Treball de Fi de Màster

Màster Universitari en Enginyeria Nuclear

Monte Carlo time dependent modelling of the MYRRHA core based on SERPENT2

MEMÒRIA

April 29, 2022

Autor: Bernat Ballester Vázquez

Directors: Augusto Hernández Solís (SCK CEN)
Pablo Romojaro Otero (SCK CEN)
Alfredo de Blas del Hoyo (UPC)

Convocatòria: Maig 2022

ETSEIB

sck cen
Belgian Nuclear Research Centre

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

UPC

Glossary

- **Analog simulation:** Monte Carlo calculation that simulates all the physical phenomena involved in a history without the use of any variance reduction techniques.
- **Criticality calculation:** Transport simulation in which the source of neutrons includes fissionable material, such a nuclear reactor. Used to solve steady-state problems.
- **Decay constant (λ):** Defines the rate at which the precursors present in the system decay to create delayed neutrons. Usually, precursors are grouped according to their decay constant.
- **Delayed neutron:** Neutron born from the decay of a fission product. They represent a small fraction of the neutrons present in the reactor, but are critical for its control since they increase the time between consecutive neutron generations.
- **Delayed neutron fraction (β):** Fraction of delayed neutrons present in a system among the total neutron population.
- **Dragon:** Deterministic code that simulates the neutronic behaviour of a unit cell or a fuel assembly in a nuclear reactor.
- **Dynamic Monte Carlo simulation:** Monte Carlo simulation that takes into account the time-dependency of the problem. It can be used to study transient scenarios in nuclear reactors.
- **Dynamic reactivity:** Reactivity that has been calculated considering time as a variable. In the context of this project, a dynamic reactivity is the reactivity obtained as a solution of a time-dependent Monte Carlo simulation.
- **Effective multiplication factor (k_{eff}):** Ratio of the neutron population at generation $(\mathrm{n+1})$ to the population at generation (n). If $k_{eff} = 1$, the neutron population remains constant with time, and the reactor is said to be critical. Reactors with $k_{eff} < 1$ are subcritical, and those with $k_{eff} > 1$ are supercritical.
- **Fixed-source calculation:** Monte Carlo simulation in which the distribution of the neutron source is known. This is the case of dynamic simulations, as they start from a precalculated neutron source.
- **k-eigenvalue calculation:** method to solve criticality calculations in time-dependent systems by modifying the fission term so that the neutron population remains constant and the problem becomes steady-state.
- **Mean neutron generation time (Λ):** Average amount of time that elapses between the

creation of two consecutive neutron generations.

- **Perturbation theory:** Method to solve time-dependent problems by treating the variations in the configuration of the system as “perturbations" of a simplified case, which in the case of a nuclear reactor would be the critical configuration. It can be used to calculate the reactivity variation between the unperturbed and perturbed states of the system.
- **Perturbed system:** System that has suffered a modification in one or more of its parameters which has brought it away from criticality.
- **Point kinetics approximation:** Reduced order representation of a system in which the shape of the neutron flux and the neutron population distribution is not taken into account. It results in a system of equations that describes the rate of change with time of the neutron population and the density of the precursor groups.
- **Precursor:** Unstable fission product that eventually decays producing a delayed neutron.
- **Prompt critical:** State of a reactor in which the prompt neutrons generated in fissions are enough to maintain the reaction chain, so the delayed neutrons are not required. As a result, the growth rate of the energy produced is extremely fast, which makes the reactor very difficult to control.
- **Prompt neutron:** Neutron born from a fission reaction. Prompt neutrons are the vast majority of the neutrons present in a reactor, but in normal operating situations cannot sustain the chain reaction on their own, having to rely on the contributions of delayed neutrons instead.
- Serpent: Multi-purpose three-dimensional continuous-energy Monte Carlo particle transport code, developed at VTT Technical Research Centre of Finland.
- **Static reactivity:** Approximation of the time-dependent reactivity that is calculated neglecting the time variable. In the context of this project, a static reactivity is the reactivity calculated with either of the static approaches proposed
- **System:** Each of the model used in the calculations performed in this work, defined by a geometry, materials, boundary conditions, properties, etc.
- **Unperturbed system:** System that is in its critical state.

Resum

Els fenòmens transitoris són de gran importància per a l'anàlisi de seguretat en nous dissenys de reactors. Actualment, amb MYRRHA en procés de ser llicenciat, enginyers i científics a l'SCK CEN estan fent tots els estudis de seguretat del disseny per demostrar que compleix la regulació. Part d'aquests estudis són els càlculs del comportament dinàmic del reactor en escenaris transitoris, que van des del moviment de barres de control que té lloc regularment durant l'operació del reactor fins a condicions d'accident.

És essencial conèixer l'evolució temporal de la reactivitat del nucli, ja que és un paràmetre necessari per a determinar la població de neutrons, la potència o el comportament termo-hidràulic, però calcular-lo no és senzill. Normalment s'opta per aproximar la reactivitat en funció del temps definint una reactivitat estàtica, que no depèn del temps i es pot calcular més fàcilment. Aquesta reactivitat estàtica s'utilitza com a input a les equacions de cinètica puntual (point kinetics equations), per a mesurar l'evolució de la població, o en codis termo-hidràulics, per a estudiar els efectes de feedback i altres paràmetres rellevants per a la seguretat.

Tot i això, les millores en potència computacional i el desenvolupament d'eines dinàmiques en codis de Monte Carlo que han tingut lloc els darrers anys permet calcular la reactivitat, així com altres paràmetres d'interès, en funció del temps. Donat que els codis de Monte Carlo no introdueixen cap simplificació en els càlculs, permeten obtenir els resultats més propers a la realitat possibles, cosa que fa el mètode de Monte Carlo una opció interessant.

En aquest projecte s'explora la possibilitat d'utilitzar el mètode de Monte Carlo en funció del temps per a l'anàlisi de seguretat de MYRRHA. La metodologia per a fer els càlculs, així com l'aplicació en diversos casos utilitzant diferents models, s'hi presenta. També s'hi inclou una valoració de les deficiències i els reptes que s'han de superar per tal d'utilitzar el mètode en el futur.

Actualment, però, el mètode de Monte Carlo encara no és aplicable, i els mètodes estàtics encara es prefereixen. En un esforç per a proporcionar a l'equip de seguretat de MYRRHA una eina fiable per al càlcul de la reactivitat en un escenari transitori, el mètode utilitzat actualment a l'SCK CEN ha estat avaluat i comparat amb les simulacions dinàmiques de Monte Carlo. Addicionalment, un mètode alternatiu per al càlcul de la reactivitat estàtica, derivat de la teoria de perturbació, s'ha proposat i estudiat.

Abstract

Transient phenomena are of major importance in the safety analysis of new reactor designs. At the moment, while MYRRHA is in the process of being licensed, engineers and scientists at SCK CEN are performing all the necessary safety studies in the design to ensure it complies with the regulations. Part of such studies are the calculations of the dynamic behaviour of the reactor in transient scenarios, which can range from movements of control rods bundles that take place regularly in the operation of a nuclear reactor to accident conditions.

Knowing the time-dependent evolution of the reactivity in the core is essential, as it is a parameter that is required for determining the neutron population, power or thermal-hydraulic behaviour of the reactor, but calculating it is not a simple task. Usually, the approach followed is to approximate the time-dependent reactivity by determining a static reactivity instead, which is not dependant on time and can be more easily calculated. This static reactivity is then used as an input in the point kinetics equations, to measure the evolution of the neutron population, and in thermal-hydraulic codes, to study the feedback effects and other safety-related parameters.

However, in the last years the general improvements in computational power and the development of dynamic tools in various Monte Carlo neutron transport codes offer the possibility of calculating the reactivity, as well as other parameters of interest, using a fully time-dependent approach. As Monte Carlo method does not introduce major simplifications in the calculations, they allow to reach best-estimate solutions, which makes the Monte Carlo approach an interesting possibility.

In this work, the possibility of using time-dependent Monte Carlo simulations for the safety analysis of the MYRRHA reactor is explored. The approach for performing such calculations, together with case studies using various models are presented. An assessment of the main challenges and shortcomings that need to be overcome for the use of the method in the future is also included.

At the moment, though, the Monte Carlo method is not yet applicable in safety studies, and static methods are preferred instead. In an effort to provide the safety team with a reliable tool to calculate the reactivity of the core in a transient situation, the method currently used at SCK CEN is evaluated and compared with the dynamic Monte Carlo simulations. Additionally, an alternative approach for the calculation of the static reactivity, derived from perturbation theory, is proposed and studied.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

In the field of reactor physics, Monte Carlo simulations are a method that is widely used to solve neutron transport problems and to calculate relevant parameters of nuclear reactors in static situations. Nevertheless, transient scenarios have been traditionally calculated by means of deterministic approaches [2], or static approximations that neglect the time-dependency of the problem. Such scenarios are of great importance in the safety analysis required for the commissioning of new reactor designs, in which the dynamic evolution of the reactor after a perturbation, e.g. the insertion or withdrawal of control rods, a change of density of the coolant, an accident that induces a change in the parameters of the reactor, etc., is introduced in the system is studied.

The reasons cited for the selection of static methods are usually related to the excessive amount of time and computational resources needed to obtain solutions using Monte Carlo simulations, which are even larger when the time-dependency is introduced to the problem. In the last century, the limited computational power of the computers available lead to the conclusion that it was not possible to apply Monte Carlo techniques to transient problems, and that alternative approaches were preferable due to the fact that they are not as computationally demanding. Nevertheless, static methods have a major disadvantage since considering a transient problem to be time-independent introduces a major simplification to the neutron transport equation, so the accuracy of the result needs to be proved.

However, there is an alternative to the static approximation of the reactivity. In the last decades the improvements in the field of computation have lead to much more powerful computers that are currently capable of solving time-dependent problems using Monte Carlo methods in a relatively short time and with good statistics, so an effort has been made to develop Monte Carlo codes that solve such problems. For instance, dynamic tools have been implemented in the Monte Carlo codes *OpenMC* [3] [4] [5] or *Serpent* [6]. They pose an interesting possibility, as using the Monte Carlo approach allows to find more accurate solution thanks to the fact that it does not require any major simplification of the problem to find a solution, contrary to the static approach.

The use of Monte Carlo as a tool to solve time-dependent problems has increased in the last years, as various studies have been performed to assess the possibilities of applying it by comparing the results obtained with the code to experimental measurements [7] [8]. These studies show promising results, as accurate solutions are found using complex, 3-dimensional heterogeneous models of nuclear reactors. In the future, with increasingly powerful computers, it is possible that fully time-dependent Monte Carlo simulations can substitute the point kinetics equations, widely used today in the study of reactor kinetics. Nevertheless, the point kinetics approach, together with other static methods [9] [10], has been traditionally used and is still

preferred today in the design of nuclear reactors, including MYRRHA.

