to confirm that the ALAMOS Python scripts for generating the super cell geometry were correct.

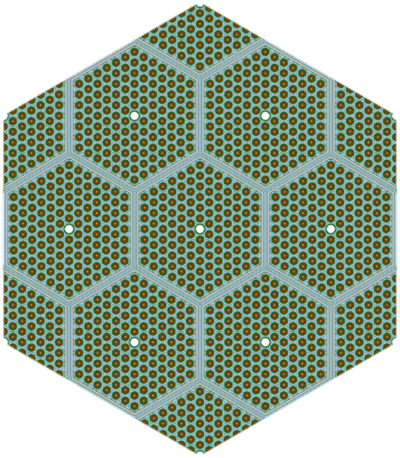


Figure 5.8: *Super cell structure composed by fuel assemblies*

Once done, the analysis of the presence of absorber material has been carried out assuming an enrichment in ^{10}B of 42%, which corresponds to the upper part of the control rod zone. To reduce the computational time, a one-twelfth symmetry geometry was adopted and so the ALAMOS scripts properly modified. A finer mesh was implemented in the boron pins to take into account correct self-shielding effect. This fine mesh was crucial for accurately evaluating the flux depression in the absorber zone as also indicated in [46].

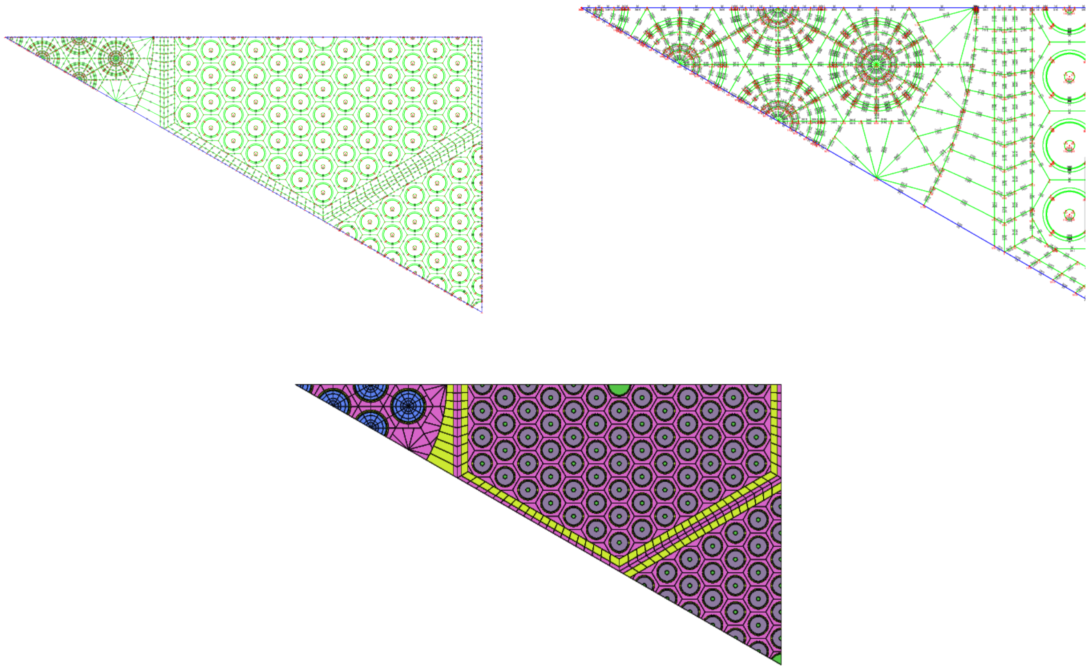


Figure 5.9: *Reference super cell finer mesh and symmetry geometry*

As expected, the reactivity of the system decreases significantly with the presence of absorber material in the center of the super cell. The k_∞ of the super cell configuration (Figure 4.7) is equal to 1.05219 (JEFF 3.1.1 data library), corresponding to a discrepancy of 17650 pcm relative to the fuel assembly case.

Figure 5.10 illustrates the relatively high absorption cross section, especially in the thermal region, in the boron carbide zone and in the fuel zone. The absorption cross-section decreases with the energy but remains always above the fuel one.

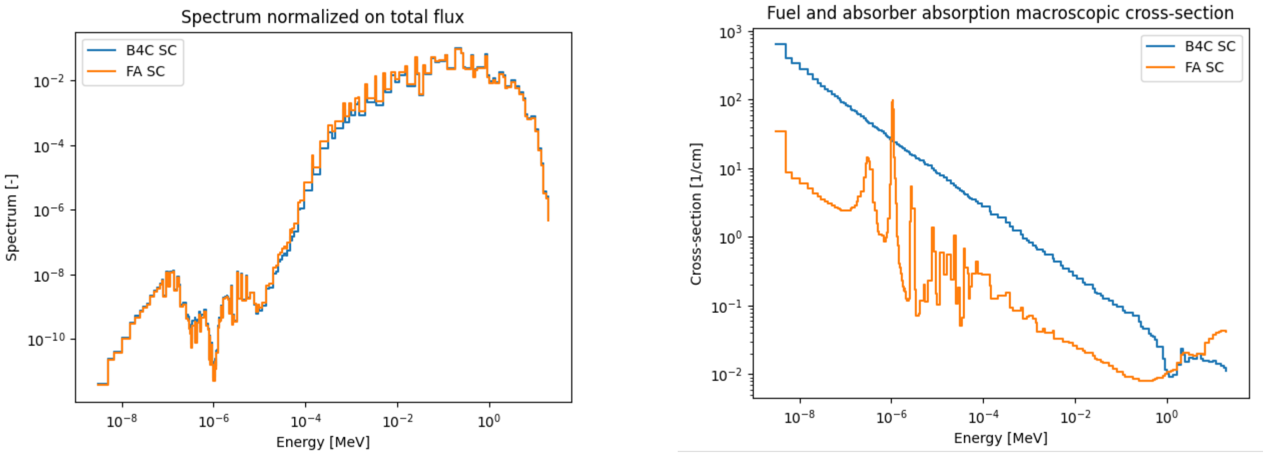


Figure 5.10: *Fuel assembly and boron super cell comparison in terms of spectrum and macroscopic absoprtion cross-section*

An higher concentration of the isotope ^{10}B significantly increases the neutron absorption in the reactor, even in the fast regions zone. In particular, the configuration with 42% of ^{10}B enrichment presents an absorption microscopic cross-section equal to 1.887 barn, instead, the case with 92% shows a lower value equal to 1.505 barn, as a result of the spatial self-shielding effect [75]. Instead, the macroscopic absorber cross-section increases with the enrichment as it is presented in the Figure 5.11.

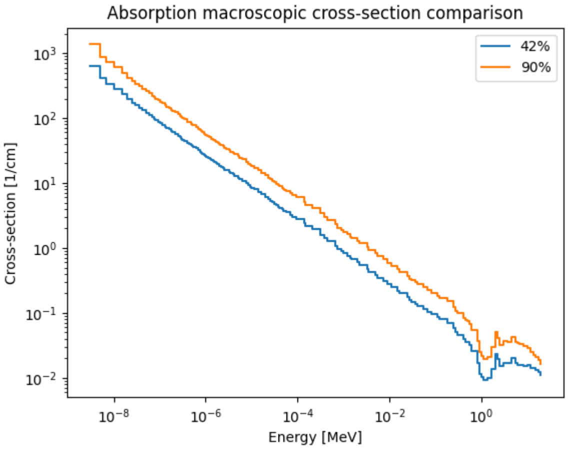


Figure 5.11: *Absorber macroscopic cross-section comparison at different enrichment*

Several tests have been performed as shown in Table 5.9 which confirmed the above considerations. The effects of the libraries and of the self-shielding calculation models are negligible (43 pcm and 3 pcm respectively) and also in terms of adopted. In particular, as in the previous cases, TONE model permits to reduce the calculation time more then 10 times.

In Tables 5.10, 5.11 and 5.12 are collecting the main results concerning the control rods super cell model having a ^{10}B enrichment equal to 42%. The obtained values are in line with the cell and fuel assembly cases, in terms of discrepancies and order of magnitude with the partners results [74] and with the results reported in a similar study conducted in [76].

Table 5.9: *2D super cell: k_∞ comparison*

	APOLLO3®			TR4®	OpenMC
Library	JEFF3.1.1		JEFF3.3	JEFF3.1.1	JEFF3.3
SSH model	SgECCO	TONE		-	-
Symmetry	one-twelve				full
E groups	1968	1760		cont.	
k_∞	1.05219	1.05216	1.05168	1.05081	1.05469
$\Delta\rho$ [pcm]	-	-3	-46	-125	225
Time (s)[19]		637	574	~5.5 h	2927

---

[19] The calculations times are indicatives, a refined study has not been performed. The calculations have been run on Machine Intel(R) Xeon(R) CPU E5-2630 v4 2.20GHz.

Table 5.10: *Microscopic cross-sections calculated in barn (b)*

Isotope	Fission [b]		Absorption [b]		Capture [b]	
	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3
239Pu	1.71796	1.73501	2.14154	2.13260	0.42358	0.39759
241Pu	2.36450	2.34153	2.79301	2.76059	0.42851	0.41906
235U	1.77732	1.78754	2.26119	2.31156	0.48386	0.52402
238U	0.03844	0.03900	0.29650	0.29069	0.25805	0.25169
56Fe	-	-	0.00643	0.00640	0.00643	0.00640
208Pb	-	-	0.00073	0.00046	0.00073	0.00046
10B	-	-	1.88665	1.89487	1.88665	1.89487

Table 5.11: *Number of neutrons emitted by fission*

Isotope	NuFission [b]		NuBar [-]	
	JEFF3.1.1	JEFF3.3	JEFF3.1.1	JEFF3.3
239Pu	5.05818	5.06404	2.94429	2.91873
241Pu	7.03768	6.99053	2.97640	2.98545
235U	4.37107	4.39366	2.45935	2.45793
238U	0.10614	0.10688	2.76081	2.74043

Table 5.12: *Control rods super cell model criticality factors*

Criticality Factor	Value [-]	
	JEFF3.1.1	JEFF3.3
f	0.91780	0.91337
η	1.45182	1.45772

5.4 Model at core level

The OECD/NEA benchmark is characterized by a 3rd phase dedicated to 3D core calculations. For reasons of time, this phase has not been investigated giving more attention to the consolidation of the results at lattice level. The calculations performed and presented at Paragraphs 5.1, 5.2 and 5.3 allow already to generate output files in the MPO format that may be read at core level. Further analyses are expected as present activity follow up. Among those the application of the INCA interface [20] will be considered as well.

However, in order to start to investigate the 3D effect, a 3D pin structure has been set up profiting of the new capabilities of APOLLO3® which give access to MOC3D solver [19], [77], [28]. The preliminary results are indicated here just as an example of possible application of the 3D model, thus, further investigations are foreseen in a following phase of the study.

The 3D cell configuration is composed by several axial zones representing the axial core layout of the LFR case [16], including fuel active part and lower and upper zones[20]. A simplified sketch is presented in Figure 5.12. On the basis of this configurations several tests have been performed using the latest APOLLO3® versions.

Some discrepancies remain with respect to Monte Carlo calculations. Further studies will help to investigate more in detail the reasons of those discrepancies with the support of the developers. Some analysis concerning the 3D coolant density variation have been performed and recalled at Annex D.3.

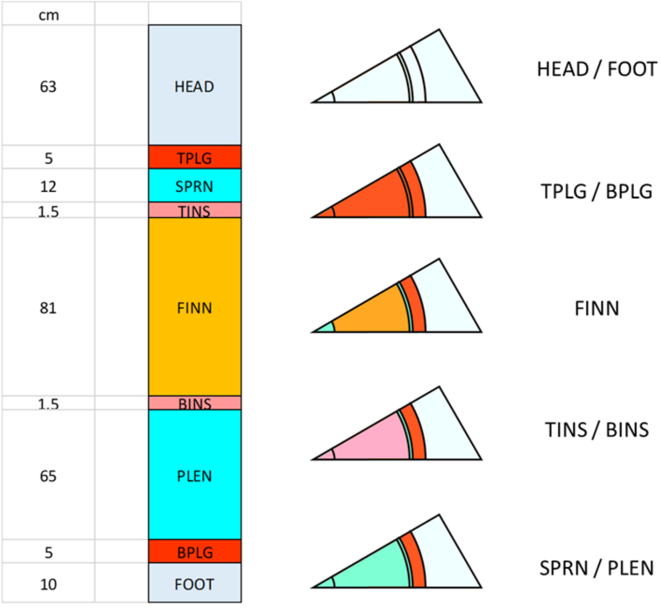


Figure 5.12: *3D cell model based on LFR configuration: axial and radial structure*

---

[20] The different materials are reported with different colors. In particular HEAD/FOOT are the homogeneous lead region; TPLG/BPLG correspond to bundle grids regions made with an AIM1 pin surrounded by lead; FINN to the fuel pins region; TINS/BINS are the insulator pin regions made by YSZ; SPRN/PLEN are void regions.

6 Complementary studies

Different complementary analysis have been performed with respect to the LFR case. The complementary studies includes:

- Several parametric studies concerning material, density and temperature variation oriented to a first investigation of the safety reactivity coefficients;

- Depletion investigations to complement previous analysis;

- Sodium-cooled Fast Reactor (SFR) analysis to benefit from CEA large experience on this subject;

- Simplified Open-literature pool-type Lead-cooled Fast Reactor (OLFR) analysis to open the work to fuel cycle studies.