This project was born from the request of the safety group in charge of the licensing of MYRRHA to have a tool for the calculation of a time-independent reactivity that correctly approximates the real reactivity of a system while being conservative, together with the interest in exploring time-dependent Monte Carlo methods applied to safety calculations. Therefore, the first of the motivations of this thesis is to contribute to the research performed in the use of Monte Carlo simulations in time-dependent problems. This is to be done by studying transient scenarios in various models, including a heterogeneous model of the MYRRHA reactor currently being developed at SCK CEN. Also, as part of the development of reliable tools for the approximation of the dynamic reactivity, another motivation of the project is to study various approaches for calculating the static reactivity and their applicability in different transient scenarios by comparing them to the time-dependent solutions.

1.2 Scope of the thesis

The thesis focuses on the purely neutronics aspects of transient scenarios in reactors. Most of the calculations are made with the Monte Carlo code *Serpent* [11]. It is used to solve time-dependent problems and for calculating kinetic parameters that are part of the point kinetics equations. Nevertheless, the code has shortcomings, for instance in the calculation of adjoint fluxes, which cannot be obtained as an output. Since they are necessary for the calculations intended in the thesis, an alternative code has been included: the deterministic code *Dragon*. The inclusion of a deterministic code allows to obtain the required parameters, and also widens the scope of the project.

The calculations are performed in three models. The first is the heterogeneous MYRRHA 1.6 model. It is a 3-dimensional model of the MYRRHA reactor, which contains all the components of the core. The subcritical configuration of the reactor and the contribution of the linear accelerator are not of interest in this project, so the model used does not include it. Instead, the critical configuration of MYRRHA is used. Additionally, two slab models have been used. They are geometrically identical, the main difference is on the energy spectrum of the neutron flux: one has a thermal spectrum, while the other is fast. All the models are in open loop, as the reactivity feedbacks are not of interest. Neglecting those allows to study and make comparisons of the neutronic behaviour of fast and thermal systems.

Two approaches are to be followed for the calculation of the dynamic behaviour of the systems. The first consists in the use of time-dependent Monte Carlo simulations, while the second relies on the calculation of the reactivity variation of the system by means of static calculations to be used in the point kinetics equations. A comparison between the methods is performed in the thesis.

1.3 Objectives

This thesis has the purpose of studying the possibility of using the dynamic Monte Carlo method for calculations of transients relevant for the safety analysis of MYRRHA that is currently taking place at SCK-CEN, specifically using *Serpent*, a Monte Carlo code that was originally created as a neutron transport code for reactor physics applications [12]. Originally its main purpose was group constant generation, but further development has included other capabilities to the code, such as time-dependent calculations, which makes the code suitable for the purposes of the thesis.

As part of the effort to provide a tool to effectively calculate the reactivity change in a reactor during a transient scenario, this project also has the objective of presenting a methodology that allows the calculation of such parameter using static calculations through the application of perturbation theory to the neutron transport equation. The static reactivity calculated via perturbation theory is to be compared with the same parameter calculated through the $\Delta\lambda$ method, currently used at SCK CEN, in order to assess the advantages and disadvantages of each approach.

Since the result obtained using static methods does not take into account the time-dependent evolution of the system, the reactivity obtained is an approximate result, and therefore it is of key importance from a safety point of view that it is conservative when compared to the actual time-dependent reactivity evolution. Hence, testing the conservativeness of the method is another objective of the thesis. This is to be done by comparing the result obtained from perturbation theory, the so-called static reactivity, with the time-dependent reactivity that is obtained from dynamic Monte Carlo simulations, which can be considered the closest possible to reality since Monte Carlo is a best estimate tool.

With these objectives in mind, the final aim of the project is to make an evaluation of the results obtained and provide reference cases that include a variety of perturbations using different models. With these cases, initial conclusions regarding the suitability of the methods to calculate the static reactivity in transient scenarios and the use of the point kinetics equations can be reached, both qualitatively and quantitatively. This conclusions are expected to contribute to explore the capabilities of the static approximations, so that the safety analysis of new reactor designs can be improved in the future. Ultimately, this thesis wants to serve as a first step for the implementation of time-dependent Monte Carlo simulations in the safety analysis of MYRRHA. Although it is still a future prospect, studying the current advantages and limitations of the method, as well as providing knowledge on how to run such simulations and the parameters of importance, are key aspects from which to continue working, and this thesis has the objective of contributing in this sense.

1.4 The MYRRHA project

MYRRHA (Multi-purpose HYbrid Research Reactor for High-tech Applications) [13] is a project originated in 1998 in SCK-CEN, the Belgian Nuclear Research Center. It is intended as a multi-purpose research plant, with applications such as the production of radioisotopes used in multiple medical applications, the transmutation of spent nuclear fuel in order to reduce the radiotoxicity and lifetime of the waste that is stored in deep geological disposal facilities, and the study and research of new materials to be used in nuclear fusion reactors [14].

It is a one-of-a-kind project, the world's first large scale accelerator driven system (ADS) [15] [16]. It consists in a generation IV reactor [17] with a subcritical configuration that is driven by a linear accelerator. Since the reactor is subcritical, the fissile material in the core is not able to create the required amount of fission reactions to have a self-sustaining reaction chain. To provide the necessary amount of neutrons to keep the flux constant with time, the linear accelerator accelerates protons that are then directed to a target, in which the high energy protons and the nuclei present in the target interact through spallation reactions, which result in the creation of neutrons. The use of an ADS contributes to the safe operation of the reactor, since the core is not able to have a self-sustained chain and therefore turning the proton accelerator off ensures the shut down of the reactor. Nevertheless, the design of the reactor also includes the possibility of a critical configuration, so MYRRHA is also able to operate in a self-sustained way, just like a conventional fission reactor.

The reactor itself has a power output of 100 MWth, and is cooled by lead bismuth eutectic (LBE) [18]. The use of LBE has several significant benefits for the design: it has a low melting point (123 ºC), which allows to operate at a relative low temperature limiting the corrosion problems [19]. Since the coolant is a metallic alloy it has a high heat conductivity at atmospheric pressure, so the vessel does not need to be pressurized for the reactor to operate. The coolant is also practically invisible to neutrons, so there is no neutron thermalization and the flux has a fast energy spectrum. LBE can also be used as the spallation target for the generation of the neutrons required to achieve criticality. Finally, the properties of the LBE allow for the use of various types of fuel, such as mixed oxide fuels (MOX). The fuel may also include a fraction of long-lived minor actinides that are present in the spent fuel of other reactors, which can be burned with transmutation reactions, reducing the toxicity of spent fuel.

The project will be carried out in three phases [20]: in the first phase the first section of the linear accelerator, that will produce protons with an energy up to 100 MeV, will be built. The proton target facility and the fusion target station will also be built, and the reactor pre-licensing will be completed. It is expected to be finished in 2026. The second phase consists in the extension of the accelerator to reach the 600 MeV required to drive the reactor, and is scheduled to finish by 2033. Finally, the third phase consists in the construction of the reactor itself, which will be commissioned in 2036, provided that all phases of the project are completed according to the schedule.

2 Theoretical frame

2.1 Formulation of the neutron transport equation

The physics of nuclear reactors are determined by the distribution of the neutrons present in the system, i.e., the neutron flux distribution, with respect to three parameters: energy, space and time. The neutron transport equation is the governing equation of the movement of neutrons inside a nuclear reactor [1]. The behaviour of neutrons inside a nuclear reactor is similar to that of an ideal gas [21], so the neutron transport equation takes the form of the linearised Boltzmann equation. It is obtained by studying the rate at which neutrons enter and leave a differential phase space element, determined by a position, an orientation and an energy (dr, $d\Omega$, , dE), as shown in figure 1, in which a particle in a differential volume dV moves along direction Ω with an energy between E and E + dE.

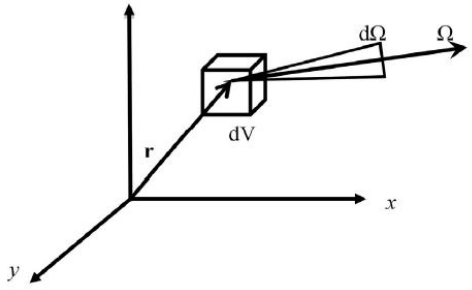


Figure 1: *Phase space element. Source:* [1].

Using a phase space element such as the one described by figure 1, one can derive the neutron transport equation by considering the rate of gain and loss of neutrons, which are defined by the following mechanisms:

Gain mechanisms:

- Neutrons of energy E and direction Ω entering the volume V.
- Neutrons born in volume V from fissions or from external neutron sources.
- Neutrons with energy E′ and direction Ω' scattered to (E,Ω).

Loss mechanisms.

- Neutrons leaking outside of volume V.
- Neutrons of energy E and direction Ω being scattered to energy and direction (E′, Ω').
- Neutrons being absorbed in volume V.

Considering these mechanisms, the neutron transport equation becomes:

$$\frac{1}{v}\frac{\partial\Psi(r,E,\hat{\Omega},t)}{\partial t}+\left(\Omega\cdot\nabla+\Sigma_t(r,E)\right)\Psi(r,E,\hat{\Omega},t)= \\ =\iint\Sigma_s(E'\rightarrow E,\Omega'\rightarrow\Omega)\Psi(r,E,\hat{\Omega},t)d\Omega' dE'+ \\ +\chi(E)\iint\nu\Sigma_f(r,E')\Psi(r,E,\hat{\Omega},t)d\Omega dE'+S(r,E,\hat{\Omega},t) \quad (2.1)$$

Where $\Psi(r,\hat{\Omega},E,t)$ is the angular neutron flux, $S(r,\Omega,E,t)$ is the external neutron source, $\Sigma_t(r,E)$, $\Sigma_f(r,E)$ and $\Sigma_s(r,E)$ are the total, fission and scatter cross sections, ν is the amount of neutrons produced per fission, $\chi(E)$ is the probability density function of neutrons born from fission to have energy E and v is the neutron velocity.

The first term of the left hand side in equation 2.1 accounts for the change of the neutron density in the system with time, while the second and third terms account for the leakage of neutrons and the losses due to collisions, respectively. On the right hand side, the first term relates to the gain of neutrons due to scattering, i.e., neutrons that are scattered into the energy and direction range of interest, the second terms relates to the gains of neutrons due to fissions and, finally, the third term accounts for the gains due to external sources of neutrons.

2.2 Solution of the neutron transport equation

To reach the solution of the neutron transport equation there are two main approaches that can be used. One of them is the Monte Carlo method, which solves the equation with the use of statistic techniques, while the other consists in obtaining a deterministic solution by solving the neutron transport equation in an analytical way.

Regarding the time dependency of the problem, it is possible to distinguish between two types of solution to the neutron transport problem: static and dynamic. On one hand, the static approach eliminates the time dependency from the neutron transport equation, instead, and provides the solution of a time-independent system. Time can be removed from the transport equation using either stochastic or deterministic methods, the most relevant for this thesis being discussed in sections 2.2.1 and 2.2.2. On the other hand, the dynamic approach considers the time as a variable of the neutron transport problem and solves it in a physically accurate way, providing a solution that reflects the actual time dependence of the problem. The dynamic approach using Monte Carlo methods is discussed in section 2.2.3.

2.2.1 Static approach using the Monte Carlo method

The Monte Carlo method is a statistic technique that allows to solve physical problems that involve random processes for which the probability density function is known [22]. An example of such problems is the transport of neutrons through a medium, for instance a nuclear reactor,

which is the focus of this thesis. To do so, it relies on randomly or pseudo-randomly generated numbers that are used to determine the outcome of the random events (e.g. type of interaction, length of the path between interactions, direction of the neutron after an interaction, etc.). In this way it is possible to simulate the life of a neutron from the moment it is born until the moment it disappears, either because it has been absorbed or it has travelled outside of the domain of the simulation. The lifetime of an individual particle is known as a history. Since the Monte Carlo method is a statistical process, the results obtained have a certain uncertainty associated, which depends on the number of repetitions performed in the calculation, i.e, the number of histories simulated: the more histories are simulated in a Monte Carlo calculation, the more the statistical uncertainty can be reduced. After several histories have been simulated, the outcomes of all of them are averaged to obtain a solution of the problem. One of the main advantages of the method is that, since the problem is solved as a mean of numerous random situations that fulfil the neutron transport equation, no major assumptions or simplifications are introduced, which makes the solution from a Monte Carlo calculation a best estimate solution, i.e., the best possible approximation to the real solution of the problem. Nevertheless, since the accuracy of the solution depends on the amount of histories that are simulated, as well as the complexity of the model that is being used, in some cases the large number of histories required to obtain the desired solution can result in the computational time being exceedingly long, which is a disadvantage of the method when compared to other approaches to solve the neutron transport equations.

In a neutron transport problem, the equation that one wants to sove is the neutron transport equation (see equation 2.1), repeated in equation 2.2 without the variables on which each parameter depends for better readability. Note that the external source term is removed since it is considered that the fission chain in the reactor is self-sustained, and the fission and scattering gain terms are united in a single gain term.

$$\frac{1}{v}\frac{\partial \Psi}{\partial t} + \Omega \cdot \nabla \Psi + \Sigma_t \cdot \Psi = \iint \left[\chi \nu \Sigma_f + \Sigma_s\right] \Psi dE' d\Omega' \tag{2.2}$$

There are two methods to solve static criticality problems using the Monte Carlo approach, namely the k-eigenvalue method and the α-eigenvalue method [23]. These two methods differ in the way in which the time dependency of the first term in the left hand side of equation 2.2 is treated [24].

As the problem that the method is trying to solve is static, and therefore time-independent, the first term in equation 2.2 is removed. Nevertheless, only critical systems are truly independent of time, as the neutron population is constant at all times and therefore the term that contains the time derivative is cancelled. In any other case, i.e. for a supercritical or subcritical system, the neutron population varies with time, so a method to remove time-dependency is required. This is done by applying population control to the system. Population control refers to the action of

either creating or eliminating neutrons from the system after each time step of the simulation, so that the total population remains constant. For example, if a system is supercritical, the neutron population tends to increase as the population of generation *n+1* is higher than the population of generation *n* by a factor k_{eff}. Therefore, after a given period of time the neutron population will have increased with respect to the population that was in the system at the beginning of the period. What the k-eigenvalue and α-eigenvalue methods do is modify the final population so that it remains constant with time and the derivative of the first term of equation 2.2 is equal to zero. The two methods differ in the way they perform population control.

The code *Serpent*, used in this thesis, solves criticality calculations by applying the k-eigenvalue method and not the α-eigenvalue, so the latter is not treated in this thesis. Instead, the focus is placed in the former method. Nevertheless, it is worth saying that other Monte Carlo codes use the α-eigenvalue method, for instance MCNP [25] or TORTE [26], and that is is actually more accurate than the k-eigenvalue method for non-critical systems and equivalent to the dynamic solution of the neutron transport equation, as is shown in [24].

The procedure followed by the k-eigenvalue method to eliminate the time dependency of the neutron transport equation is to add a parameter *K* that modifies the fission reaction rate of the system so that it becomes critical. This means that the fission cross section is either increased or decreased in a way that the generation and removal of neutrons of the system become equal. With this modification, the population of neutrons becomes constant, so the flux no longer depends on time. Therefore,

$$\frac{1}{v}\frac{\partial \Psi}{\partial t} = 0$$

And equation (2.2) becomes:

$$\Omega \cdot \nabla \Psi + \Sigma_t \cdot \Psi = \iint \left[\frac{\chi \nu}{K} \Sigma_f + \Sigma_s \right] \Psi dE' d\Omega' \tag{2.3}$$

Equation 2.3 can also be written in the form of an eigenvalue problem, which is equivalent and very common in the literature that deals with criticality simulations in Monte Carlo [22] [27]:

$$M\Phi = \lambda F \Phi \tag{2.4}$$

In equation 2.4, the migration and loss of neutrons, described by the term $M\Phi$ is equalised to the modified fission source, described by the term $\lambda F\Phi$. Note that Φ represents the time-independent flux, to differentiate it from the time-dependent flux represented by Ψ. λ is the eigenvalue of problem, defined as:

$$\lambda = \frac{1}{K} \tag{2.5}$$

As it can be seen in equation (2.3), the k-eigenvalue method eliminates the time-dependent term from the transport equation by introducing a factor K to the fission neutron production term, modifying it so that the population of neutrons remains constant at all times and the problem becomes time-independent.

It is important to keep in mind that the factor K that divides the fission reaction rate is not equal to the actual effective multiplication factor (k_{eff}) of the original time-dependent system. Instead, it should be understood as the factor by which the production of neutrons should be reduced (or increased) in order to have a critical system. In other words, the static problem that the k-eigenvalue method solves is not the original problem, but a different one in which the original system is modified to become critical. There are several examples of studies made to assess to which extent the method is appropriate to solve criticality calculations ([24]). The conclusion they reach is that the method produces precise results when the system studied is close to criticality ($k_{eff} = 1$), but as the system is brought farther away from criticality and becomes either super- or subcritical the factor K can differ greatly from the actual k_{eff}.

Therefore, the k-eigenvalue method is only precise for critical systems (or systems close to criticality), while it badly approximates the multiplication factor if the system is not critical. For this reason obtaining the reactivity of the system during a transient scenario from a k-eigenvalue calculation can result in unreliable results, especially if the system is perturbed in a way that it becomes highly super- or subcritical, so it is important to be careful with the results obtained with this method.

2.2.2 Static approach using deterministic methods

Deterministic methods rely on finding an analytical solution to the neutron transport equation. To find the numerical solution, the spatial, angular and energy variables of the equation have to be discretised. To discretise the energy variable, most codes use the multi-group formalism [28], which is based on establishing an energy structure formed by a discrete number of energy groups. The flux is defined as an integral quantity within each group, and the cross sections are averaged over each group. The amount of groups that form the structure and the energy intervals they cover have a significant effect in the solution of the problem, some being more appropriate for thermal spectra, while others are designed for fast spectra. The accuracy of the result depends on the number of groups used, and the energy range of each group, which must be carefully chosen [29].

For the discretisation of the angular and space variables, a possibility is to use diffusion theory, which considers that neutrons diffuse through media following the Fick's law [30] [31]. This approach ignores the angular dependence of the neutron flux, which makes it appropriate for

calculations of large geometries, such as full-core or lattice calculations [32], especially if they are homogenised, whereas it is not recommended for calculations at the cell level, since the removal of the angular dependency yields imprecise results. Another possible approach is the use of the discrete ordinates method, or S_n method [33] [34]. This method is based on the discretisation of the domain in a finite set of angles, in which the neutrons are transported. For the spatial discretisation, a finite element method approach is followed, dividing the domain into a finite number of elements to calculate the transport of neutrons.

In this thesis, the code *Dragon* has been used to solve the neutron transport problem [35]. *Dragon* is a lattice code that solves the neutron transport equation following a modular approach. It contains several models that can simulate the neutronic behaviour of a nuclear reactor in different scales, from a unit cell to a fuel assembly in 1-D, 2-D or 3-D geometries. To generate multi-group cross sections, it interpolates microscopic cross sections that are obtained from standard libraries, which are later modified via the use of resonance self-shielding models to adapt them to the problem being studied and used to perform neutron flux calculations using several techniques.

2.2.3 Dynamic approach

Apart from the possibility of performing k-eigenvalue criticality calculations, *Serpent2* has recently included modules that make it capable of performing time-dependent transient simulations [6]. For a Monte Carlo code to solve a time-dependent problem, a complex set of phenomena has to be considered. Such problems are governed by three equations, one of them being the neutron transport equation, which has already been discussed in previous sections. In the case of dynamic calculations, though, the neutron transport equation differentiates between prompt and delayed neutrons. In static calculations the delayed neutrons are not considered, as they only appear as a consequence of the decay of precursors, something that is exclusive to time-dependent situations [36]. The system of equations is the following:

$$\frac{1}{v}\frac{\partial \Psi}{\partial t} + (\Omega \cdot \nabla + \Sigma_t)\,\Psi = \iint \Sigma_s \Psi d\Omega' dE' + \chi(E) \iint \nu(1-\beta)\Sigma_f \Psi d\Omega dE' + S_d + S \tag{2.6}$$

$$\frac{\partial C_i}{\partial t} = -\lambda_i C_i + \iint \beta \nu \Sigma_f \Psi dE' d\Omega' \tag{2.7}$$

and

$$S_d = \frac{1}{4\pi} \sum_i \lambda_i C_i \chi_i \tag{2.8}$$

Where β is the fraction of delayed neutrons present in the system, S_d is the source of delayed neutrons, which is formed by the decay of precursors, and C_i is the concentration of precursors in group i. Since in a system there are several nuclides that are precursors of delayed neutrons, they are usually discretised in a small number of groups of precursors, each of them with a representative decay constant, to simplify the problem.