6.1 Reactivity coefficients investigation

For reason of time, the present work has been restricted to the analysis of a single physics (neutronics), as a first step towards coupled calculations and multi-physics analysis. In order to get an idea of the interaction with other physics, typically feedback coefficients [53], [54],[78] have been evaluated. The approximations adopted during the calculation are described in the following parts.

In particular, the investigated parameter is the reactivity variation due to boundary and operation conditions. Unless otherwise indicated, the calculations have been performed on the fuel assembly geometry, using JEFF3.1.1 data library, 1968 energy group structure and SgECCO method for self-shielding.

6.1.1 Impact of materials compositions

In terms of materials variation, three cases have been considered:

- impact of the presence of impurities in the lead (coolant);

- impact of the material filling the internal guide tube;

- impact of the plutonium content in the fuel.

In the OECD/NEA benchmark, the lead composition includes the presence of several impurities (e.g. ^{209}Bi, ^{23}Na, ^{24}Mg). The calculations presented at Chapter 5 have been all carried out considering impurities in the coolant. The adoption of pure lead leads to an increase of the reactivity of about 13 pcm. Indeed, the presence of impurities increases the absorption within the coolant region and so causes a decrease in the number of neutrons available for the fission reaction. In particular, Figure 6.1 illustrates the microscopic absorption cross-section peaks within the energy range of approximately 5 eV and 10 keV. This behaviour is confirmed also by the coolant macroscopic absorption cross-section values. For the nominal case it is equal to $1.5255E^{-4}$ 1/cm, instead in the case without impurities $1.5162E^{-4}$ 1/cm.

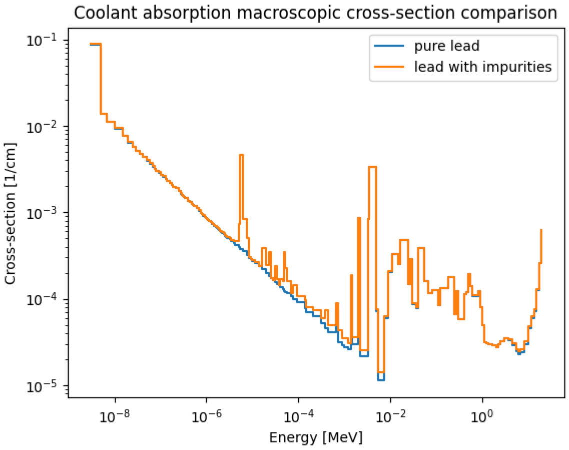


Figure 6.1: *Coolant absorption macroscopic cross-section comparison between pure lead and lead with impurities as coolant*

The influence of the material inside the central tube guide has been investigated as well in order to consider several configuration in core (tube instrumented or not, etc.). The reference case, as mentioned in the Section 4.2.1, involves the helium at 1 atm, and, for the purpose of this work, it was replaced by AIM1 steel (indicating some structure components) and with coolant (lead containing impurities).

The obtained k_∞ are shown in Figure 6.2. The higher reactivity is reached with the presence of helium due its low absorption cross-section and the lowest correspond to the use of AIM1 steel, as confirmed by the macroscopic absorption cross-section values evaluated on the whole fuel assembly. In the nominal case it is equal to $1.5776E^{-2}$ 1/cm, using the AIM1 is $1.5790E^{-2}$ and with lead $1.5781E^{-2}$ 1/cm. The maximum relative difference in reactivity corresponds to 174 pcm.

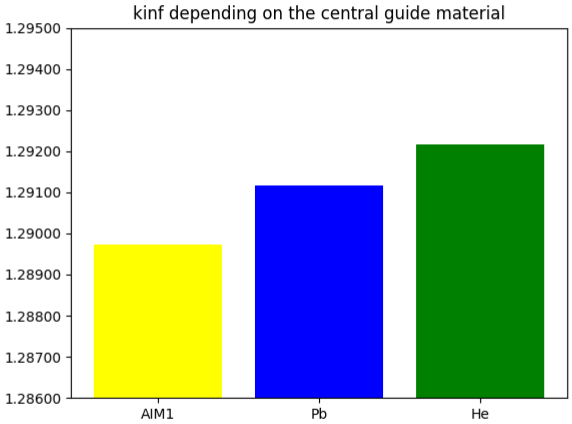


Figure 6.2: k_∞ *values depending on the central guide material*

To conclude, the fuel assembly located in the outer region of the core, having a plutonium content equal to 26.2 wt%, has been studied as well[21]. The outer fuel case has a higher reactivity (9670 pcm) if compared to the inner fuel cases. The spectrum of the two cases is compared in the Figure 6.3 and the fission macroscopic cross-section is shown as well in Figure 6.3 (on the right).

---

[21] The same fuel density as in the inner case has been adopted

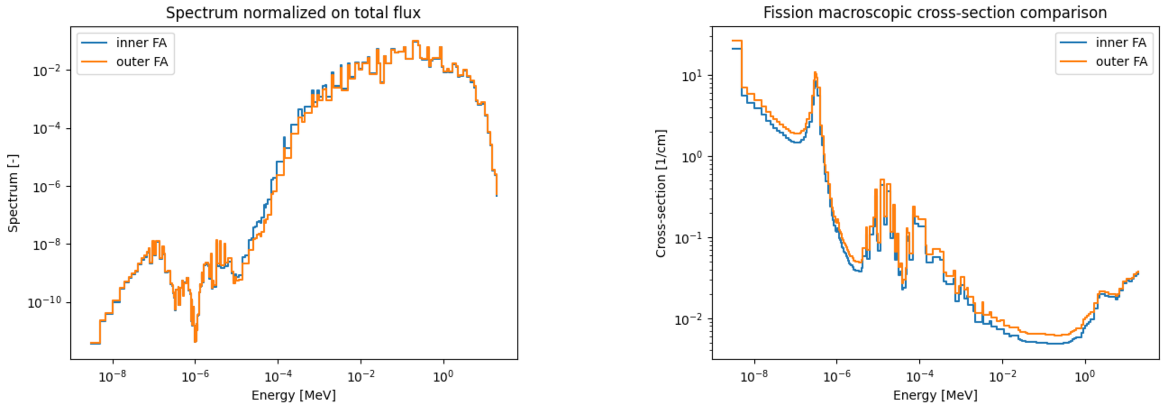


Figure 6.3: *Inner and outer fuel assembly spectrum and fission macroscopic cross-section comparisons*

6.1.2 Coolant density variation

Another aspect investigated is the reactivity feedback associated with coolant density variation. During transients, changes in power unbalance and distribution within the reactor, directly affect the temperature and, consequently, the coolant density that may affect the transient behaviour[78], [60], [51]. Normally, in thermal reactors, a decrease in coolant density results in reduced core reactivity due to a reduction of the moderation effects. In the contrary, fast reactors exhibit typically the opposite behaviour[79], [80].

As the number density of lead decreases, the capture rate decreases. These reductions lead to a positive reactivity feedback, as a consequence of a grater number of fast neutrons available for the fission reaction. However, a decrease in lead within the system also increases the neutron leakage, which has a negative impact to the reactivity that, under some specific condition tends to balance the overall core reactivity [81].

To better understand and verify this tendency, the lead density was progressively decreased up to 0% which corresponds to an hypothetical condition without coolant (voided condition). It has to be notice that this 2D case has been carried out under reflective boundary conditions therefore without contribution on the leakage term.

As reported in the Figure 6.4, the reactivity increases of 4800 pcm between nominal and voided conditions, following a linear trend.

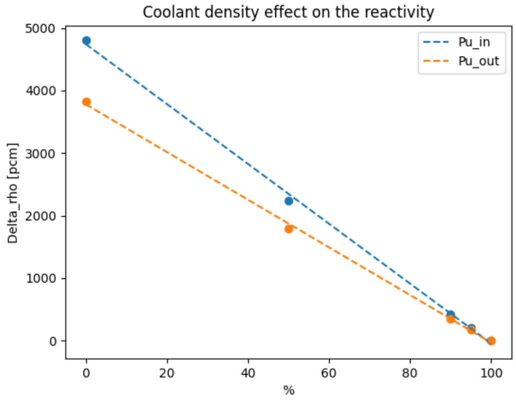


Figure 6.4: *Reactivity variation depending on the coolant density for the two Pu content*

It is important to underline that, the fuel assembly voided condition is a very hypothetical case given the relatively high boiling point of lead (boiling temperature of about 1750 °C); indeed a complete void scenario is highly unlikely in lead-cooled fast reactors compared to other concepts (e.g. SFR cases [78]). Depending on the core size, the coolant density reactivity effect may be mitigated by the leakages contribution, due to the low lead absorption cross-section even at higher densities [82].

The coolant density effect has been investigated also on the fuel assemblies placed in the outer part of the core corresponding to a grater amount of plutonium content (26.2 wt%). As it is possible to appreciate from the Figure 6.4, the main difference lays in the fact that in the lower plutonium content the reactivity increases faster with the reduction of coolant density [83].

6.1.3 Doppler effect

The fuel temperature variation has been investigated in order to estimate the Doppler reactivity effect. As indicated in the OECD/NEA benchmark, the fuel temperature has been increased of 600K while the other materials are kept at the operational temperature.

The fuel temperature increase leads to a change in relative velocity between particles and so in the effective interaction cross-sections. This phenomena is called nuclear Doppler effect and its effect on the microscopic cross-section is showed in Figure 6.5. Specifically, the capture cross-section tends to decrease in the peak region and to increase in the wings area, resulting in a reduction of energy self-shielding, but the total area is always conserved [84]. Consequently, the net capture rate increases, leading to a negative Doppler reactivity feedback. The magnitude of this phenomenon is measured by the Doppler constant, which is calculated as in Eq. 6.1:

$$K_D = \frac{\Delta\rho}{\ln\frac{T_2}{T_1}} \quad [pcm] \tag{6.1}$$

Where $\Delta\rho$ is evaluated with the Formula 5.1 and T_1 and T_2 are the two fuel temperatures before and after the perturbation [85].

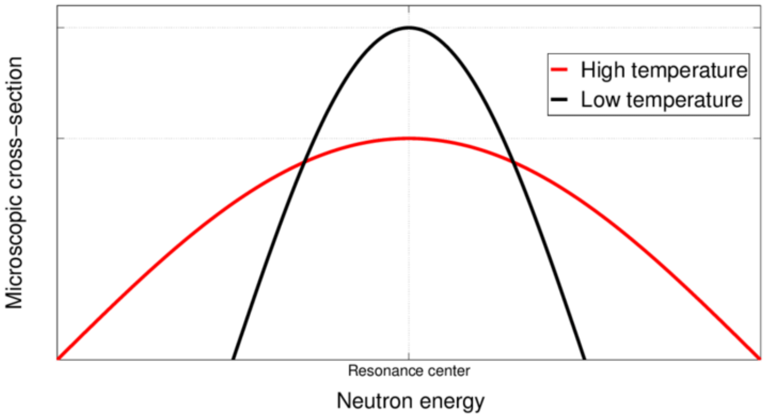


Figure 6.5: *The nuclear Doppler effect* [86]

The Doppler feedback for the 2D fuel assembly case is equal to -227 pcm with a corresponding K_D of -560 pcm. The value obtained is in agreement with the values obtained by the other partners participants [74].

As in the Paragraph 6.1.2, even in this case the outer fuel assembly case was investigated. In particular, the relative decrease in reactivity due to fuel temperature increase (600 K) is of 129 pcm having a Doppler constant equal to -317 pcm, slightly lower than the inner case, due to a lower amount of ^{238}U.

6.1.4 Geometry variation effects

As a direct consequence of temperature variations within the reactor core, the Sub-Assemblies can undergo a dimensional perturbations during both normal or accidental conditions. An increase in temperature may lead to the radial expansion of the structure and to the so-called *Flowering effect*. This last represents the non-uniform radial expansion of the core due to a non-uniform temperature distribution in the axial direction. The expansion typically causes an introduction of a negative reactivity in the core [87], [88]. Axial expansion is not treated in this study.

For the purposes of this work, a simplified model was investigated. The first analysis has been performed at assembly level simulating the expansion and the compaction of the fuel pins. For doing that, the cell pitch has been changed. The case is hypothetical, therefore the variation of the pitch have been chosen under arbitrary values (i.e. not connected to any expansion low[22]). The overall dimensions of the fuel assembly remain constant. To treat this case, two new geometries using the ALAMOS tool have been produced, as shown in Figure 6.6.

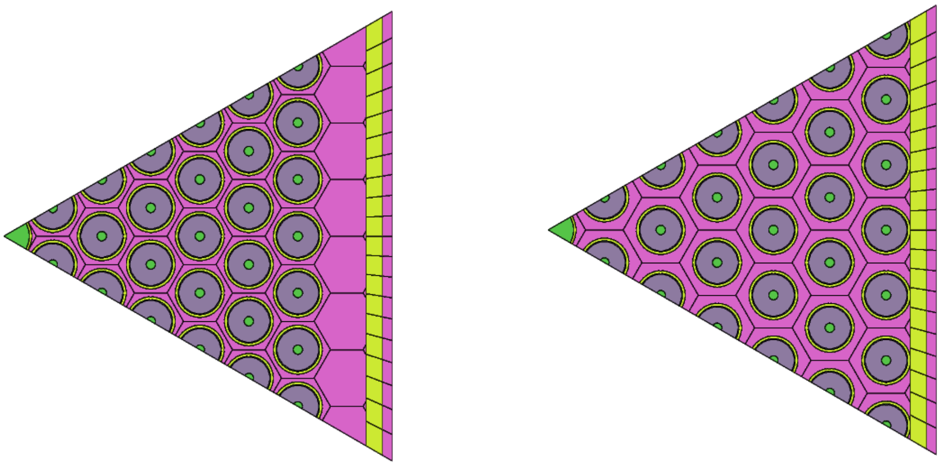


Figure 6.6: *Compacted and expanded simplified assembly geometries*

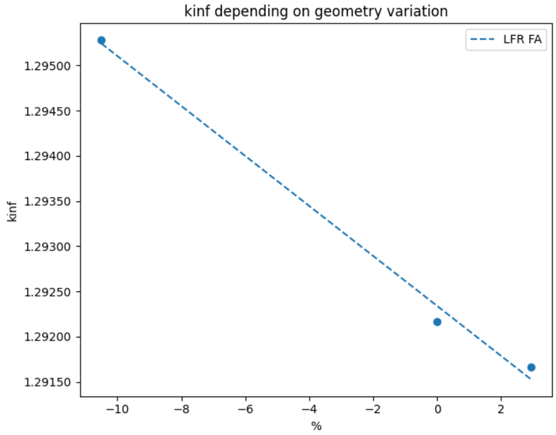


Figure 6.7: k_∞ *depending on the percentage variation of the cell pitch*

---

[22] On the left, the cell pitch has been reduced by 10.5%, while on the right, it was increased by 2.93%.