The approach followed by *Serpent* to calculate transients is comprised by two steps. The first step consists in performing a criticality (k-eigenvalue) calculation of the system in order to obtain a distribution of the live neutrons present in the beginning of the transient, as well as the precursors that will generate delayed neutrons during the transient. The neutrons and precursors are sampled in a way so that they are at different points of their lifetimes, ensuring that they are correctly distributed over space and time. To obtain the initial source correctly, it is important that the system is critical, since the sources generated are a result of a static calculation, and so they can be seen as a "snapshot" of the system at a random time. Therefore, we are interested in simulating a system that is time-independent (critical), otherwise the sources obtained in different runs would vary, biasing the transient simulation.

The initial precursor source can be stored using two different methods: mesh-wise or point-wise. In the mesh-wise approach the volume of the system is divided into a number of cells defined by the user, and the precursor density is averaged over the volume in each cell. The size of the cells must be chosen carefully in order to preserve spatial fidelity while minimizing the memory usage of the stored source as much as possible. On the other hand, the point-wise approach stores the position (x, y, z) of each of the precursors that are present in the system. In this case, spatial fidelity is defined by the number of precursors present in the system. In both cases increasing the number of histories simulated in the source generation run results in a higher spatial fidelity of the source obtained because the number of precursors obtained from the simulation is higher, which improves the statistics. The drawback of increasing the number of neutrons is that the memory usage will increase, so the size much be chosen carefully to maximize spatial fidelity while not surpassing the memory limit that is imposed by the computer in which the simulations are run.

Once the source has been generated, the second step is to perform the dynamic simulation to study the evolution of the system with time starting from steady state conditions. The first thing to note about the dynamic simulation is that, unlike static calculations discussed in section 2.2.1, it is no longer a criticality calculation but a fixed-source calculation, because in this case there is a known distribution of neutrons and precursors at the beginning of the run. This change in the type of problem is reflected in the input by changing the card *set pop*, which is used for criticality calculations, for *set nps*. This card requires the user to specify the total number of particles to be simulated and the number of batches they will be divided into. For a transient simulation the card also requires a time binning, i.e. the time structure (amount of time to be simulated and time bins in which this will be divided) to be used. This time binning is defined with the

card *tme*. Time binning must also be provided for the detectors one wants to use to obtain tallies from the simulation, although the binning used for the tallies does not have to be the same as the one used in the simulation.

Time-dependent calculations have a number of issues that need to be addressed: they are the modelling of the delayed neutron population, the treatment of the branching neutron chains that are created from secondary neutrons, the explicit treatment of the time-dependency of the problem, the selection of the initial conditions and the application of time-dependent changes into the system (e.g., movement of control rods with time, increase or decrease of neutron absorber concentrations, etc.).

The approach that *Serpent* uses to treat the time dependency of the problem is to divide the transient into discrete time steps. At the beginning of every time step of the simulation a source is produced. It is based on the population of live neutrons that survived until the end of the previous time step (or the initial source distribution in the case of the beginning of the transient), and the delayed neutrons that should be emitted during the time step by the precursors that survived until the end of the previous time step (or the initial precursor distribution in the case of the beginning of the transient). From these neutrons, a number of particles is sampled based on the input created by the user. This number is limited by the computational power available and the model used, as a larger number of particles require more allocated memory to run the transient. The sampled particles are then simulated until they are either absorbed, leak out of the geometry, or reach the end of the time-interval. If secondary particles are created during the lifetime of the primary particles, they are stored and are also simulated eventually. Once all the histories have been simulated, those particles and precursors that survived to the end of the time interval form the initial source of the following interval.

A major problem that transient Monte Carlo simulations have to solve is the time scale difference between prompt and delayed neutrons [37]. Delayed neutrons are generated from the decay of a precursor that is generated as a result of a fission event, so the delayed neutron is not born until the precursor that has to generate it has decayed. In contrast, prompt neutrons are born directly after a fission event, so their lifetimes begin immediately. The time period between the moment a neutron is born and the moment it disappears from the system, either because it is absorbed or it leaks out, is of the order of 10^{-4} s, while the decay time of a precursor can reach up to 10^{2} s, depending on the characteristics of the nuclide that generates the delayed neutron. This great difference in the order of time poses a problem for the simulation. For instance, suppose an initial distribution of prompt neutrons and delayed neutron precursors from which a transient simulation is started. The prompt neutrons will begin their histories immediately, undergoing interactions with matter and generating secondary prompt neutrons, as well as precursors that will eventually decay and form delayed neutrons. The secondary prompt neutrons will, in turn, have their own interactions and will generate further neutrons and precursors in what is known as a prompt neutron chain. The time length of the chain can vary due to the stochastic nature of

the process simulated, but studies show that, on average, it is of the order of 10^{-2} s. After this time, there will be no more neutrons in the system, due to the fact that the prompt neutrons have already disappeared and the delayed neutrons have not been born since the precursors have not decayed yet. Once the precursors decay, the delayed neutrons generated start their respective chains. Therefore, the transient is divided into periods of time in which prompt neutrons are present in the system, separated by periods in which no histories can be simulated since the precursors have not decayed yet, so no power is generated and the tallies cannot be scored. Instead of this situation, it would be much better to have the delayed neutrons produced more often.

The way in which the precursors are treated in *Serpent* to solve the problem of the different time scales between prompt and delayed neutrons is to force the precursors to decay at the end of each time step in order to maintain a population of delayed neutrons present in the system at all times [38]. During the time interval the precursors are allowed to decay following the exponential decay law, generating delayed neutrons. At the end of the interval, the precursors that have not yet decayed are forced to emit a delayed neutron and are stored in the initial precursor source of the following interval. To maintain the physical meaning of the result the precursor has a certain statistical weight before it decays, which is divided between the delayed neutron and the post-decay precursor. Since it is possible that in some cases the statistical weight of either the generated delayed neutron or the precursor is very small, a round of Russian roulette is played to decide if the particles are killed or continue in the simulation. In this way the code achieves the purpose of always having a population of delayed neutrons in the system.

Finally, another aspect of the transient simulations that needs to be discussed is how to treat the branches created by secondary particles. A branch can be understood as all the events that happen during the lifetime of a neutron. When a primary neutron (i.e., a neutron present in the initial source) undergoes certain interactions, such as fissions, secondary neutrons are created. These secondary neutrons can then have their own interactions start their own branches and create tertiary neutrons, and so on. In a multiplying medium it is possible that the amount of secondary branches becomes too large, increasing the simulation time or even overflowing the buffers allocated to store particles. One possible approach is to simulate the system in a fully analog way, without applying any restrictions to the creation of secondary particles and simulating the totality of physical processes that take place in the system. This is possible to do in systems in which the rate of creation of secondary particles is low, or when a powerful computer is being used that can cope with a very large number of calculations. Nevertheless, in a Monte Carlo simulation it is also possible to use variance reduction techniques with the aim of speeding up the calculations by achieving a certain precision with a lower number of histories [39]. One of such methods which is used in this thesis is the branchless collision method, which is based on treating the fission interactions in an implicit way, so that in the event of a fission only one neutron is created, instead of the multiple secondary neutrons that would be created in an actual fission reaction. The born neutron has a modified statistical weight to account for

the neutrons that are not created, maintaining the physical meaning of the final solution. In this way the amount of secondary branches generated in a simulation is controlled, decreasing the time required by the computer to make the calculations.

2.3 Perturbation theory

In a non-critical reactor the reactivity evolves as a function of time [27], as in:

$$\rho(t) = \frac{1}{F(t)} \left(\Phi_0^*, [\Delta F - \Delta M] \, \Psi(t) \right) \tag{2.9}$$

Where F(t) is the fission operator, Φ_0^* is the adjoint flux of the system at time t=0, ΔF and ΔM are the differences of the operators F and M between time t=0 and t=t, and $\Psi(t)$ is the shape function of the flux of the system at time t.

While this expression yields the exact time evolution of the reactivity of the system, calculating it is not an easy task since it is necessary to know exactly the shape of the flux at every instant. The evolution of the flux shape depends on many factors, such as the geometry, the materials present in the system or the perturbation that is introduced in the system, so computing it at all times is difficult and computationally expensive, so the expression that appears in equation (2.9) is not suitable for a quick assessment of the reactivity in a transient.

Instead, perturbation theory allows us to derive an expression of the so-called *static reactivity*, with which it is not required to know the evolution of the flux over time to estimate the change in reactivity induced by a perturbation.

2.3.1 First order perturbation theory

To derive the expression of the static reactivity, it is useful to clearly define the two states of the system being studied: the first is the *unperturbed* state, which is the initial state of the system (before the introduction of the perturbation), while the second is the *perturbed* state, which is the state of the system once the perturbation has been introduced. Generally, before the perturbation is introduced the system is in a critical state, so its parameters are stable and time independent, and can be calculated easily (e.g., with a Monte Carlo eigenvalue simulation). On the other hand, when the system has been perturbed it is brought away from criticality, and therefore its parameters become time dependent, so determining them precisely is not an easy task. For this reason, it is desirable to find an expression that allows to calculate $\Delta\rho$ with just knowing the information from the unperturbed system.

The static neutron balance equations of the unperturbed and perturbed systems can be written as in equations 2.10 and 2.11, respectively:

$$M_0\Phi_0 = \lambda_0 F_0 \Phi_0 \tag{2.10}$$

$$M\Phi = \lambda F \Phi \tag{2.11}$$

Each of the parameters in equation 2.11 can be decomposed in an unperturbed term and an increment term. For example, the perturbed flux can be written as a function of the unperturbed flux as:

$$\Phi = \Delta\Phi + \Phi_0 \tag{2.12}$$

Applying the decomposition of the flux of equation 2.12 into equation 2.11 we obtain:

$$M\Phi_0 = \lambda F \Phi_0 - (M - \lambda F)\Delta\Phi \tag{2.13}$$

Expanding the first term of the right hand side into a zero- and first order terms, and introducing it into equation 2.13, we get:

$$M\Phi_0 = \lambda F_0 \Phi_0 + \lambda_0 \Delta F \Phi_0 - (M - \lambda F)\Delta\Phi \tag{2.14}$$

The quantity that we ultimately want to determine, $\Delta\lambda$, is a number, so the dependency on energy and space of the flux needs to be removed through integration. To avoid the loss of generality that this integration brings to equations 2.10 and 2.14, for the unperturbed and perturbed systems, respectively, they are multiplied by a weighting function before being integrated. The integration of the product between the weighting function and the flux results in the scalar product of the two functions [1]. Equations 2.10 and 2.14 then become:

$$(\Phi^w, M_0\Phi_0) = \lambda(\Phi^w, F_0\Phi_0) \tag{2.15}$$

$$(\Phi^w, M\Phi_0) = \lambda(\Phi^w, F_0\Phi_0) + \lambda_0(\Phi^w, \Delta F\Phi_0) - (\Phi^w, [M - \lambda F]\Delta\Phi) \tag{2.16}$$