Typically, fast reactors are not arranged in their most critical configuration. As indicated in Figure 6.7 a fuel compaction corresponds to a global reactivity increases of about 186 pcm. Specifically, the closer proximity of the fuel pins results in a higher probability of neutron capture by the fuel, thus increasing the possibility of fission reactions, leading to an hardening of the neutron spectrum [89].

The expanded configuration is shown in Figure 6.6 on the right leads to reduction of the reactivity of about 30 pcm, as expected [85].

A similar test in terms of expansion has been performed on the super cell structure. In this case the wrapper and the cell pitch have been kept constant, instead the lead gap in between the Sub-Assemblies was increased of 1 mm, as shown in Figure 6.8. The calculations took advantages of the one-twelfth symmetry and refer to a ^{10}B enrichment equal to 42%.

The obtained k_∞ is equal to 1.04343 having a corresponding difference in reactivity equal to 797 pcm with respect to the nominal case.

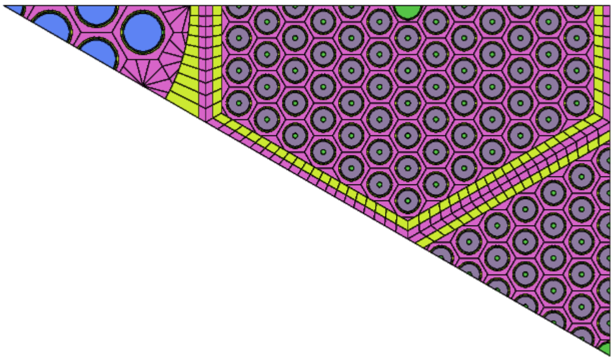


Figure 6.8: *Expanded control assembly super cell geometry*

On the basis of these calculations, it is possible to determine the expansion reactivity coefficients (typically indicated in pcm/K) for feeding for instance point kinetic models in system codes (e.g. CATHARE3). It is necessary to correlate the dimension variation due to material dilation to the associate temperature variation using expansions laws [90], [91], [**expansion**].

6.2 A SFR case

As already indicated APOLLO3® has been extensively applied to sodium-cooled FR. In order to profit of extensive experience, data, and knowledge at CEA, some of the parametric studies considered for the LFR case have been repeated for an SFR case. In particular, the case considered corresponds to a large size SFR core proposed as international benchmark to the OECD/NEA in the WPRS [21].

With this purpose, the 2D fuel cell and assembly configurations were derived by the 3600 MW_{th} SFR specifications. These configurations are depicted in Figure 6.9. The SFR cell shares a similar structure with the LFR cell, differing only by approximately 2 mm in cell pitch, although the dimensions, steel material, and fuel composition vary between the two. Concerning the fuel assembly, the major differences are a grater number of fuel pins (271 instead of 126 for LFR), the absence of a central guide, the different Pu content (˜17 wt% PuO_2) and different fuel/steel/coolant volume fractions[23]. The operational temperature adopted are the

---

[23] The volume fractions of the fuel assembly are the following [92]:

ones suggested in the benchmark, where the fuel temperature is set equal to 1500 K and the other materials to 743 K.

Additionally, in order to have more comparable results, the SFR fuel assembly calculation have been repeated adopting the same fuel composition (20.5 wt%) as the ALFRED case.

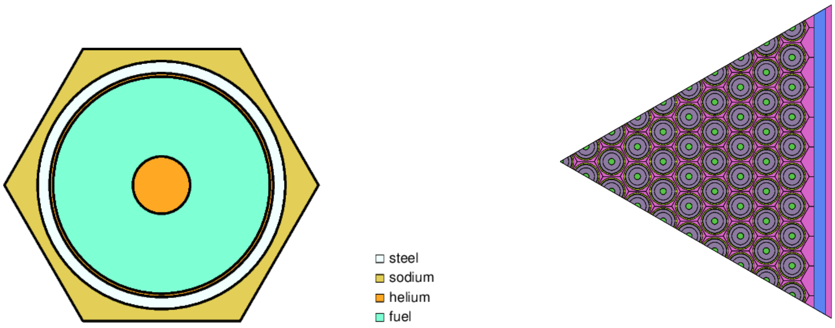


Figure 6.9: *3600 MW_{th} sodium-cooled fast reactor cell and fuel assembly*

The single cell simulation, shows a k_∞ equal to 1.17126 mainly due to the different Pu content and vector[16], [21].

In Figure 6.10, the LFR and SFR spectra for the cell case are compared, on the left, instead, on the right, the two spectra corresponding to the LFR MOX composition and the SFR one are reported.

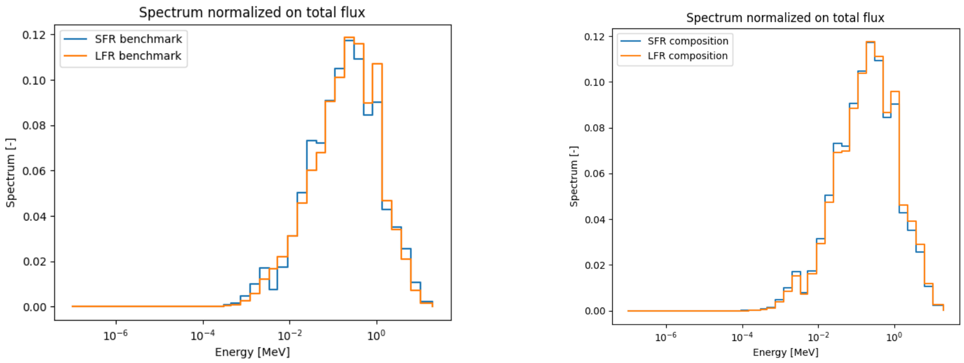


Figure 6.10: *Benchmark cases spectra comparison at 33 and MOX composition impact on the SFR spectrum*

In Figure 6.10 (left-side) the typical spectrum for SFR case is depicted with sodium resonance between 2 keV and 5 keV. The plutonium content has a slightly effect on the spectrum as indicated in 6.10 (right-side).

Figure 6.11 (on the left) shows the absorption macroscopic cross-section for the coolant in the two cases. As shown, the cross-section is higher at lower energies and then decreases in the fast

---

- fuel: 45.64% (SFR) - 31.54% (LFR)
- coolant: 27.73% (SFR) - 46.82% (LFR)
- clad: 11.84% (SFR) - 9.73% (LFR)
- wrapper: 8.16% (SFR) - 8.01% (LFR)

energy region. Furthermore, because of the higher density of fissile material in the ALFRED MOX composition, the corresponding macroscopic cross-section is greater than that of the SFR MOX. However, the difference is minimal, indicating that the primary influence is from coolant absorption rather than the fuel composition.

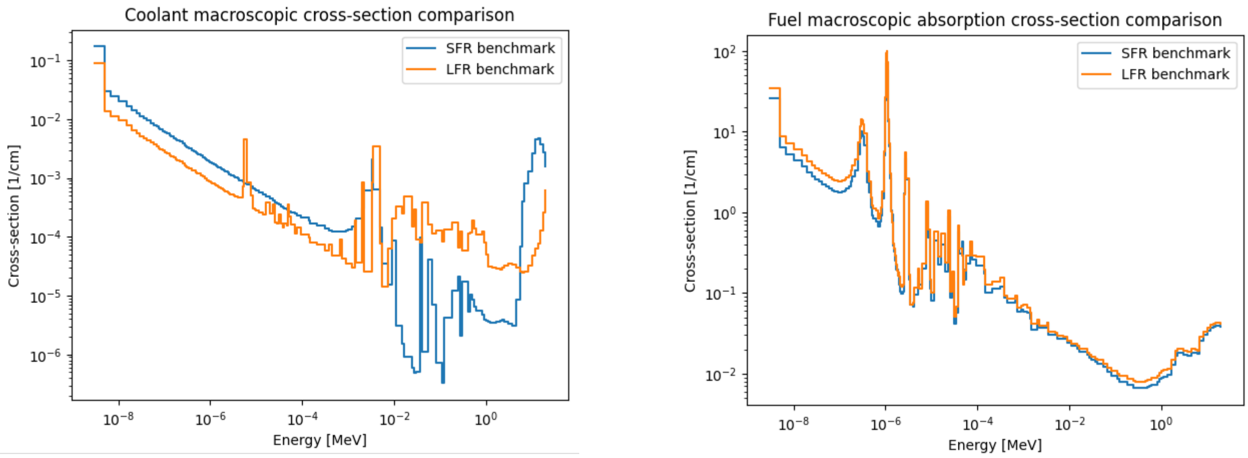


Figure 6.11: *Absorption macroscopic cross-sections comparison between the different regions*

The SFR case has been selected as well for allowing comparison with available data at CEA, and the results discussed above were confirmed by the ones reported in [92], allowing to confirm the choice done for the calculation scheme.

A similar parametric investigation as the one conducted in the Paragraph 6, have been done for SFR case. The cases corresponds to the MOX composition reported in the SFR benchmark (˜17 wt% PuO_2). The impact of temperatures, coolant densities and geometry variations has been investigated. The tendencies identified for the LFR case have been confirmed for the SFR case too.

As shown in the Figure 6.12, when the sodium density is progressively reduced to 0%, the reactivity decreases following a almost linear trend with increasing coolant density. However, in the sodium-cooled reactor, the slope of the curve is less steep, indicating a smaller increase in reactivity as the coolant density decreases from 100% to 0%. This effects may be due to the different considered geometry and structural material compositions (and eventually due to a more compact configuration for the SFR case).

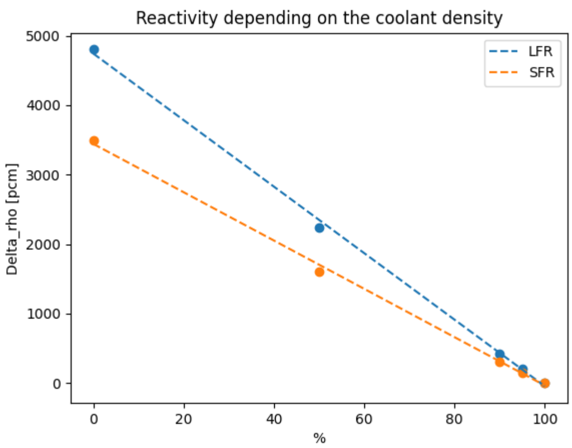


Figure 6.12: *Sodium density reduction effect on the reactivity compared with the LFR case*

For the SFR case, the Doppler constant has been calculated as well obtaining a value of -875 pcm (versus -560 pcm for the LFR case). The difference is mainly due by the plutonium content and consecutively by the amount of U238. In addition, the different initial temperature (1500 K for SFR and 1200 K for LFR) may contributes. More investigation have not been carried out. The fuel microscopic absorption cross-section are shown in Figure 6.13.

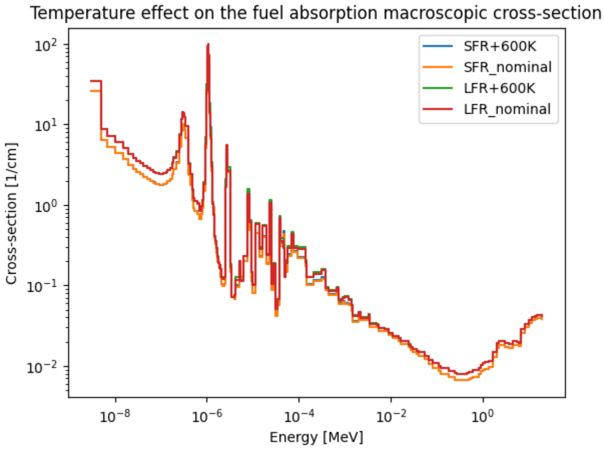


Figure 6.13: *Impact of the temperature on the fuel macroscopic absorption cross-section*

Similar considerations can be done for the expanded and compacted geometry configurations as indicated in Figure 6.14.