Subtracting equations 2.16 and 2.15 yields:

[1] The notation (A, B) refers to the scalar product of operators A and B

$$(\Phi^w, [\Delta M - \lambda_0 \Delta F]\Phi_0) = \Delta\lambda(\Phi^w, F_0\Phi_0) - (\Phi^w, [M - \lambda F]\Delta\Phi) \tag{2.17}$$

The next step is to eliminate the term containing the change in the flux shape, $\Delta\Phi$, from equation 2.17. To do so, we expand the term in first- second- and third order terms and introduce them into the equation after neglecting the second- and third order terms, so that we obtain:

$$(\Phi^w, [\Delta M - \lambda_0 \Delta F]\Phi_0) = \Delta\lambda(\Phi^w, F_0\Phi_0) - (\Phi^w, [M_0 - \lambda_0 F_0]\Delta\Phi) \tag{2.18}$$

Finally, $\Delta\Phi$ can be removed from equation 2.18 by choosing the unperturbed adjoint flux (Φ_0^*) as the weighting function. The reason is that by choosing Φ_0^* the term becomes equal to zero:

$$(\Phi_0^*, [M_0 - \lambda_0 F_0]\Delta\Phi) = (\Delta\Phi, [M_0^* - \lambda_0 F_0^*]\Phi_0^*) = 0 \tag{2.19}$$

Since Φ_0^* is the solution of the adjoint eigenvalue problem (equation 2.20), the term containing $\Delta\Phi$ is removed.

$$M_0^*\Phi_0^* = \lambda_0 F_0^*\Phi_0^* \tag{2.20}$$

After removing the term containing $\Delta\Phi$ and solving equation 2.18 for $\Delta\lambda$, the expression of the first order perturbation formula is finally obtained:

$$\Delta\rho = \frac{(\Phi_0^*, [\lambda_0 \Delta F - \Delta M]\Phi_0)}{(\Phi_0^*, F_0\Phi_0)} \tag{2.21}$$

The expression obtained in equation 2.21 is very convenient, since it allows to approximate a change in reactivity by knowing only the unperturbed state, which is easy since the system is critical, i.e., time-independent, and the perturbation introduced to know the change in the fission and migration operators. Nevertheless, the fact that it uses the flux shape of the unperturbed system limits its effectiveness. For instance, if the flux shape varies significantly after the perturbation, the information corresponding to this change is lost when using the 1st order perturbation formula, so the approximation becomes poor.

In conclusion, the 1st order perturbation formula is a convenient approximation of the dynamic reactivity of the system since the parameters it contains are easily obtainable from the unperturbed system and the perturbation induced. Nevertheless, it is only an effective approximation if the flux shape does not vary significantly after the perturbation. Therefore, it is possible that in certain situations, such as local perturbations, the approximation might behave poorly.

2.3.2 Exact perturbation theory

An alternative to the 1st order perturbation method that can potentially be an improvement to the solution provided by equation 2.21 is the so-called *exact perturbation theory*. The difference between the two approaches is that the exact perturbation equation takes into account the solution of the perturbed problem, contrary to what the first order perturbation equation does. Although obtaining the parameters to apply this approach is more difficult since the solution of the perturbed state of the system is necessary, it involves some advantages compared to the first order solution. Since it takes into consideration the perturbed system, the information corresponding to the change in the flux shape is not lost when calculating $\Delta\rho$, so the solution obtained for a local perturbation is more precise than the one coming from a first order approximation.

The derivation of the exact perturbation formula begins with the eigenvalue problem equation of the perturbed system and the adjoint of the unperturbed system:

$$M\Phi = \lambda F\Phi \tag{2.22}$$

and

$$M_0^*\Phi_0^* = \lambda_0 F_0^*\Phi_0^* \tag{2.23}$$

In order to be able to obtain an expression with ΔM and ΔF, it is useful to multiply equations 2.22 and 2.23 by Φ_0^* and Φ, repectively, which yields equations 2.24 and 2.25. Changing the order of the elements in the scalar products obtained in equation 2.25 results in equation 2.26:

$$(\Phi_0^*, M\Phi) = \lambda(\Phi_0^*, F\Phi) \tag{2.24}$$

$$(\Phi, M_0^*\Phi_0^*) = \lambda_0(\Phi, F_0^*\Phi_0^*) \tag{2.25}$$

$$(\Phi_0^*, M_0\Phi) = \lambda_0(\Phi_0^*, F_0\Phi) \tag{2.26}$$

Since $\Delta F = F - F_0$, equation 2.26 can also be written as:

$$(\Phi_0^*, M_0\Phi) = \lambda_0(\Phi_0^*, F\Phi) - \lambda_0(\Phi_0^*, \Delta F\Phi) \tag{2.27}$$

Finally, subtracting equations 2.24 and 2.27 and solving the resulting expression for $\Delta\lambda$ yields

the exact perturbation formula:

$$\Delta\rho = \frac{(\Phi_0^*, [\lambda_0 \Delta F - \Delta M]\Phi)}{(\Phi_0^*, F\Phi)} \tag{2.28}$$

Note that the expression found in equation 2.28 is very similar to the first order perturbation formula (equation 2.21. The only difference between the two is that the exact perturbation uses Φ and F, while first order perturbation uses Φ_0 and F_0. By using the parameters of the perturbed system, the solution found with exact perturbation is more precise than the one found with first order perturbation. Note, however, that referring to this approximation with the name *exact* can be misleading. The dynamic reactivity can only be exactly approximated with this method if the perturbed flux used in equation 2.28 is the actual time-dependent flux of the system. Since this cannot be known because how difficult it is difficult to calculate, a flux that comes from a static criticality calculation of the perturbed system is generally used. The static reactivity obtained is therefore only an approximation, although in this thesis it will still be referred to as the *exact* perturbation approach to comply with the literature [27].

At a first glance it can seem that the solution of equation 2.28 does not improve the change of reactivity $\Delta\rho = \lambda - \lambda_0$ that can be found just by solving the criticality problems of the unperturbed and the perturbed systems, for example using a k-eigenvalue Monte Carlo simulation. Nevertheless, accurately finding the solution of a non-critical system using this method is not always possible, especially when the degree of sub- or supercriticality increases. Therefore, the $\Delta\rho$ calculated as the difference of the eigenvalues of the two systems is not necessarily accurate, and the perturbation theory approach might yield better approximations.

2.4 Point kinetics equations

The point kinetics equations are a reduced order representation of the reactor that define its kinetic behaviour, i.e., the time-dependent evolution of the neutron population and the precursor concentration. They can be obtained from the exact kinetics equations [27], which are:

$$\frac{dN(t)}{dt} = \frac{\rho(t) - \beta(t)}{\Lambda(t)} N(t) + \sum_k \lambda_k C_k(t) \tag{2.29}$$

and

$$\frac{dC_k(t)}{dt} = -\lambda_k C_k(t) + \frac{1}{\Lambda(t)} \beta_k(t) N(t) \tag{2.30}$$

Where N(t) is the neutron population, $\rho(t)$ is the reactivity of the system, $\beta_{eff}(t)$ is the effective delayed neutron fraction, calculated as the sum of the delayed neutron fractions of each of the

precursor groups, β_k, $\Lambda(t)$ is the mean neutron generation time, λ_k is the decay constant of the precursors in group k and C_k is the precursor concentration in group k.

The time-dependent parameters are defined as:

$$\Lambda(t) = \frac{\left(\Phi_0^*, \frac{1}{v}\Psi\right)}{\left(\Phi_0^*, F\Psi\right)} \tag{2.31}$$

$$\rho(t) = \frac{\left(\Phi_0^*, [F-M]\Psi\right)}{\left(\Phi_0^*, F\Psi\right)} \tag{2.32}$$

$$\beta(t) = \sum_k \beta_k(t) = \sum_k \frac{\left(\Phi_0^*, F_{dk}\Psi\right)}{\left(\Phi_0^*, F\Psi\right)} \tag{2.33}$$

The integral kinetic parameters $\rho(t)$, $\beta_k(t)$ and $\Lambda(t)$ depend on the time-dependent shape function of the flux, $\Psi(r, E, t)$. If this could be easily obtained, then the parameters $\rho(t)$, $\beta_k(t)$ and $\Lambda(t)$ could be calculated and the exact kinetics equations could be solved to determine N(t) and $C_k(t)$. Nevertheless, $\Psi(r, E, t)$ can only be obtained by solving a complex time-, energy- and position-dependent problem, so usually a simpler approximation is preferred, the point kinetics equations.

From equations 2.29 and 2.30, the point kinetics equations can be obtained by applying three simplifications. The biggest simplification if to eliminate the time dependence of the flux shape, so the initial flux shape is used to calculate the kinetics parameters since:

$$\Psi(r, E, t) \approx \phi_0(r, E) \tag{2.34}$$

The two remaining simplifications consist in replacing the operators F and F_d by the initial operators of the critical system, F_0 and F_{k0}

These simplifications have the consequence that the parameters $\beta_{eff}(t)$ and $\Lambda(t)$ become time-independent, equal to the values corresponding to the initial critical system, $\beta_{eff,0}$ and Λ_0. For simplicity, though, the subindex 0 is removed from the parameters, so the point kinetics equations finally become:

$$\frac{dN(t)}{dt} = \frac{\rho(t) - \beta}{\Lambda} N(t) + \sum_k \lambda_k C_k(t) \tag{2.35}$$

and

$$\frac{dC_k(t)}{dt} = -\lambda_k C_k(t) + \frac{1}{\Lambda}\beta_k N(t) \tag{2.36}$$

Serpent uses eight precursor groups by default, which means that the point kinetics equations form a system of nine equations with nine unknowns, the neutron population N(t) and the concentrations of the eight precursor groups, $C_k(t)$.

The parameters that become time-independent can be readily calculated with various methods, such as Monte Carlo techniques [40]. The only kinetic parameter that remains being time-dependent is the reactivity of the system. It can be left as a function of time, obtaining it from a dynamic Monte Carlo simulation, for instance, or the dependency can be removed by calculating an approximation of its value at a given point in time, known as *static reactivity*. There are several methods to make such approximation, one of them being the perturbation theory approach, discussed in section 2.3.

3 Models used

The models used in this thesis are detailed in this section. Firstly, an overview of the most relevant kinetic parameters calculated with *Serpent* static simulations with the critical configuration of each of the four models is made. With it, an initial assessment of the expected behaviour of the models is performed, which will later be confirmed with the results of the dynamic Monte Carlo simulations.

After the kinetic parameters are reviewed, a more detailed description of each of the models is made, including the description of the materials, the geometry and how it is implemented in the different codes used for the study, together with the general approach followed for its implementation and the calculation of the results.

3.1 Kinetic parameters of the models

The final objective of determining the static reactivity is to use it as an input of the point kinetics equation in order to calculate the evolution of the neutron population. Apart from the static reactivity, other parameters need to be known to be able to solve the point kinetics equations. They are the effective delayed neutron fraction (β_{eff}) and the mean neutron generation time (Λ). Although these parameters are time-dependent, as they are affected by the temperature feedbacks of the system in a real situation, for an open loop system, such as the ones that are presented in this thesis, they can be considered essentially constant as feedback effects are neglected (see equation 2.35. Two other kinetic parameters of importance are the prompt neutron lifetime (l_p) and the effective decay constants of the delayed neutron precursors (λ_{eff}) [41]. Together, these parameters can give a very clear idea of the dynamic behaviour of a system, as they provide information such as the magnitude of a perturbation required to make it prompt critical, or the space of time between consecutive neutron generations, which governs how fast the dynamic evolution of the population is and has safety implications for a reactor since a very sudden evolution makes a reactor difficult to control.

Serpent has the advantage that it provides these parameters [42], and since they are considered to be time independent because the systems presented previously are in open loop, they can be easily obtained from static k-eigenvalue calculations. Table 1 presents the values obtained in the three models that are object of study in this thesis.

Table 1: *Values of the main kinetic parameters of the studied models.*

	β_{eff} **(pcm)**	Λ **(μs)**	l_p **(s)**	λ_{eff} **(s^{-1})**
Thermal slab	814 ± 1	55.84 ± 0.0399	55.77 ± 0.0394	0.5191 ± 0.0009
Fast slab	525 ± 1	0.2870 ± 0.0001	0.2864 ± 0.0001	0.6471 ± 0.0013
Heterogeneous MYRRHA	381 ± 1	0.9257 ± 0.0013	0.9245 ± 0.0011	0.5387 ± 0.0011

The effective delayed neutron fraction (β_{eff}) of the system is dependant on the composition of the fuel and has an important role in reactor kinetics, as delayed neutrons have a dominant role in the fission reaction chain control [43]. The delayed neutrons that are present in a system are generated from the precursors, and these are created as a result of fission reactions. Therefore, the fissile nuclides that are fissioned determine which are the precursor nuclides, which have a certain decay time. If the precursors have a longer decay time, then the delayed neutrons take longer to appear in the system and cause fissions of their own, so the dynamic evolution of the system is slower. β_{eff} is also dependent on the amount of prompt neutrons that are generated in fissions, as it is defined as the fraction of delayed neutrons over the total population of neutrons (delayed plus prompt) in the system. The average amount of prompt and delayed neutrons that are generated in a fission event is determined by the total neutron yield (ν_t) and the delayed neutron yield (ν_d), which depend on the nuclide [44]. The fraction of delayed neutrons determines which is the magnitude of the insertion of reactivity that makes a reactor prompt critical. If a perturbation introduces a reactivity equal to β_{eff}, the amount of prompt neutrons generated in fissions becomes enough to maintain the fission chain by itself, so the delayed neutrons are no longer needed and the dynamic evolution of the system becomes much faster, as the time between consecutive neutron generations is considerably decreased, which makes the reactor very difficult to control. Therefore, β_{eff} is a very important parameter for the control, design and operation of a reactor, as it provides information of the maximum reactivity insertions that can be introduced in the reactor that do not make it prompt critical.

The thermal slab model is formed by UOX fuel and light water as a moderator, which causes the majority of fissions being induced by neutrons in the thermal energy group. As the fissions in the fuel are mainly caused by ^{235}U, the β_{eff} in the thermal slab model is governed by the data of this nuclide. Looking at the neutron data provided by [44], ^{235}U has a total neutron yield of 2.4355 and a delayed neutron yield of 0.0162. In comparison, the fast slab model has fissions in the fast spectrum causes mainly by ^{239}Pu, which has a higher fission cross section than ^{238}U. The total neutron yield of ^{239}Pu is of 2.8836, while the delayed neutron yield is of 0.0065. It can be seen, then, that uranium has a higher delayed neutron yield than plutonium, which makes the delayed neutron fraction in a system that has fissions caused by ^{235}U higher than the one of a system fuelled by ^{239}Pu. This agrees with what is seen in table 1, in which the β_{eff} of the thermal slab system is higher than that of the fast slab system. The consequence is that the thermal system is somewhat easier to control, as it can absorb larger amounts of reactivity in a given perturbation before it turns prompt critical. The β_{eff} of the MYRRHA model is more difficult to measure, because the composition of the fuel is much more complex than in the slab models. Nevertheless, as MYRRHA is a fast system that contains significant amounts of plutonium and other actinides in the fuel, such as americium, the delayed neutron fraction obtained is also lower than the one of the thermal slab model.

The mean neutron generation time (Λ) and the prompt neutron lifetime (l_p) measure similar parameters of the system. On the one hand, Λ measures the mean time between consecutive

neutron generations are created, i.e., the average amount of time for a neutron (prompt or delayed) to be born, thermalise or not, depending on the characteristics of the system, and have a fission reaction with a fissile nuclide of the fuel. On the other hand, the prompt neutron lifetime measures the duration of the "life" of a prompt neutron from the time when it is born to the time when it causes a fission. When a system is not prompt critical, the mean neutron generation time is governed not only by the action of the prompt neutrons, but also of the delayed neutrons, which have longer lifetimes as the time for the precursor to decay also need to be taken into consideration. Since Λ considers both prompt and delayed neutrons, it is larger than l_p, which only considers prompt neutrons. Comparing both parameters can provide information on the velocity of the dynamic evolution of a reactor, as well as the importance of the delayed neutrons in a given configuration. In table 1 it can be seen how in the thermal model Λ is two orders of magnitude larger than in the other models, which have a fast spectrum. The difference is caused by the amount of time that is required to thermalise the neutrons to the required energy to cause fissions. When there is no thermalisation, the neutron generations are created much faster, which means that the dynamic evolution of the reactor is quicker. The Λ of the fast slab and the MYRRHA model also shows that the geometry and the fuel configuration of the system also play a role in determining the time between generations, as it also depends on the length of the path the neutrons have to travel before they can interact with fissile nuclides, as well as the number of interactions they undergo before doing so.

Finally, the effective precursor decay constant (λ_{eff}) is the last of the kinetic parameters of interest of a nuclear system. As there are several precursor nuclides, each with its own decay constant, the most usual thing is to group the nuclides with a similar decay constant in a small number of groups. In *Serpent* the discretisation is made into eigth groups, each containing a fraction of the precursors. The fraction of precursors in each group depends on the model studied, so the effective decay constant depends on the characteristics of the system. If the effective decay constant is larger, it means that the short-lived precursors are more numerous than the long-lived ones, so the delayed neutrons appear and have their effect quicker, contributing to decrease the mean neutron generation time. This can be seen in table 1 by comparing the fast slab model with the heterogeneous MYRRHA model, since the fast slab has a higher λ_{eff}, which implies that the decay time time of the precursors is shorter, meaning that the delayed neutrons required to begin the next generation of neutrons appear more quickly than in the heterogeneous MYRRHA case, making the Λ of the fast slab model lower.

In conclusion, the kinetic parameters of the four models show that the fast slab model should be the one with the fastest dynamic evolution in terms of neutron population and flux, since it has the lowest mean neutron generation time. The thermal system should have the slowest evolution thanks to the need of the neutrons to thermalise before they cause fissions, which increases the value of Λ with respect to the models with a fast spectrum. As for the β_{eff} of the systems, the model with the biggest fraction of delayed neutrons is the thermal slab, which means that it is the model that can absorb the largest perturbations before turning prompt critical. On the

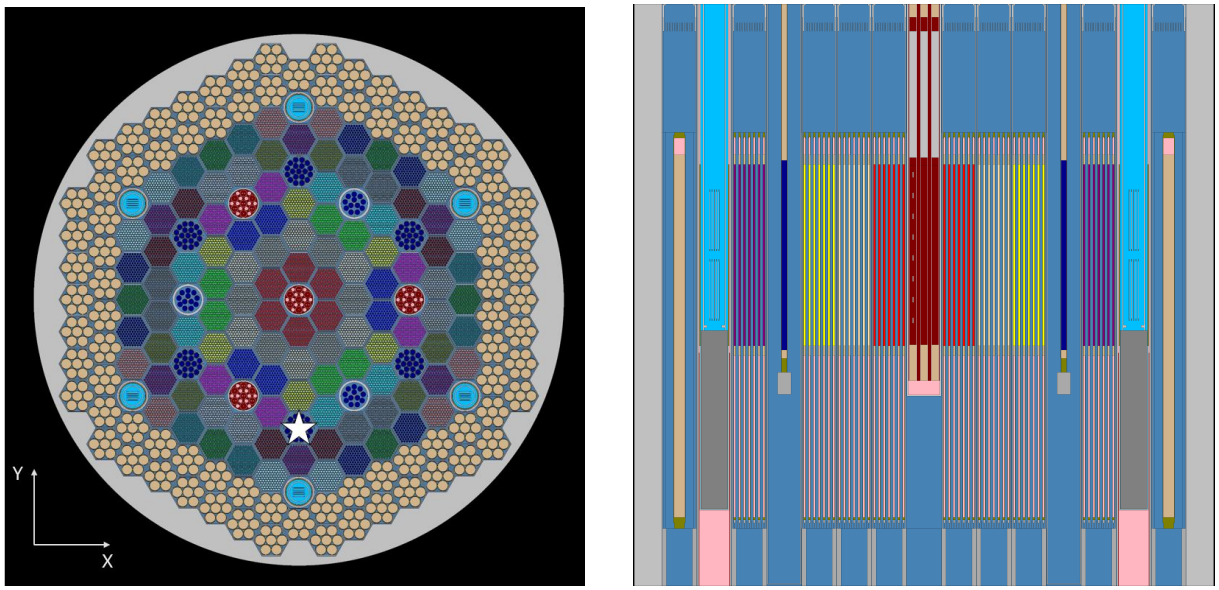
other hand, the MYRRHA model is the one where perturbations have the largest effects, since an insertion of 381 pcm of reactivity already turns it prompt critical.