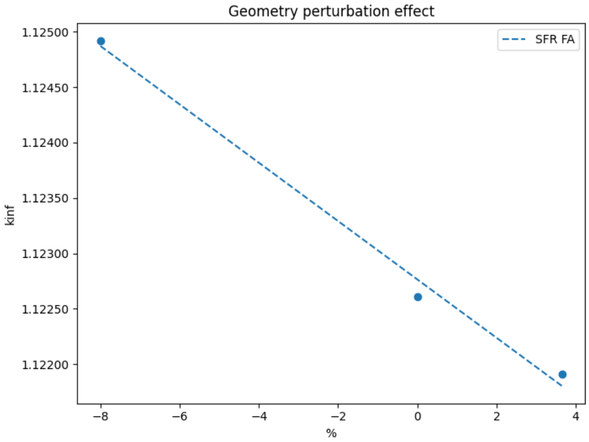


Figure 6.14: *Reactivity perturbation depending on the geometry expanded/compaction*

6.3 The Open LFR case

A different fuel assembly type based on Open-literature pool-type Lead-Cooled Fast reactor theoretical case has been investigated. The OLFR case is proposed for carrying out several investigation concerning multi-physics and fuel cycle analyses [66], [60].

The OLFR assembly is shown in Figure 6.15. It is composed by 19 central dummy pins normally filled with helium. The plutonium content is intermediate to the inner and outer content of the LFR case [66], [60] but in the study the same composition of the inner LFR case has been adopted. The central pins may allow the accommodation of instrumentation and/or fertile (e.g. as blanket) pins in order to improve the material balance of the core.

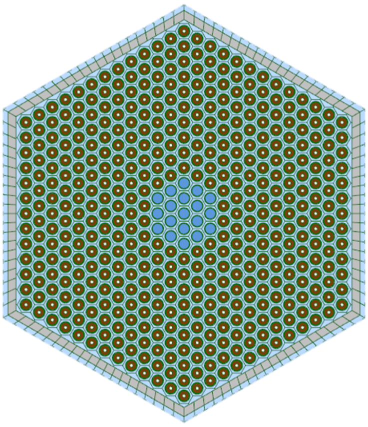


Figure 6.15: *OLFR fuel assembly geometry*

All the calculations have been performed using TONE self-shielding model and the same lead, fuel and steel compositions as the LFR case. The obtained k_∞, for one of the configurations considered for the OLFR [66], [60], using the JEFF3.1.1 libraries is equal to 1.26712, instead using the JEFF3.3 gives a value of 1.26641 with a relative difference in reactivity of about 44 pcm. This results are in line with the ones obtained during the ALFRED calculations mentioned in Paragraph 5.2.

Similar evaluations as performed before in terms of parametric studies have been done even in this case. In particular, the Doppler constant was investigated showing a Doppler constant equal to -587 pcm , with a difference in reactivity of -238 pcm.

Also the void conditions were slightly analyzed, providing a difference in reactivity of 5326 corresponding to a k_∞ of 1.35883.

6.4 Depletion calculations

Fast reactor concepts allow to exploit uranium oxide but also the plutonium oxide made using the spent fuel from light water reactor [93]. Indeed, the isotopic composition within the reactor core, and thus within the various assemblies, evolves continuously throughout the reactor's operational life. Fuel irradiation causes material changes in isotopic composition due to fuel transmutation under irradiation and decay. The isotopic composition evolution is governed by the Bateman equations [56].

For the purpose of this work, a preliminary investigation of the fuel depletion has been conducted using APOLLO3® and its depletion module MENDEL. The calculations have been performed on 2D fuel assembly case. The calculation scheme is reported in Annex C.1.

The LFR cell and the two fuel assembly cases (LFR and SFR) have been considered. For the three cases, the evolution has been carried out considering 340 effective full power days (efpd) and a mass power density equal to 53.336 W/g (according to the LFR benchmark [16]). The plutonium content in the three cases correspond to 20.5 wt%.

Figure 6.16 shows the reactivity evolution during irradiation. The three cases exhibit a similar trend, with differences in reactivity over the considered period being 1610 pcm for the cell case, 1700 pcm for the ALFRED fuel assembly, and 1620 pcm for the SFR case.

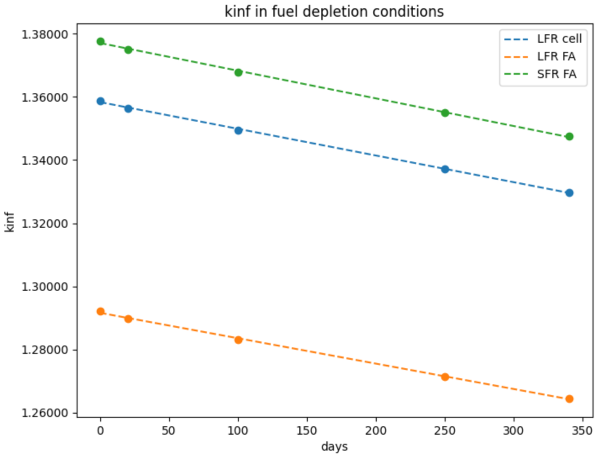


Figure 6.16: k_∞ *in fuel depletion conditions comparison between the different studied configurations*

Looking to the isotopes concentration evolution, Figure 6.17 shows some selected isotope (^{239}Pu, ^{241}Pu, ^{235}U, ^{238}U, ^{244}Cm, ^{241}Am): both fissile and fertile isotopes decrease linearly in time, with a similar behaviour in the three cases. The difference in the ^{239}Pu evolution may be due by the slightly different volume fractions of the three cases.

Curium 244 increases as the result of the successive neutron capture by plutonium and americium isotopes (Figure 6.17. As a minor actinide, ^{244}Cm gives a significant contributor to short-term radiotoxicity[24], but also to the decay heat and neutron emissions, necessitating improvements in the fuel cycle management process. Instead, in the long-term, the impact of ^{244}Cm is negligible, due to its relatively short half-time [95]. ^{241}Am is also analyzed. A linear increase is indicated in Figure 6.17.

Referring to the Figure 6.17, it is possible to notice that also the uranium content decreases in time as a consequence of the progressive capture into the fuel. A simplified evolution chain is indicated in Figure 6.18.

---

[24] The term radiotoxicity describes the adverse effects of incorporated substances (taken up into the body e.g. with food or breathing air) on health due to their radioactive property, as opposed e.g. to the chemical effects ("chemical toxicity") [94]

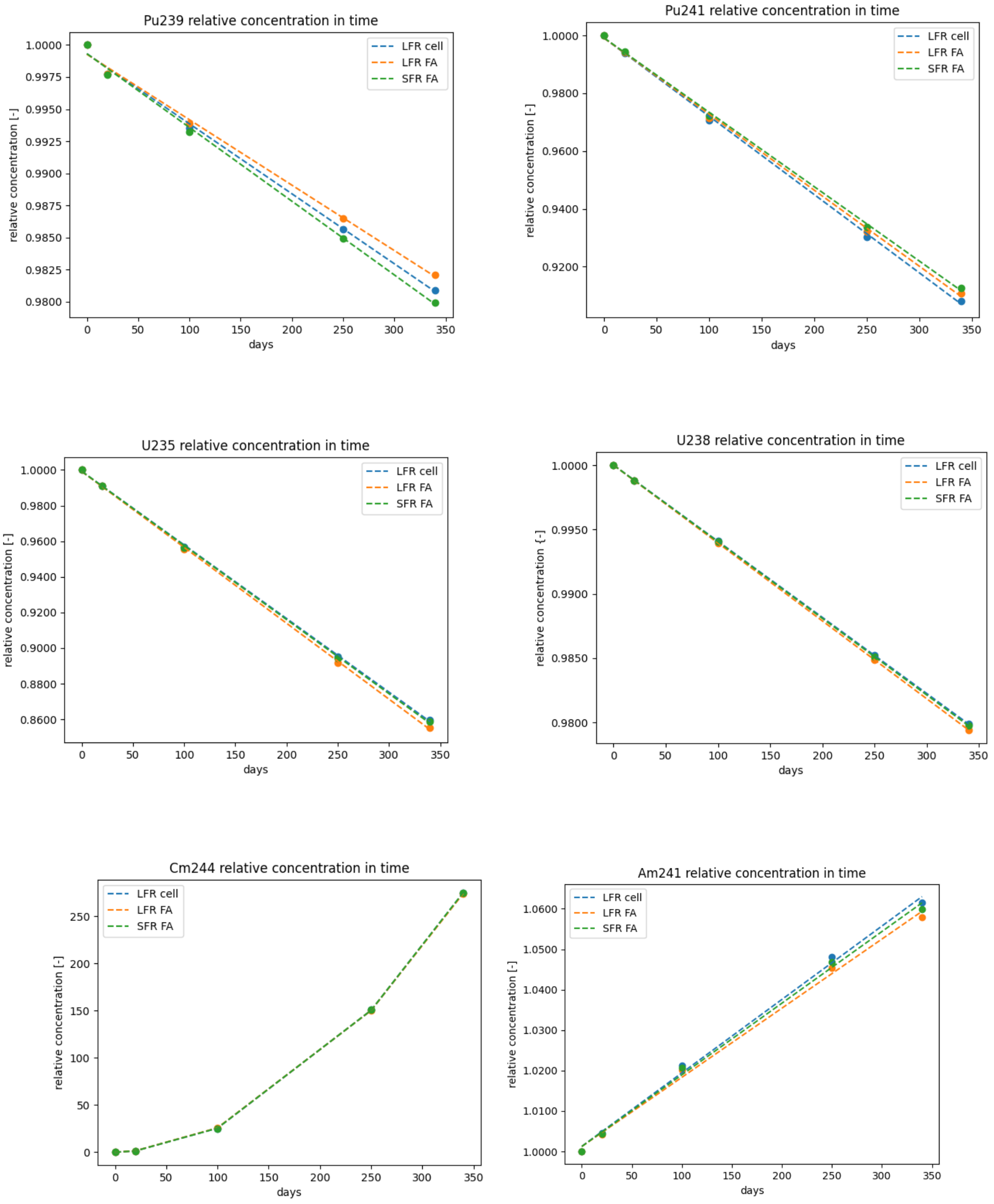


Figure 6.17: *Most relevant isotopes concentration in time*

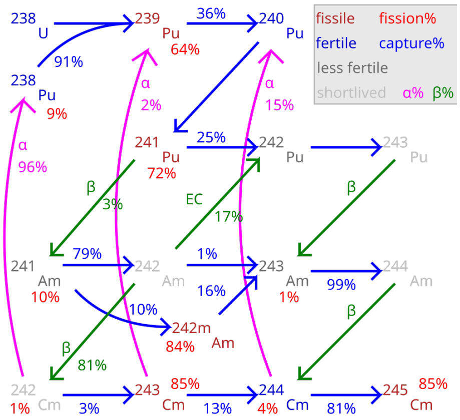


Figure 6.18: *Reaction chain starting from U_{238} [96]*

Additionally, to better understand the role of the depleted uranium within the reactor core, opening the discussion to fuel cycle needs, the OLFR case has been considered as well. In particular, a configuration where the central dummy pins were filled with depleted uranium has been investigated. The parameters considered for the evolution are the same as the previous case.

As shown in Figure 6.19, the presence of depleted uranium pins affect the reactivity variation. After 340 efpd, the reactivity is reduced of 1496 pcm with respect to the 1700 pcm for the LFR case due to the presence of the depleted uranium, which reacts with the neutrons forming fissionable plutonium. Some effect may also come from the different volume fraction of the two cases but this point has not been investigated in detail.

In the Figure 6.20 are reported the evolution in relative density on the concentration at the beginning of the cycle of the isotopes of interest. In particular, they are compared with the ones within the LFR reactor, reported above. The ^{239}Pu concentration reduces during operational lifetime but much less than the LFR case.

More investigations are foreseen, for instance changing the number of fertile pins or the plutonium vector and content, before evaluating the impact at core level. These analyses will be conducted as future work.

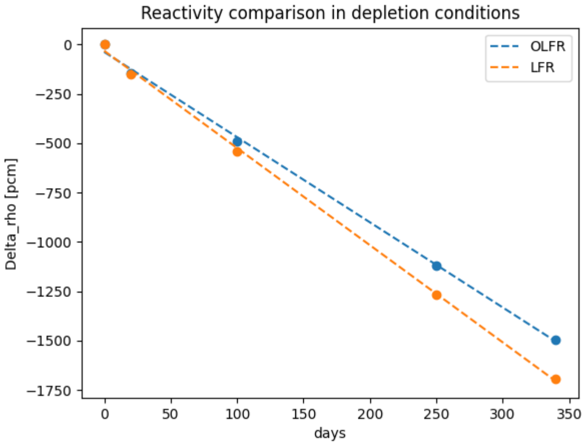


Figure 6.19: *Comparison of the reactivity under depletion conditions between the OLFR and ALFRED reactor*

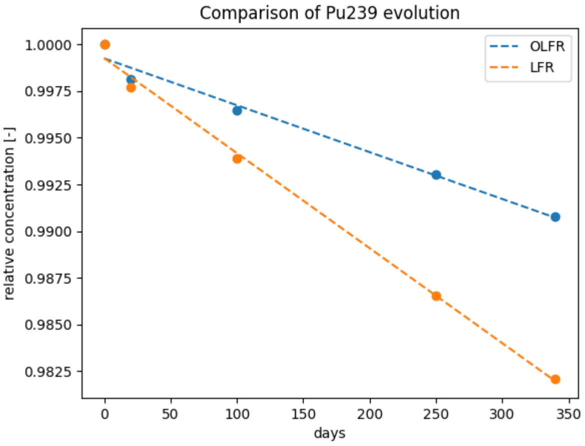


Figure 6.20: *Pu239 concentration in time: comparison between OLFR and LFR reactor*

7 Conclusions and perspectives

The activity performed during the Master's thesis work has been oriented to investigate all necessary steps for developing and evaluating a neutronic calculation scheme based on the CEA deterministic new generation code APOLLO3® .