3.2 Heterogeneous MYRRHA model

3.2.1 Geometry and materials

The heterogeneous model of the MYRRHA core used in this study corresponds to the version 1.6 of the design [45], which is currently under development. This is a slightly outdated version, the most recent being the 1.8 [46]. Although it is not the latest version of the design, access to version 1.6 is easier due to confidentiality issues with the most recent version, and the Serpent model was readily available. For these reasons it was decided to work with version 1.6 rather than with version 1.8. The model has already been implemented and studied in detail in the Monte Carlo codes MCNP [18], OpenMC [47] [48] and Serpent [49].

The model includes an accurate representation of the geometry of the core, including all the fuel assemblies (FA), the control rods (CR), reflector assemblies and in-pile systems (IPS) that will be used for experimental measurements and the production of radioisotopes, as well as the stainless steel jacket that surrounds the core. Figure 2 shows the radial (left) and axial (right) cross sections of the model.

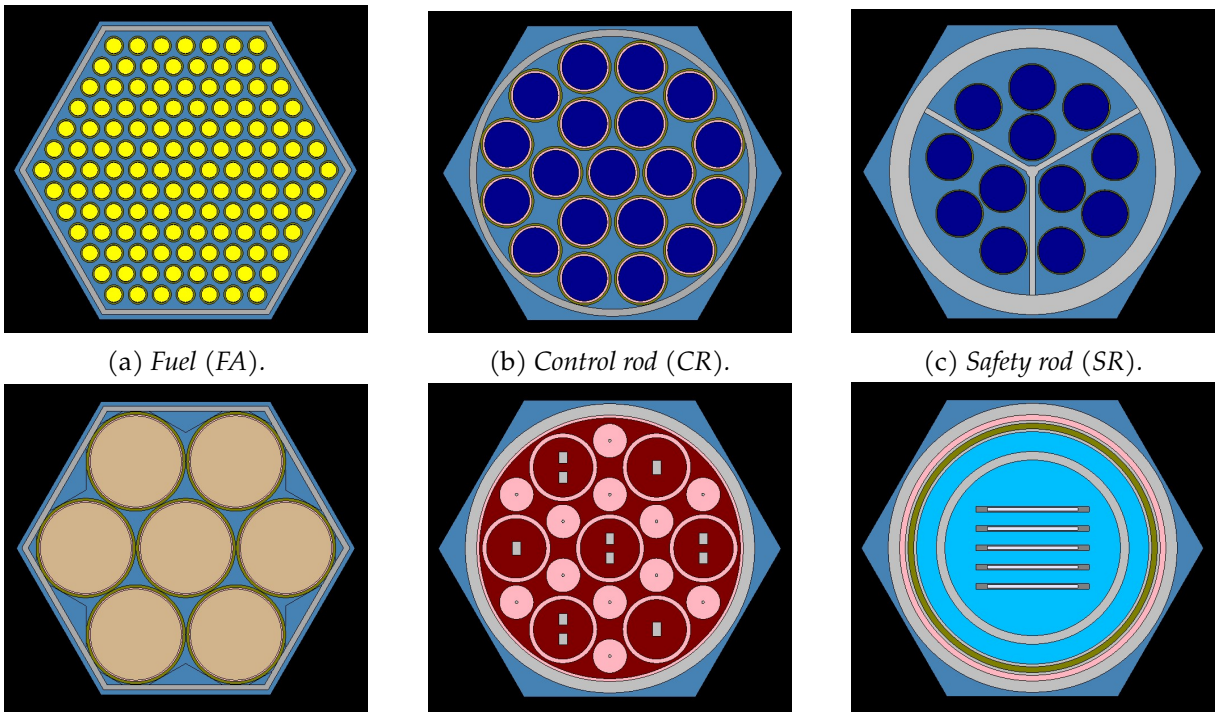


(a) *Radial cross section.* (b) *Axial cross section.*

Figure 2: *Heterogeneous model of the MYRRHA core.*

The different assemblies present in the core can be appreciated in figure 2, as well as the active region and the top and bottom reflector. Each individual assembly has a hexagonal geometry, and can be seen in more detail in figure 3. The fuel assemblies (FA) are formed by 127 individual fuel pins, surrounded by a gap of helium gas and the cladding, made of austenitic 15-15 Ti stainless steel. The fuel pins are surrounded by LBE coolant and by a stainless steel support

plane. The composition of the fuel varies depending on the position of the fuel assembly in the core, as can be seen in figure 3 (a), in which the several types of MOX fuel compositions are represented in different colors. For the control of reactivity, MYRRHA has two types of assembly. One of them is the control rod assembly (CR). The core has 6 of these assemblies, each of them is made of 19 B_4C absorber rods and is tasked with the control of small reactivity variations, and the other type is the safety rod assembly (SR). It is formed of 12 B_4C, and the 3 assemblies present in the core have the function of immediately bringing the reactor to a halt in case it is required. Surrounding the fuel and control assemblies there is a layer of reflector assemblies (RA), which have the function of minimising the loss of neutrons by leakage. These assemblies are made of beryllium, and there are a total of 84 in the core. Finally, MYRRHA also incorporates assemblies that are dedicated to experimentation with irradiated materials and production of radioisotopes for medical applications. These are the so-called in-pile systems (IPS), and two types can be found in the reactor, depending on the neutron flux that is present in the region: one of them has a fast neutron spectrum, while the other has a thermal spectrum. The 4 IPS assemblies with a fast neutron flux (FIPS) are located in the inner region of the core, while the 6 assemblies with a thermal flux (TIPS) are found in the outer region, next to the reflector assemblies. In contrast with the rest of the assemblies, that are cooled with LBE, the TIPS assemblies are filled with light water in order to thermalise the neutron flux that is present in the assembly.



(a) *Fuel* (*FA*). (b) *Control rod* (*CR*). (c) *Safety rod* (*SR*).

(d) *Reflector* (*RA*). (e) *Fast IPS* (*FIPS*). (f) *Thermal IPS* (*TIPS*).

Figure 3: *Assembly types present in the heterogeneous model.*

In total, the model includes 55 material definitions and 327 nuclides. Serpent loads the data

for each of the nuclides present in the model and stores it before starting the run. This makes Serpent faster in comparison to other Monte Carlo codes such as MCNP, which in contrast loads the nuclear data as it is required (i.e., when an interaction with a specific nuclide takes place during the simulation, the code loads the data for that nuclide, uses it to calculate the collision and then deletes the data), but the disadvantage is that it is more costly memory-wise. For the heterogeneous model of MYRRHA, the memory usage to load the model is shown in table 2, separated by the type of data.

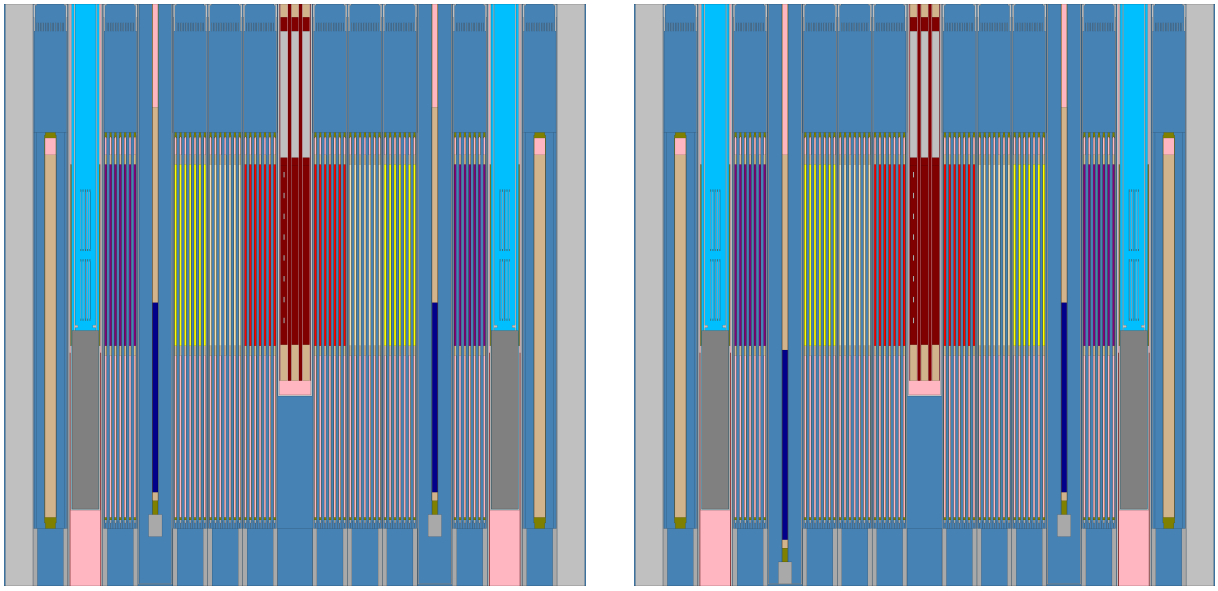
Table 2: *Memory usage of the heterogeneous MYRRHA model for a transient simulation.*

Type of data	**Memory (MB)**
Cross sections	63762.67
Materials	5093.26
Others	68814.87
Total allocated memory	**139208.97**
Total available memory	**192070.15**
Memory left for histories	**52861.18**

As can be seen in table 2, loading the heterogeneous model takes up a significant part of the available memory, as much as 72%. This is a serious limitation to the amount of histories that can be run, as the information of the neutrons and the precursors that are simulated, as well as the files with the calculated results, must be stored using the available memory after loading the model. The observed memory usage, together with several preliminary transient simulations have lead to the conclusion that the memory available is not enough for the calculations that need to be made, and that the time required for the calculations is too high for the purposes of the project, so it was decided that the MYRRHA model would not be used for the study of the static reactivity, and simpler models that allowed to perform the time-dependent simulations with better statistics would be used instead (see section 3.3). Nevertheless, the model has been used for the study of feasibility of the dynamic Monte Carlo method, as its complexity and memory consumption makes it appropriate for the study of the limitations of the method and the Newton cluster.

3.2.2 Studied cases

Two different transient cases have been studied for the heterogeneous MYRRHA model. The first case is the null transient, and the other is a local positive reactivity insertion consisting in the ejection of one control rod bundle. The parameters that have been tallied are the reactivity of the system, the neutron population, which has been condensed in one energy group, and the neutron flux distribution, which has been obtained in several ways: two different spectral flux



(a) *Critical configuration.* (b) *Positive reactivity insertion.*

Figure 4: *Core configurations of the transients studied in the heterogeneous MYRRHA model.*

distributions have been obtained, one for the whole system, comprising all the volume of the reactor, and another in the thermal IPS. This last distribution serves to illustrate the differences in the spectra present inside the core, highlighting its complexity. Also, three spatial flux distributions have been tallied, one in the axial direction along the active region of the core, and two in the radial direction, following the x and y axes as defined in figure 2.