The use of the unitary codes (APOLLO3®, ALAMOS and TRIPOLI4®) has allowed to get experience with the whole calculation chain from the generation of geometries up to Lattice results analysis. The work performed allowed to set up inputs and options for generating all necessary data for 3D core calculations. For reason of time, the core 3D calculations have not been included in the present work. Several feedback have been provided to the code developers for reinforcing the robustness of the codes.

The study has been inserted within the activities of the OECD/NEA Working Party on Scientific Issues and Uncertainty Analysis of Reactor Systems (WPRS) and in particular of the EGPRS [67]. In this NEA working group, several benchmarks are provided. Among those, a LFR and SFR benchmark exercises are performed, of which specifications have been investigated with the help of specific pre-processing Pyhton scripts. The study has focused mainly on the LFR case based on the ALFRED reactor.

It has been possible, during the internship, to provide a full set of results for the first two phases of the benchmark and to analysis the impact of different modeling options (self-shielding models, libraries, solvers options, etc.) on those results.

In addition, several parametric studies have been conducted for opening the discussion to multiphysics and fuel cycle analysis. To do that fuel temperature, coolant density and geometries have been changed in order to prepare the evaluation of the typical feedback coefficients used in transient analysis. In addition, an industrial size SFR and a Open-literature pool-type Lead-cooled Fast Reactor (OLFR) have been also investigated. All of the obtained results have been properly analyzed with the support of specific post-processing Python scripts.

All the calculations are made with the support of the main CEA tools, with a particular focus on APOLLO3® for neutronic calculations and ALAMOS for the geometries construction. The stochastic codes (TRIPOLI-4® and OpenMC) played a crucial role in terms for supporting the verification of the results.

Several activities may be considered as follow up of this work in particular involving 3D evaluations. Future studies could include a full 3D core evaluation based on the homogenized quantities provided by the 2D lattice calculations of the individual sub-assemblies. This is a preliminary step for performing coupled calculations (neutronic/thermal-hydraulic/thermos-mechanics) and multi-physics analysis. Concerning the fuel cycle, the OLFR case may be further investigate for optimizing the material balance of the system. An additional activity to be consider is the application of the INCA platform for this kind of study.

Among the activities, a more detailed comparisons against results with other codes and options provided by the benchmark will be considered.

References

[1] Yoshihiko Sakamoto et al., *Selection of sodium coolant for fast reactors in the US, France and Japan*, Nuclear Engineering and Design, 254, pp. 194–217, 2013

[2] M. Salvatores et al., *Nuclear fuel cycle synergies and regional scenarios for Europe*, tech. rep., OECD Publishing, 2009

[3] C. Fazio et al., *Massimo Salvatores' seminal ideas and comprehensive views on nuclear fuel cycle studies*, Annals of Nuclear Energy, 157, pp. 108–223, 2021

[4] *TANDEM Project*, URL: `https://tandemproject.eu/` (visited on 05/31/2025)

[5] NEA, *Advanced Nuclear Fuel Cycles and Radioactive Waste Management*, OECD Publishing, Paris, 2006

[6] IAEA, *Operational and Decommissioning Experience with Fast Reactors*, International Atomic Energy Agency, Vienna, 2004

[7] F. Gauché, *Generation IV reactors and the Astrid prototype: Lessons from the Fukushima accident*, Comptes Rendus Physique, 13, pp. 365–371, 2012

[8] J. P. Grouiller et al., *Plutonium Recycling Capabilities of Astrid Reactor*, Proceedings International Conference on Fast Reactors and Related Fuel Cycles (FR17), IAEA, Yekaterinburg, Russia, 2017

[9] H. Noel, *Projet EFR (European Fast Reactor)*, Techniques de l'Ingénieur, 1995

[10] G.L. Fiorini et al., *European Commission – 7th Framework Programme The Collaborative Project on European Sodium Fast Reactor (CP ESFR)*, Nuclear Engineering and Design, 241, pp. 3461–3469, 2011

[11] *ESFR-SMART Project*, URL: `https://esfr-smart.eu/about-the-project/` (visited on 08/23/2024)

[12] *GEN IV International Forum webinars*, URL: `https://www.gen-4.org/gif/jcms/c_84279/webinars` (visited on 08/23/2024)

[13] CEA, *Sodium-Cooled Nuclear Reactors*, A Nuclear Energy Division Monograph, 2016

[14] D. G. Ceballos, *Modelling of the DHR Heat Exchanger Using the Computer Code AC2-ATHLET*, ETSEIB, 2024

[15] V. V. Orlov et al., *The closed on-site fuel cycle of the BREST reactors*, Progress in Nuclear Energy, 47, pp. 171–177, 2005

[16] G. Grasso, *EGPRS Lead-cooled Fast Reactor (LFR) Benchmark*, OECD/NEA, 2023

[17] P. Mosca et al., *Overview of the New Code Capabilities for Reactor Physics Analysis*, Nuclear Science and Engineering, 0, pp. 1–14, 2024

[18] E. Garcia et al., *Flux distribution in the Superphénix start-up core with APOLLO3®*, Annals of Nuclear Energy, vol. 133, 2019, pp. 889–899

[19] L. Graziano, *Méthode accélérée aux caractéristiques pour la solution de l'équation du transport des neutrons, avec une approximation polynomiale axiale*, Master's thesis, Palaiseau, France: Université Paris-Saclay, 2018

[20] V. Pascal, *PHENIX: Interpretation of ECRIX-H transmutation experiment with TRIPOLI-4®*, Proceedings of the International Conference on Mathematics Computation, M&C 2017, Jeju, Korea, Apr. 16, 2017

[21] L. Buiron, G. Rimpault, P. Sciora, et al., *Benchmark for uncertainty analysis in modelling (UAM) for design, operation and safety analysis of SFRs*, AEN-WPRS, 2020

[22] M. Burrone, *Study of Eigenvalue Formulations in the P_N Approximation of the Neutron Transport Equation*, Master's thesis, Turin, Italy: Politecnico di Torino, 2018, pp. 7–14

[23] A. Barbarino, *Numerical Methods for Neutron Transport Calculations of Nuclear Reactors*, Master's thesis, Politecnico di Torino, 2014

[24] S. Dulla and P. Ravetto, *Nuclear fission reactor physics and transport theory*, Lecture notes, A. Y. 2022/2023

[25] G. I. Bell and S. Glasstone, *Nuclear Reactor Theory*, Van Nostrand Reinhold Co. 1970

[26] E. E. Lewis and W. F. Miller, *Computational Methods of Neutron Transport*, American Nuclear Society, 1993

[27] Y. Wu, *Fusion Neutronics*, Springer, 2017

[28] S. Santandrea, L. Graziano, and D. Sciannandrone, *Accelerated polynomial axial expansions for full 3D neutron transport MOC in the APOLLO3® code system as applied to the Astrid fast breeder reactor*, Annals of Nuclear Energy, 113, pp. 194–236, 2018

[29] D. Sciannandrone, S. Santandrea, and R. Sanchez, *Optimized tracking strategies for step MOC calculations in extruded 3D axial geometries*, Annals of Nuclear Energy, 87, pp. 49–60, 2016

[30] William Boyd, Scott Shaner, et al., *The OpenMOC method of characteristics neutral particle transport code*, Annals of Nuclear Energy, 68, pp. 43–52, 2014

[31] T. Mazumdar and S. B. Degweker, *Solution of neutron transport equation by Method of Characteristics*, Annals of Nuclear Energy, 77, pp. 522–535, 2015

[32] M. Ajami et al., *A Pure Dynamic Monte Carlo Code for the Neutronic Analysis of Nuclear Reactors*, Annals of Nuclear Energy, 165, p. 108627, 2022

[33] J. Leppänen, *Development of a New Monte Carlo Reactor Physics Code*, PhD thesis, Aalto University, in collaboration with VTT Technical Research Centre of Finland, 2007

[34] A. Greganti, *Validation of an Isotope Evolution Model for APOLLO3® Calculations in SFR Core*, Politecnico di Milano, 2016

[35] N. Gibson, *Novel Resonance Self-Shielding Methods for Nuclear Reactor Analysis*, Rensselaer Polytechnic Institute, 2016

[36] A. Dos Santos, M. V. C. Galia, and L. Leal, *Multigroup Covariance Matrix Self-Shielding Effects for Thermal Reactors Fueled with Slightly Enriched Uranium*, SSRN Electronic Journal, 2022

[37] Y. Chai et al., *Solving point burnup equations by Magnus method*, Nuclear Engineering and Technology, 51, pp. 949–953, 2019

[38] P. K. Romano et al., *OpenMC: A state-of-the-art Monte Carlo code for research and development*, Annals of Nuclear Energy, 82, pp. 90–97, 2015

[39] M. Pusa, *Numerical methods for nuclear fuel burnup calculations*, VTT Science, 2013

[40] G. Rimpault et al., *The APOLLO3® scientific tool for SFR neutronic characterization: current achievements and perspectives*, Proceedings International Conference on Fast Reactors and Related Fuel Cycles (FR17), IAEA, Yekaterinburg, Russia, 2017

[41] A. Brighenti et al., *Development of a Multi-Parameter Library Generator Prototype for VVER and PWR Applications Based on APOLLO3®*, Proceedings of the International Conference on Mathematics Computational Methods Applied to Nuclear Science and Engineering (M&C 2023), Niagara Falls, Ontario, Canada, Oct. 2023

[42] D. Tomatis et al., *Overview of SERMA's Graphical User Interfaces for Lattice Transport Calculations*, Energies, 15, p. 1417, 2022

[43] *CAMIVVER H2020 – Civil Nuclear Safety*, URL: `https://www.camivver-h2020.eu/` (visited on 08/23/2024)

[44] F. Inzirillo, *Development of a Python library for the generation of 2D unstructured geometries of radial reflectors in support of industrial model verification*, Master's thesis, Polytechnique Montréal, 2023

[45] J. F. Vidal et al., *New reference APOLLO3® calculation scheme for light water reactors - analysis of the BEAVRS bench*, Proceedings of the International Conference PHYSOR2020, 2020

[46] H. Guo et al., *Advanced method for neutronic simulation of control rods in sodium fast reactors: Numerical and experimental validation*, Annals of Nuclear Energy, 129, pp. 90–100, 2019

[47] B. Vezzoni et al., *Best estimate schemes in lattice calculations for industrial cases, with the help of a new leakage synthetic algorithm*, Proceedings International Conference MC 2021, 2021

[48] E. Brun et al., *TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code*, Annals of Nuclear Energy, 82, pp. 151–160, 2014

[49] CEA, *Une monographie de la direction de l'énergie nucléaire – La neutronique*, CEA Saclay, 2015

[50] C. Fiorina et al., *On the development of multi-physics tools for nuclear reactor analysis based on OpenFOAM: state of art, lessons learned and perspectives*, Nuclear Engineering and Design, 387, p. 111604, 2021

[51] B. Calgaro and B. Vezzoni, *Advanced Couplings and Multiphysics Sensitivity Analysis Supporting the Operation and the Design of Existing and Innovative Reactors*, Energies, 15, p. 3341, 2022

[52] C. Vaglio-Gaudard et al., *Challenge for the validation of high-fidelity multi-physics LWR modeling and simulation: Development of new experiments in research reactors*, Proceedings of the International Conference on Frontier Energy Research 2023, January 26, 2023, 2023

[53] G. Grasso et al., *Stress-testing the ALFRED design – Part I: Impact of nuclear data uncertainties on Design Extension Conditions transients*, Progress in Nuclear Energy, 106, pp. 372–386, 2018

[54] G. Grasso et al., *Stress-testing the ALFRED design – Part II: Quantification of uncertainties on the fuel assembly temperature field*, Progress in Nuclear Energy, 105, pp. 301–308, 2018

[55] S. Lahaye et al., *First verification and validation steps of MENDEL Release 1.0 cycle code system*, Proceedings of the International Conference PHYSOR2014, Kyoto, Japan, September 2014, 2014

[56] J. Cetnar, *General solution of Bateman equations for nuclear transmutations*, Annals of Nuclear Energy, 33, pp. 640–645, 2006

[57] S. Santandrea et al., *Codes And Methods Improvements for VVER Comprehensive Safety Assessment*, Proceedings of the International Conference CAMIVVER, 2023

[58] D. Schneider et al., *CEA/DEN deterministic multi-purpose code for reactor physics analysis*, Proceedings of the International Conference PHYSOR2016, Sun Valley, United States, May 2016, 2016

[59] L. Mao, I. Zmijarevic, and R. Sanchez, *Resonance Self-Shielding Methods for Fast Reactor Calculations—Comparison of a New Tone's Method with the Subgroup Method in APOLLO3®*, Nuclear Science and Engineering, 188, pp. 1–18, 2017

[60] B. Calgaro and B. Vezzoni, *Advanced Couplings and Multiphysics Sensitivity Analysis Supporting the Operation and Design of Existing and Innovative Reactors*, Proceedings International Conference BEPU 2024, Lucca, Italy, 2024

[61] Paul K. Romano et al., *OpenMC: A state-of-the-art Monte Carlo code for research and development*, Annals of Nuclear Energy, 82, pp. 90–97, 2015

[62] Paul K. Romano and B. Forget, *The OpenMC Monte Carlo particle transport code*, Annals of Nuclear Energy 51, 274–281, 51, pp. 274–281, 2013

[63] A. Hébert, *DRAGON5 and DONJON5, the contribution of École Polytechnique de Montréal to the SALOME platform*, École Polytechnique de Montréal, 87, pp. 12–20, 2014

[64] *MEDCoupling tutorial (Salome)*, URL: `https://docs.salome-platform.org/latest/dev/MEDCoupling/tutorial/index.html` (visited on 08/23/2024)

[65] M. Robin, *Développement d'une librairie Python pour la génération de géométries déstructurées cartésiennes pour APOLLO3®*, Master's thesis, Grenoble INP - Phelma, 2023

[66] B. Calgaro and B. Vezzoni, *Numerical Modeling driving Advanced Reactor Analyses: From Theory to Cutting Edge*, European Nuclear Young Generation Forum (ENYGF), Zagreb, 2025

[67] *Expert Group on Physics of Reactor Systems (EGPRS)*, URL: `https://www.oecd-nea.org/jcms/c_12860/expert-group-on-physics-of-reactor-systems-egprs` (visited on 08/23/2024)

[68] *Working Party on Scientific Issues and Uncertainty Analysis of Reactor Systems (WPRS)*, URL: `https://www.oecd-nea.org/jcms/c_12832/working-party-on-scientific-issues-and-uncertainty-analysis-of-reactor-systems-wprs.html` (visited on 08/23/2024)

[69] A. I. Orlov and B. A. Gabaraev, *Heavy liquid metal cooled fast reactors: peculiarities and development status of the major project*, NUCET, 9, pp. 1–18, 2023

[70] A. Alamberti et al., *The ALFRED Project*, Proceedings International Conference SIEN, Bucharest, Romania, 2013

[71] D. Schneider et al., *CEA/DEN deterministic multi-purpose code for reactor physics analysis*, Proceedings of the International Conference PHYSOR 2016, Sun Valley, United States, May 2016

[72] A. Koning et al., *The JEFF-3.1 Nuclear Data Library*, Technical Report JEFF Report 21, Paris, France: OECD Nuclear Energy Agency (NEA), Data Bank, 2006

[73] A. J. Plompen et al., *The Joint Evaluated Fission and Fusion Nuclear Data Library, JEFF-3.3*, The European Physical Journal A, 56, p. 181, 2020

[74] *B. Vezzoni, personal communication.*

[75] H. Guo and L. Buiron, *Innovative Sodium Fast Reactors Control Rod Design*, Proceedings of the International Conference Atoms for the Future 2018 and 4th GIF Symposium, Paris, France, 2018

[76] David Álvarez Romero, *ALFRED neutronics benchmark*, Universidad Politécnica Madrid, 2023

[77] A. Gammicchia, *Development and acceleration of a 3D characteristics method including an axial polynomial expansion of cross sections*, Master's thesis, Palaiseau, France: Université Paris-Saclay, 2021

[78] B. Vezzoni et al., *Safety-Related Optimization and Analyses of an Innovative Fast Reactor Concept*, Sustainability, 4, pp. 1274–1291, 2012

[79] T. Ishizu et al., *Development of a simple model for estimating the design limit of core void reactivity to prevent re-criticality of MOX-fueled cores in liquid metal-cooled fast reactors*, Nuclear Engineering and Design, 374, p. 111045, 2021

[80] P. Sciora et al., *Low Void Effect Core Design Applied on 2400 MWth SFR Reactor*, Proceedings of the International Congress on Advances in Nuclear Power Plants (ICAPP 2011), Nice, France, 2011

[81] Z. I. Böröczki, À. Aranyosy, and M. Szieberth, *Comparison of calculation methods for lead cooled fast reactor reactivity effects*, Annals of Nuclear Energy, 171, p. 109042, 2022

[82] W. Wang et al., *Void effect research on MOX-fueled lead-cooled fast reactor*, Annals of Nuclear Energy, 173, p. 109122, 2022

[83] V. V. Semishin, *Evaluation of neutronic performance for the VVER-1000 reactor core with regenerated uranium-plutonium fuel*, NUCET, 3, pp. 177–182, 2023

[84] A. E. Walter and A. B. Reynolds, *Fast Breeder Reactors*, Pergamon International Library, 1981

[85] P. Puthiyavinayagam, *Joint ICTP/IAEA School on Physics and Technology of Fast Reactors Systems*, Proceedings International School on Physics and Technology of Fast Reactor Systems, Miramare, Trieste, Italy, 2009

[86] S. Qvist, *Safety and core design of large liquid-metal cooled fast breeder reactors*, Berkeley, CA, USA: University of California, Berkeley, 2013

[87] B. Fontaine et al., *Description and preliminary results of PHENIX core flowering test*, Nuclear Engineering and Design, 241, pp. 4143–4151, 2011

[88] C. Patricot, E. Hourcade, and K. Ammar, *APOLLO3® based method for 3D warped cores calculations – application to flowering tests of Phenix*, Proceedings International Conference PHYSOR 2014, Kyoto, Japan, 2014

[89] R. Lo Frano et al., *Assessment of the structural-dynamic effects caused by the core compaction*, Technical Report, ENEA, 2014

[90] S. G. Popov et al., *Thermophysical properties of MOX and* UO_2 *fuels including the effects of irradiation*, tech. rep., ORNL, 1996

[91] L. Luzzi et al., *Modeling and Analysis of Nuclear Fuel Pin Behavior for Innovative Lead Cooled FBR*, tech. rep., ENEA, 2014

[92] R. Delaplanche, *Analyse des paramètres neutroniques d'un réacteur à neutrons rapides et caloporteur plomb*, ENSI Caen, 2023

[93] M. Salvatores et al., *Fuel cycle analysis of TRU or MA burner fast reactors with variable conversion ratio using a new algorithm at equilibrium*, Nuclear Engineering and Design, 239, pp. 2160–2168, 2009

[94] O. P. Ojo et al., *Investigative study of radiotoxicity of spent nuclear fuel assembly of some commercial nuclear power plants*, Applied Radiation and Isotopes, 190, p. 110503, 2022

[95] T. Kooyman et al., *A comparison of curium, neptunium and americium transmutation feasibility*, Annals of Nuclear Energy, 112, pp. 748–758, 2018

[96] *Breeder reactor (Wikipedia)*, URL: `https://en.wikipedia.org/wiki/Breederreactor` (visited on 08/23/2024)

[97] F. Heidet, *Maximum Fuel Utilization in Advanced Fast Reactors without Actinides Separation*, University of California, Berkeley, 2010

[98] H. Noel, *Projet EFR (European Fast Reactor)*, Techniques de l'Ingénieur, 1995

[99] G. L. Fiorini et al., *European Commission – 7th Framework Programme The Collaborative Project on European Sodium Fast Reactor (CP ESFR)*, Nuclear Engineering and Design, vol. 241, 2011, pp. 3461–3469

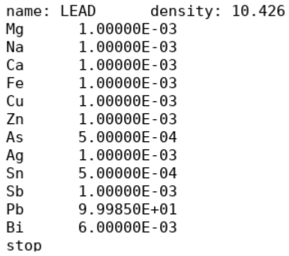
[100] S. Santandrea, R. Sanchez, and L. Mao, *Treatment of Boundary Conditions in Trajectory-Based Deterministic Transport Methods*, Nuclear Science and Engineering, 140, pp. 23–50, 2002

[101] G. Grasso et al., *The core design of ALFRED, a demonstrator for the European lead-cooled reactors*, Nuclear Engineering and Design, 278, pp. 287–301, 2014

[102] G. Rimpault et al., *The ERANOS Code and data system for fast reactor neutronic analyses*, Proceedings International Conference PHYSOR 2002, Seoul, South Korea, 2002

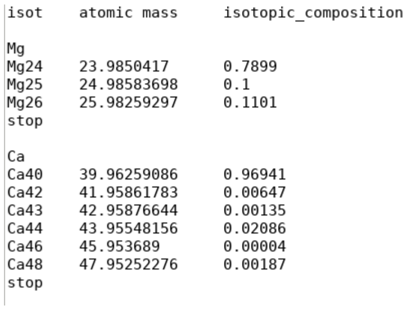
A Annex: Python Pre-processing and Post-processing scripts

A pre-processing script dedicated to composition definition has been created for converting the data provided by the benchmark (e.g. material composition) into the format accepted by the codes. In particular, it is necessary to indicate on specific .txt files the different material name with the corresponding composition (Figure A.1) and the elements isotopic composition (Figure A.2). Then, these files are used by the Python script in order to convert them in the specific format and to write them in an output file.



```
name: LEAD      density: 10.426
Mg      1.00000E-03
Na      1.00000E-03
Ca      1.00000E-03
Fe      1.00000E-03
Cu      1.00000E-03
Zn      1.00000E-03
As      5.00000E-04
Ag      1.00000E-03
Sn      5.00000E-04
Sb      1.00000E-03
Pb      9.99850E+01
Bi      6.00000E-03
stop
```

Figure A.1: *Pre-processing input file*



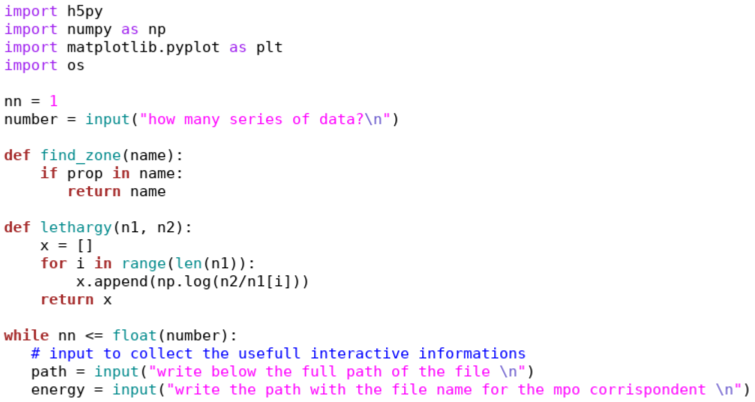
```
isot    atomic mass     isotopic_composition

Mg
Mg24    23.9850417      0.7899
Mg25    24.98583698     0.1
Mg26    25.98259297     0.1101
stop

Ca
Ca40    39.96259086     0.96941
Ca42    41.95861783     0.00647
Ca43    42.95876644     0.00135
Ca44    43.95548156     0.02086
Ca46    45.953689       0.00004
Ca48    47.95252276     0.00187
stop
```

Figure A.2: *Pre-processing input file*

In addition, during the initial stages of the work, the NEMESI-based UDF APOLLO3® interface was used [41], which required multiple input files with specific naming conventions. A Python script was created to move the files generated by ALAMOS (.za, .ap3, and .eqg) from their original directories to the designated input folder for APOLLO3® calculations renaming these files as needed.

With the purpose of collecting, plotting, and then interpreting the results some post-processing Python scripts have been developed.

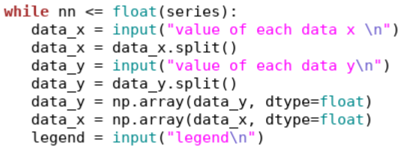
The first script is designed to generate a step graph that plots the desired quantity as a function of energy or lethargy. It operates by taking the full path of a specific .hdf file as input (both MPO and Hierarchical Data Format output file (HDF)), as shown in Figure A.3.



```python
import h5py
import numpy as np
import matplotlib.pyplot as plt
import os

nn = 1
number = input("how many series of data?\n")

def find_zone(name):
    if prop in name:
       return name

def lethargy(n1, n2):
    x = []
    for i in range(len(n1)):
        x.append(np.log(n2/n1[i]))
    return x

while nn <= float(number):
   # input to collect the usefull interactive informations
   path = input("write below the full path of the file \n")
   energy = input("write the path with the file name for the mpo corrispondent \n")
```

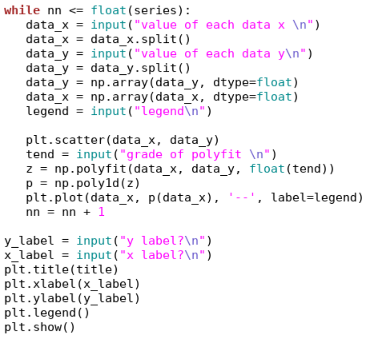
Figure A.3: *Post-processing script*

Thanks to the *input* command it is possible to choose all the different parameters of the graph, such as the names of the axis, the title, but also the scale of the two axis, the number of energy groups and the y-axis quantity. The y-axis quantity could refer to the whole geometry or to a single region of the geometry (e.g. the fuel zone). Figure A.4 shows the script content.



```python
while nn <= float(series):
   data_x = input("value of each data x \n")
   data_x = data_x.split()
   data_y = input("value of each data y\n")
   data_y = data_y.split()
   data_y = np.array(data_y, dtype=float)
   data_x = np.array(data_x, dtype=float)
   legend = input("legend\n")
```

Figure A.4: *Post-processing script for general data*



```python
while nn <= float(series):
   data_x = input("value of each data x \n")
   data_x = data_x.split()
   data_y = input("value of each data y\n")
   data_y = data_y.split()
   data_y = np.array(data_y, dtype=float)
   data_x = np.array(data_x, dtype=float)
   legend = input("legend\n")

   plt.scatter(data_x, data_y)
   tend = input("grade of polyfit \n")
   z = np.polyfit(data_x, data_y, float(tend))
   p = np.poly1d(z)
   plt.plot(data_x, p(data_x), '--', label=legend)
   nn = nn + 1

y_label = input("y label?\n")
x_label = input("x label?\n")
plt.title(title)
plt.xlabel(x_label)
plt.ylabel(y_label)
plt.legend()
plt.show()
```

Figure A.5: *Post-processing script*

The second script aims to plot as a bar graph or a simple graph some series of data defined by the user. Even in this case, using the *input* command it is possible to choose the type of the graph and to insert the different x-axis and y-axis value (Figure A.5). This script was useful to plot the different k_∞ in function of the coolant density percentage, for example.

B Annex: Examples self-shielding and flux meshes

As mentioned, in this work, and in particular in FR applications, the same mesh (namely the less refined) has been adopted for self-shielding and flux calculations.

An example, extracted from [47], of different meshes for a typical PWR case is shown in Figures B.1.

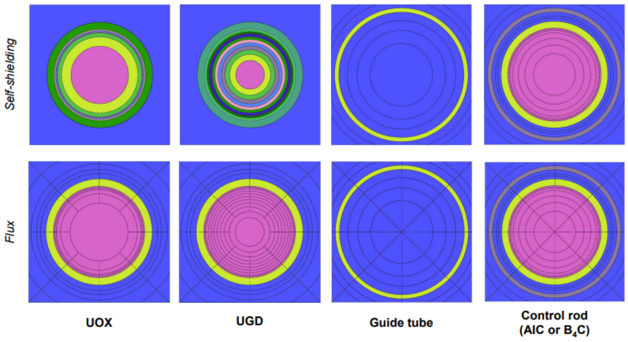


Figure B.1: *Typical self-shielding and flux calculation meshes for PWR cell [47]*

C Annex: Complementary information on the Calculation Scheme

The neutronic simulations follow a specific calculation scheme.

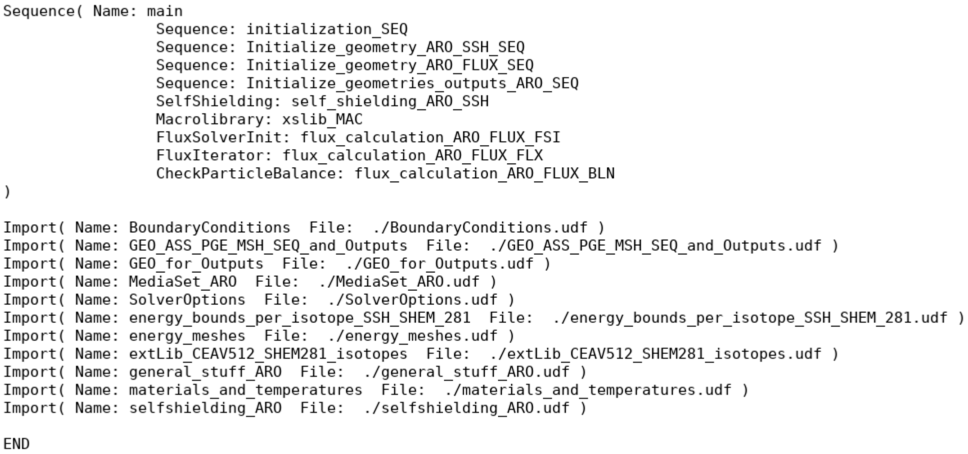
The first part refers to the initialization part where the problem is set in terms of external nuclear data libraries, materials compositions, and geometry constructions at which zones are associated the specific properties, materials and temperatures.

The following part concerns the self-shielding calculation, where the different options are set, such as the isotopes for which the user wants to perform the calculation, the boundary conditions, the self-shielding model and the more specific solver options.

The flux calculation comes after, even in this case all the solver option need to be set up.

The last part of the process involves the preparation of the output file. The useful discretized energy grids are built and so the corresponding output file called.

Using the Python interface, all this steps are called in the same input file, instead the UDF one worked using a *main* file on which are imported the specific inputs corresponding to the specific steps, according to the NEMESI approach. In Figure C.1 is shown the *main* file where it is possible to appreciate the above described calculation chain.



```
Sequence( Name: main
                 Sequence: initialization_SEQ
                 Sequence: Initialize_geometry_ARO_SSH_SEQ
                 Sequence: Initialize_geometry_ARO_FLUX_SEQ
                 Sequence: Initialize_geometries_outputs_ARO_SEQ
                 SelfShielding: self_shielding_ARO_SSH
                 Macrolibrary: xslib_MAC
                 FluxSolverInit: flux_calculation_ARO_FLUX_FSI
                 FluxIterator: flux_calculation_ARO_FLUX_FLX
                 CheckParticleBalance: flux_calculation_ARO_FLUX_BLN
)

Import( Name: BoundaryConditions  File:  ./BoundaryConditions.udf )
Import( Name: GEO_ASS_PGE_MSH_SEQ_and_Outputs  File:  ./GEO_ASS_PGE_MSH_SEQ_and_Outputs.udf )
Import( Name: GEO_for_Outputs  File:  ./GEO_for_Outputs.udf )
Import( Name: MediaSet_ARO  File:  ./MediaSet_ARO.udf )
Import( Name: SolverOptions  File:  ./SolverOptions.udf )
Import( Name: energy_bounds_per_isotope_SSH_SHEM_281  File:  ./energy_bounds_per_isotope_SSH_SHEM_281.udf )
Import( Name: energy_meshes  File:  ./energy_meshes.udf )
Import( Name: extLib_CEAV512_SHEM281_isotopes  File:  ./extLib_CEAV512_SHEM281_isotopes.udf )
Import( Name: general_stuff_ARO  File:  ./general_stuff_ARO.udf )
Import( Name: materials_and_temperatures  File:  ./materials_and_temperatures.udf )
Import( Name: selfshielding_ARO  File:  ./selfshielding_ARO.udf )

END
```

Figure C.1: *Calculation steps and corresponding imported files using the UDF interface*

C.1 Integration of Depletion in the calculation scheme

In case of depletion evaluation an additional part has to be added at the end of the Python input (Figure C.2). First, the same calculations made in stationary conditions are called, then the reactor power or burn-up must be specified and the time steps given in a vector with the corresponding unit.

So at each time step all the quantities evaluated in stationary conditions are updated and properly stocked in the output file.

```
# ---------------------------------------------------------------------------
# build Depletion objects
# ---------------------------------------------------------------------------

def SequenceAtConvergence():
  fluxgeom_SSH.run()
  xslib_MAC.run()
  fluxgeom_FLX.run()
  fluxgeom_FXN.run()
  output_HOM_33.run()
  output_RAT.run()
  output_OUT_33.run()


# only one Depletion object : it is initialized when the object is built
# then execute at each iteration
# energyMesh is now optional (1-group rates by default)
# the sequence that updates the flux for Mendel iterations is now optional (macro and flux calculation by default)

flux_dep_FXN = Apollo3.UDM.FluxNorm( Name = "flux_dep_FXN",
                                     FluxSolverInit = fluxgeom_FSI,
                                     Value = 53.336,
                                     UsePowerDensity = True )

dep_moc_DIT = Apollo3.UDM.Depletion(Name = "dep_moc_DIT",
                                    FluxIterator = fluxgeom_FLX,
                                    DepletionUnity = Apollo3.UDM.Depletion.DAY,
                                    FluxNorm = flux_dep_FXN,
                                    Precision = 0.001,
                                    NbInterpolationMax = 6
)

# then you just have to loop on the time steps, use the new method runOneStep and then execute the python function
depMesh      = Apollo3.std.vfloat([0.0, 20, 100, 250, 340])
#depMesh[:]      = [0.0, 0.36, 0.76, 1.48, 3.0]

for i in range(len(depMesh)-1):
  dep_moc_DIT.runOneStep(depMesh[i], depMesh[i+1])
  SequenceAtConvergence()
# ---------------------------------------------------------------------------
# END END END
# ---------------------------------------------------------------------------
```

Figure C.2: *Depletion calculation steps*

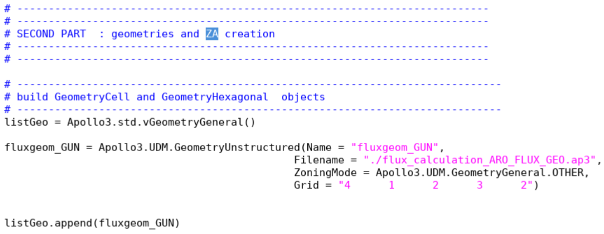
D Annex: Complementary results

D.1 2D fuel cell

D.1.1 Zone Assignment process

In order to properly allocate the materials, temperatures and properties to the different zones within the geometry, a specific *zone assignment* process is required. Depending on how the geometry is given for the neutronic calculation, this process can be made in two different ways.

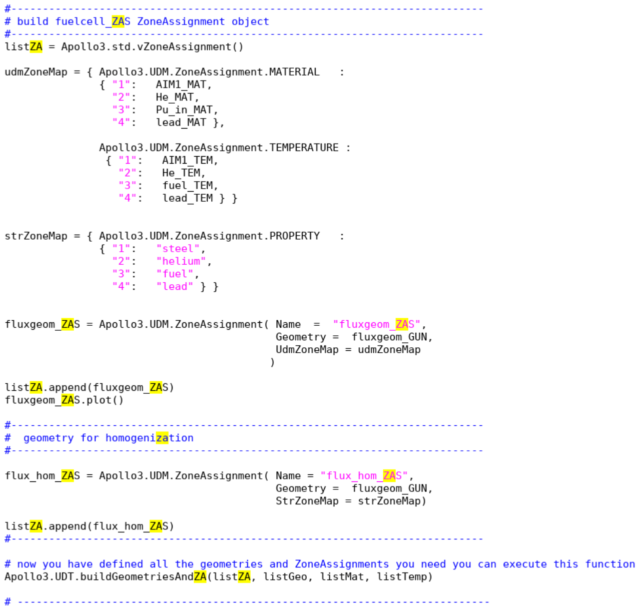
In particular, in the case where the geometry is external (constructed using ALAMOS), it is necessary to report the number geometry grid as written on the .ap3 geometry file, given as input, as shown in Figure D.1.



```
# ----------------------------------------------------------------------
# ----------------------------------------------------------------------
# SECOND PART  : geometries and ZA creation
# ----------------------------------------------------------------------
# ----------------------------------------------------------------------

# -----------------------------------------------------------------------
# build GeometryCell and GeometryHexagonal  objects
# -----------------------------------------------------------------------
listGeo = Apollo3.std.vGeometryGeneral()

fluxgeom_GUN = Apollo3.UDM.GeometryUnstructured(Name = "fluxgeom_GUN",
                                                Filename = "./flux_calculation_ARO_FLUX_GEO.ap3",
                                                ZoningMode = Apollo3.UDM.GeometryGeneral.OTHER,
                                                Grid = "4      1      2      3      2")


listGeo.append(fluxgeom_GUN)
```

Figure D.1: *Zone Assignment process using alamos imported geometry (part 1)*

Each number of the grid corresponds to a specific region and so temperature and property. They are properly associated as shown in Figure D.2 and so the zone assignment can be done using the specific option.



```
#----------------------------------------------------------------------
# build fuelcell_ZAS ZoneAssignment object
#----------------------------------------------------------------------
listZA = Apollo3.std.vZoneAssignment()

udmZoneMap = { Apollo3.UDM.ZoneAssignment.MATERIAL   :
               { "1":   AIM1_MAT,
                 "2":   He_MAT,
                 "3":   Pu_in_MAT,
                 "4":   lead_MAT },

               Apollo3.UDM.ZoneAssignment.TEMPERATURE :
                { "1":   AIM1_TEM,
                  "2":   He_TEM,
                  "3":   fuel_TEM,
                  "4":   lead_TEM } }


strZoneMap = { Apollo3.UDM.ZoneAssignment.PROPERTY   :
               { "1":   "steel",
                 "2":   "helium",
                 "3":   "fuel",
                 "4":   "lead" } }


fluxgeom_ZAS = Apollo3.UDM.ZoneAssignment( Name  =  "fluxgeom_ZAS",
                                           Geometry =  fluxgeom_GUN,
                                           UdmZoneMap = udmZoneMap
                                          )

listZA.append(fluxgeom_ZAS)
fluxgeom_ZAS.plot()

#----------------------------------------------------------------------
#  geometry for homogenization
#----------------------------------------------------------------------

flux_hom_ZAS = Apollo3.UDM.ZoneAssignment( Name = "flux_hom_ZAS",
                                           Geometry =  fluxgeom_GUN,
                                           StrZoneMap = strZoneMap)

listZA.append(flux_hom_ZAS)
#----------------------------------------------------------------------

# now you have defined all the geometries and ZoneAssignments you need you can execute this function
Apollo3.UDT.buildGeometriesAndZA(listZA, listGeo, listMat, listTemp)

# ----------------------------------------------------------------------
```

Figure D.2: *Zone Assignment process using alamos imported geometry (part 2)*

Instead, when the geometry is directly built in the Python script (native geometry), first the geometry is constructed and the different regions of the grid properly numbered and then associated to temperature and properties. At the end the zone assignment object can be properly built as shown in Figure D.3.

```
#---------------------------------------------------------------------------
# build outcell_ZAS ZoneAssignment object
#---------------------------------------------------------------------------
outcell_ZAS = Apollo3.UDM.ZoneAssignment(  Name  =  "outcell_ZAS",
                                           Geometry =  fuelcell_GCE,
                                           StrZoneMap = strZoneMap
)

listZA.append(outcell_ZAS)
outcell_ZAS.plot()


#---------------------------------------------------------------------------
# build shldgeom_ZAS ZoneAssignment object
#---------------------------------------------------------------------------

shldgeom_ZAS = Apollo3.UDM.ZoneAssignment(   Name  =  "shldgeom_ZAS",
                                             Geometry =  fluxgeom_GHX,
                                             UdmZoneMap = { Apollo3.UDM.ZoneAssignment.ZONEASSIGN : { "0" : fuelcell_ZAS } }
)

listZA.append(shldgeom_ZAS)
shldgeom_ZAS.plot()

#---------------------------------------------------------------------------
# build outgeom_ZAS ZoneAssignment object
#---------------------------------------------------------------------------
outgeom_ZAS = Apollo3.UDM.ZoneAssignment( Name  =  "outgeom_ZAS",
                                          Geometry =  fluxgeom_GHX,
                                          UdmZoneMap = { Apollo3.UDM.ZoneAssignment.ZONEASSIGN : { "0" : outcell_ZAS } }
)

listZA.append(outgeom_ZAS)
outgeom_ZAS.plot()

#---------------------------------------------------------------------------

# now you have defined all the geometries and ZoneAssignments you need you can execute this function
Apollo3.UDT.buildGeometriesAndZA(listZA, listGeo, listMat, listTemp)

# ---------------------------------------------------------------------------
```

Figure D.3: *Zone Assignment process using the native geometry*

D.1.2 Additional useful graphs

The Figure D.4 shows the microscopic absorption cross-sections at 33 groups. It is interesting to notice how, in this case, the adopted library plays an important role on the obtained results.

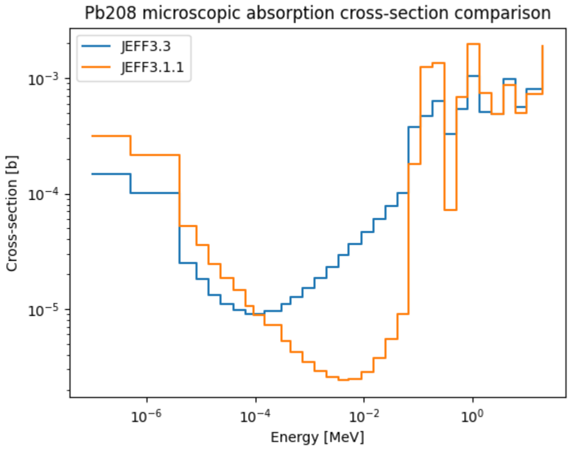


Figure D.4: *Microscopic cross-section at 33g of the* Pb_{208}

D.2 2D fuel assembly

D.2.1 Solver options investigation

As mentioned above, different solver options were investigated to understand their impact on the results. The Figure D.5 compares the main different TDT options adopted in flux and self-shielding calculations. Additionally, also the anisotropy effect was evaluated.

All of these options are not detailed investigated for the purpose of this work but they are properly discussed in [29] and [100].

```
# ---------------------------------------------------------------------------
# build SolverOptionsTDT CPM object for selfshielding
# ---------------------------------------------------------------------------

tdtcpm_SOT = Apollo3.UDM.SolverOptionsTDT(Name = "tdtcpm_SOT",
                                          SolverOptionGeneral = gen_SOG)

tdtcpm_SOT.setTRK_HorizontalAngles(24);
tdtcpm_SOT.setTRK_MaxNumberMeshesPerPeriod(100);
tdtcpm_SOT.setTRK_SkipVacuum(True);
tdtcpm_SOT.setTRK_TransversalIntegrationStep(0.04);
tdtcpm_SOT.setTRK_VacuumLimit(1.0E-07);

tdtcpm_SOT.setTKM_ThreadNumber(10);
tdtcpm_SOT.setTKM_DiscretizationMethod(Apollo3.UDM.SolverOptionsTDT.Pij);

tdtcpm_SOT.setPIJ_Normalization(Apollo3.UDM.SolverOptionsTDT.KeepPij)
tdtcpm_SOT.setPIJ_KeepCollisionMatrixOnMemory(True)
tdtcpm_SOT.setPIJ_MatrixPrint(0)

tdtcpm_SOT.update()
# ---------------------------------------------------------------------------

# ---------------------------------------------------------------------------
```

```
# ---------------------------------------------------------------------------
# build SolverOptionsTDT CPM object for selfshielding
# ---------------------------------------------------------------------------

tdtcpm_SOT = Apollo3.UDM.SolverOptionsTDT(Name = "tdtcpm_SOT",
                                          SolverOptionGeneral = gen_SOG)

tdtcpm_SOT.setTRK_HorizontalAngles(24);
tdtcpm_SOT.setTRK_MaxNumberMeshesPerPeriod(100);
tdtcpm_SOT.setTRK_SkipVacuum(True);
tdtcpm_SOT.setTRK_TransversalIntegrationStep(0.01);
tdtcpm_SOT.setTRK_VacuumLimit(1.0E-07);

tdtcpm_SOT.setTKM_ThreadNumber(4);
tdtcpm_SOT.setTKM_DiscretizationMethod(Apollo3.UDM.SolverOptionsTDT.Pij);

tdtcpm_SOT.setPIJ_Normalization(Apollo3.UDM.SolverOptionsTDT.KeepPij)
tdtcpm_SOT.setPIJ_KeepCollisionMatrixOnMemory(True)
tdtcpm_SOT.setPIJ_MatrixPrint(0)

tdtcpm_SOT.update()
# ---------------------------------------------------------------------------

# ---------------------------------------------------------------------------
```

```
# ---------------------------------------------------------------------------
# ---------------------------------------------------------------------------
# FOURTH PART  : flux calculation
# ---------------------------------------------------------------------------
# ---------------------------------------------------------------------------

# ---------------------------------------------------------------------------
# build SolverOptionsTDT objects for MOC calculation
# ---------------------------------------------------------------------------
tdt_SOT = Apollo3.UDM.SolverOptionsTDT(Name = "tdt_SOT",
                                       SolverOptionGeneral = gen_SOG)

tdt_SOT.setTRK_HorizontalAngles(24)
tdt_SOT.setTRK_VerticalAngles(4)
tdt_SOT.setTRK_MaxNumberMeshesPerPeriod(100)
tdt_SOT.setTRK_VerticalMethod(Apollo3.UDM.SolverOptionsTDT.GLegEven)

tdt_SOT.setCHM_SolutionType(Apollo3.UDM.SolverOptionsTDT.BCGST)
tdt_SOT.setCHM_MaxGroupNumberInMemory(3000)
tdt_SOT.setCHM_MaxGroupNumberForCoef(3000)

tdt_SOT.setTRK_TransversalIntegrationStep(0.04)

tdt_SOT.setTKM_DomainDecompositionLevels(1)
tdt_SOT.setTKM_ExactIntegrationMethod(1)
tdt_SOT.setTKM_MaxGridRegionsFirstLevel(20)
tdt_SOT.setTKM_ThreadNumber(10)
tdt_SOT.setTKM_DiscretizationMethod(Apollo3.UDM.SolverOptionsTDT.Characteristics_ASA)

tdt_SOT.setFXI_OuterMaxIteration(50)
tdt_SOT.setFXI_OuterAcceleration(Apollo3.UDM.SolverOptionsTDT.OutersBySynthetic)
tdt_SOT.setFXI_InnerAcceleration(Apollo3.UDM.SolverOptionsTDT.InnersByLanczos)
tdt_SOT.setFXI_OuterMaxDSA(5)

tdt_SOT.update()
```

```
# ---------------------------------------------------------------------------
# ---------------------------------------------------------------------------
# FOURTH PART  : flux calculation
# ---------------------------------------------------------------------------
# ---------------------------------------------------------------------------

# ---------------------------------------------------------------------------
# build SolverOptionsTDT objects for MOC calculation
# ---------------------------------------------------------------------------
tdtmoc_FSO = Apollo3.UDM.SolverOptionsTDT(Name = "tdtmoc_FSO",
                                          SolverOptionGeneral = gen_SOG)

tdtmoc_FSO.setTRK_HorizontalAngles(24);
tdtmoc_FSO.setTRK_VerticalAngles(4);
tdtmoc_FSO.setTRK_MaxNumberMeshesPerPeriod(100);

tdtmoc_FSO.setCHM_MaxGroupNumberInMemory(3000);
tdtmoc_FSO.setCHM_MaxGroupNumberForCoef(3000);

tdtmoc_FSO.setTRK_TransversalIntegrationStep(0.04);

tdtmoc_FSO.setTKM_DomainDecompositionLevels(1);
tdtmoc_FSO.setTKM_ExactIntegrationMethod(1);
tdtmoc_FSO.setTKM_MaxGridRegionsFirstLevel(20);
tdtmoc_FSO.setTKM_ThreadNumber(4);
tdtmoc_FSO.setTKM_DiscretizationMethod(Apollo3.UDM.SolverOptionsTDT.Characteristics_ASA);

tdtmoc_FSO.setFXI_OuterMaxIteration(50);
tdtmoc_FSO.setFXI_OuterAcceleration(Apollo3.UDM.SolverOptionsTDT.OutersBySynthetic);
tdtmoc_FSO.setFXI_InnerAcceleration(Apollo3.UDM.SolverOptionsTDT.InnersByLanczos);

tdtmoc_FSO.update()
```

Figure D.5: *Comparison between two different solver options settings*

D.2.2 Homogenized fuel assembly

The Figure D.6 shows the coarser mesh adopted during the calculation of the homogenized fuel assembly case.

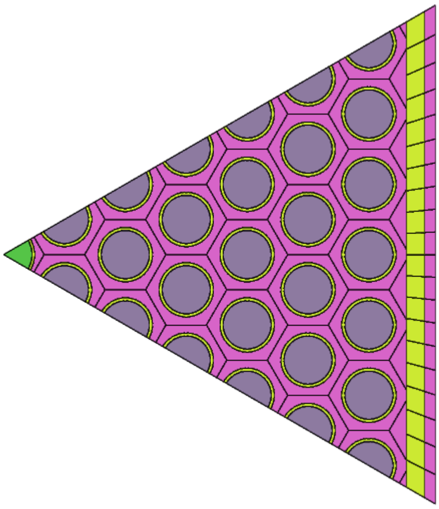


Figure D.6: *Homogenized fuel assembly structure*

In order to minimize the relative difference in reactivity in between the fuel assembly homogeneous case and the heterogeneous one, a finer mesh has been adopted for the homogenized fuel pin. In particular two additional circles as shown in Figure D.7. The obtained k_∞ is equal to 1.29179 corresponding to a difference in reactivity of 23 pcm, without improuving further the results.

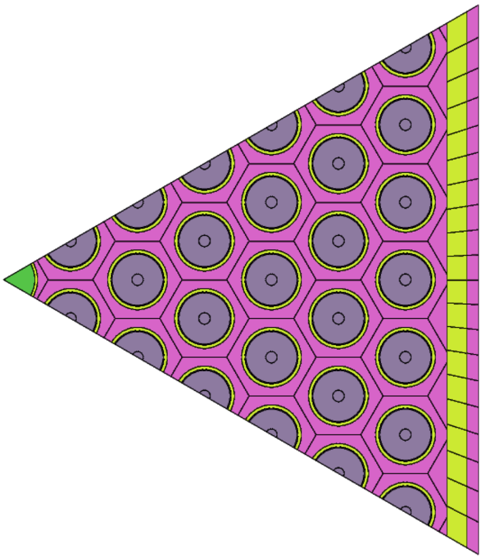


Figure D.7: *Homogenized fuel assembly finer mesh*

D.2.3 Additional verification

Also, in order to verify the values obtained through the deterministic simulation, the macroscopic cross-sections have been calculated also using the equation D.1. The corresponding relative errors are negligible, of about 0% .

$$\Sigma = \sum_{i=1}^{I} \sigma_i N_i \quad [1/cm] \tag{D.1}$$

Where I is the total number of isotopes in the specific material, σ is the corresponding microscopic cross-section [barn] and N refers to the isotope concentration $[\frac{atoms}{barn \cdot cm}]$.

D.3 3D fuel cell

In order to start to investigate the 3D effects, a 3D pin structure has been set up profiting of the new capabilities of APOLLO3® of having access to MOC3D solver [77], [28]. The results obtained are preliminary and indicated here just as an example of possible application of the 3D model. Further investigations are foreseen in a following phase of the study.

The 3D cell model (radial reflective boundary conditions imposed) is depicted in Figure 5.12. Three identical fuel zones have been identified for taking into account the evolution of the temperature as indicated in the OECD/NEA benchmark. Several tests have been performed and some discrepancies remain with respect to Monte Carlo calculations. Further studies will help on investigate more in detail the reasons of those discrepancies with the support of the developers. However, the tendencies obtained are reported in this Annex.

Figure D.8 shows the reactivity evolution as function of the coolant density. The perturbation has been imposed in the overall active zone obtaining the linear behavior as expected. Under voided conditions (hypothetical case for the lead cooled systems) the total reactivity variation is of $\sim$ 4000 pcm.

The density has been changed non-simultaneously in the three axial parts in which the active zone has been divided. The three zones are characterized by the same temperatures, compositions and height. The results obtained are shown in Table D.1. The effects are almost additive

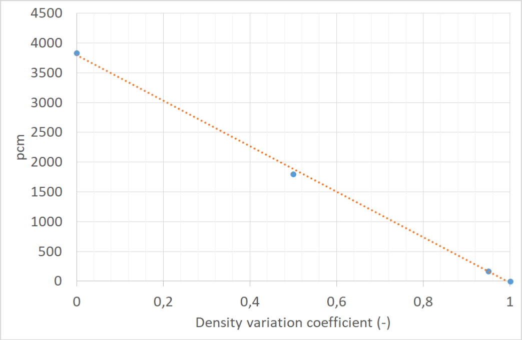


Figure D.8: *Coolant density variation based on the 3D cell model of the LFR configuration*

and the major contribution comes to the core center as expected ($\sim$ 44%). The upper and lower parts contributes to $\sim$ 28% each with a slightly difference due to the not symmetrical configuration of the case (Figure 5.12). The calculations are carried out using JEFF3.1.1 data library and Tone self-shielding method.

Table D.1: *3D density variation effects*

Fuel axial zones	lower	middle	upper
k_{eff}	1.31215	1.32394	1.31361
rho (pcm)	1090	1769	1175
%	27	44	29

These tenancies need to be consolidated once the discrepancies with respect to Monte Carlo calculations will be solved.