The null transient is run keeping the model in its critical configuration, which has already been used to generate the initial source distribution that is required for running the transient simulations. Since the model is kept critical, it is expected that the tallied parameters remain time-independent, so they should be constant during all the duration of the transient, regardless of its length. It is the simplest possible transient, as the population should not vary so there is no chance of overflowing the buffers of the live neutrons and precursors. Therefore, this case is appropriate to perform a study of the limiting parameters in the input, such as the maximum possible histories per batch that can be run, the necessary space that needs to be allocated for buffers, or the time structures that can be used to simulate transients in the model.

The rod ejection transient starts from a critical configuration of the reactor and introduces a local perturbation in the form of a step positive reactivity insertion in the region of the control rod assembly that is removed from the core (see figure 4. Since a control rod assembly is removed, the reactivity of the system is expected to increase instantaneously to its perturbed value and remain constant for the duration of the transient. The population of neutrons should also increase exponentially from its initial value indefinitely, as the constant positive reactivity keeps the reactor in a supercritical state. The amount of time that can be simulated will be dependent on how fast the neutron population grows, as a bigger population increases the simulation time,

because more neutron histories need to be simulated, but most importantly, it also increases the necessary memory to store the information of the simulated histories, which is the most limiting factor that has been encountered for transient simulations.

3.3 Slab models

3.3.1 Characteristics of the Serpent model

The slab models defined in the thesis are intended to be used in the comparison between the time-dependent Monte Carlo results and the point kinetics approach. As the objective of the study is to simulate the time evolution of the reactivity in a system after a local perturbation has been introduced, the proposed model has to comply with various requirements: firstly, it needs to have a heterogeneous configuration, so that local perturbations can be introduced. Secondly, it needs to be simple enough, both geometrically and material-wise, to allow time-dependant Monte Carlo calculations to run with sufficiently low uncertainties and in an acceptable amount of time, in contrast to the more complex MYRRHA model. Thirdly, even though the study results will no longer be as realistic as they would be with the heterogeneous MYRRHA model, it is also important that they are representative of the phenomenon studied, so that in the future, when more powerful computers are available for other researchers, the methodology can be applied to the full heterogeneous model and be useful for the safety analysis of the reactor. With that in mind, it has been decided that the model used should be an infinite slab, with the appropriate materials defined to obtain two models, one with a thermal spectrum and the other with a fast spectrum. The spatial composition of the slab has been taken and adapted from [50], and is formed by nine layers of fuel, each with an axial length of 7 cm, separated by layers of moderator of 5 cm, as shown in figure 5. The dimensions of the model can also be seen.

The fact that the model is an infinite slab helps the simplification of the problem, since it allows to reduce the number of dimensions that need to be considered in the calculations. For *Serpent*, the model is 2-dimensional, infinite in the z direction and with reflective boundary conditions in the x and y directions. Although it is radially infinite since the boundary conditions used are reflective, figure 5 includes the radial dimensions of the model as they were defined in the input file. This simplification of the model helps to reduce the computational requirements, so the calculation time is decreased and therefore the number of histories can be increased in the Monte Carlo calculations.

Two different material compositions have been used for the study of the slab model in order to have examples of a thermal and a fast spectrum. To obtain the thermal spectrum, the fuel used is UOX, containing a mixture of U^{235}, U^{238} and O^{16}, while the coolant used is light water, which acts as a moderator at the same time. For the fast spectrum case, the fuel considered is MOX, formed by a mixture of U^{238}, Pu^{239} and O^{16}, while the coolant is LBE, the composition of which has been taken from the material definition of the heterogeneous MYRRHA core model to potentially extrapolate the results to the full 3-dimensional MYRRHA case. A more precise

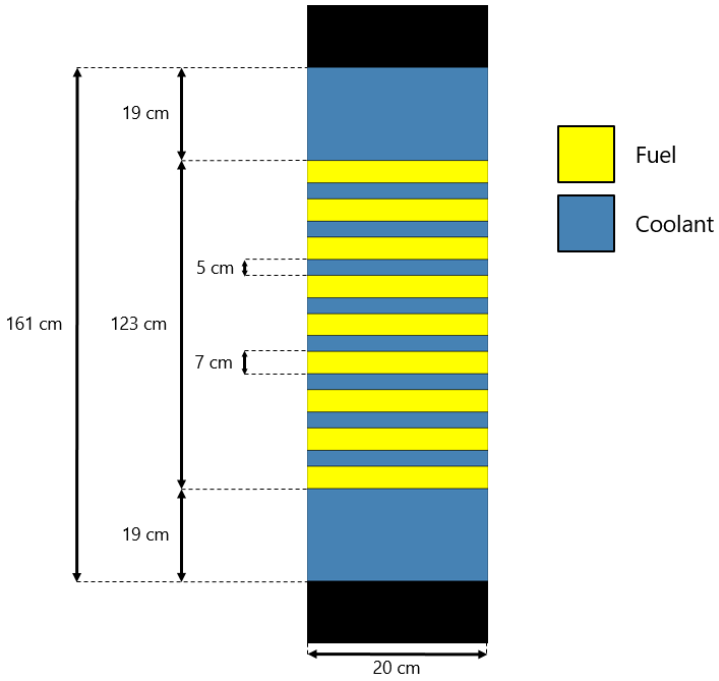


Figure 5: *Geometry and materials of the unperturbed slab configurations in Serpent.*

description of the materials used and their compositions can be found in section 3.3.3. Nevertheless, it is worth to point out that, similarly to what has been done with the geometry, the definition of the fuels have been made deliberately simplified by removing isotopes of uranium and plutonium that would be found in real fuel compositions in order to reduce the complexity of the model and maximize the memory available to run the Monte Carlo transient simulations with low uncertainties.

In order to introduce local perturbations to the system, the method used consists in varying the composition of the fuel in one of the fuel regions, the one in the bottom of the slab, as shown in figure 6, where the modified fuel region can be seen in green. This type of perturbation modifies the fission cross section of the system locally, inducing a perturbation through a change in the fission rate. In the case of the thermal spectrum slab the perturbation is introduced by varying the weight fraction of the U^{235} and U^{238}, and for the fast spectrum slab it is done in an analogous way with the weight fractions of the Pu^{239} and U^{238}. Again, more detailed compositions of the fuels used for each reactivity insertion studied can be found in section 3.3.3.

The memory usage of the slab models is much lower than the one observed in the heterogeneous MYRRHA model. In the case of the slab models (see table 3), it can be seen how the memory available for particle simulation is more than 70% of the total available memory, whereas in the MYRRHA model (see table 2) it is only of 28%. Therefore, in the slab models it is possible to perform calculations with a larger amount of particles, leading to more meaningful results, which is the reason why they have been selected as the models in which the static reactivity study is performed.

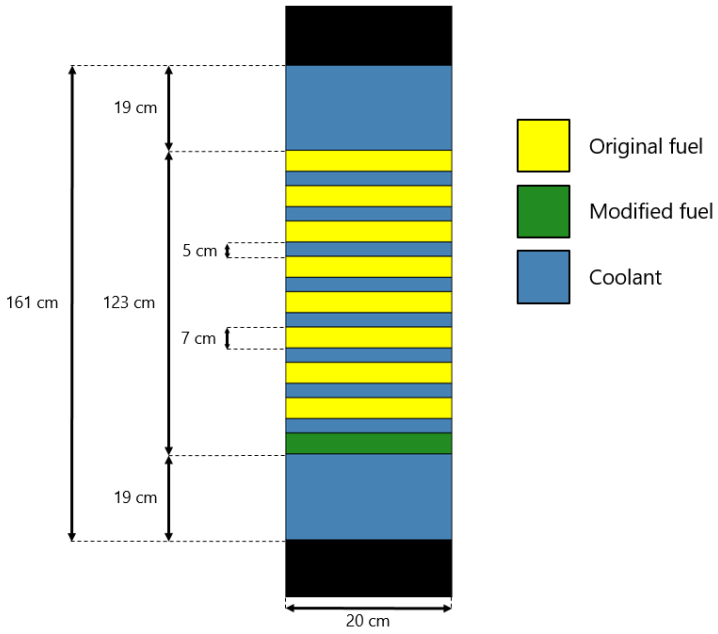


Figure 6: *Geometry and materials of the perturbed slab configurations in Serpent.*

In order to compare the differences in the behaviour between the thermal and fast spectra in time-dependent simulations the geometry and method to introduce perturbations in the model is the same for the two slabs. Moreover, the magnitude of the perturbations introduced is the same (in \$) in the two models. The perturbations are: -0.8\$, -0.5\$, 0.5\$, 1\$, 2\$ and 5\$. The reactivity changes in pcm are not the same in the two models for a given perturbation in \$ because the models include different material configurations and different spectra, so their effective delayed fractions are different. Nevertheless, the dollar approach is a good method to compare the behaviour of the fast and thermal models, because it represents how far a reactor is to become prompt critical for a given reactivity insertion (an insertion of 1\$ makes the reactor prompt critical), so it allows to make comparisons between two different reactors. Table 4 contains a comparison of the insertions in pcm required in each model to achieve each of the desired perturbations.

As can be seen, the fast spectrum model needs a smaller reactivity insertion (in pcm) to reach prompt criticality (1\$) in comparison with the thermal spectrum model. This is an expected behaviour that is observed in real thermal and fast reactors, and is related to the materials that form the fuel: the UOX fuel used in the thermal slab has a higher effective delayed neutron fraction (β_{eff}) than the MOX used in the fast slab because the former contains ^{235}U while the latter contains ^{239}Pu. As the fraction of prompt neutrons is higher in the fast spectrum model, a smaller reactivity insertion is enough to generate the necessary amount of extra prompt neutrons to make the reactor prompt critical, i.e., critical relying just on prompt neutrons, not delayed ones.

The set of perturbations proposed has enough variety to allow the study of the behaviour of the model in a wide range of situations. It includes both positive and negative reactivity insertions,

Table 3: *Memory usage of the slab models for a transient simulation.*

Type of data	**Memory (MB)**
Cross sections	421.91
Materials	17.66
Others	43055.06
Total allocated memory	**43945.16**
Total available memory	**191848.34**
Memory left for histories	**147903.18**

Table 4: *Comparison of the reactivity insertions (in pcm) introduced in each perturbation of the slab models.*

	Thermal slab	**Fast slab**
-0.8$	-651 pcm	-420 pcm
-0.5$	-407 pcm	-263 pcm
0.5$	407 pcm	263 pcm
1$	814 pcm	525 pcm
2$	1628 pcm	1050 pcm
5$	4070 pcm	2625 pcm

so that it is possible to study if the methods to approximate the dynamic reactivity through the static reactivity calculations are conservative for both types of insertions. The magnitudes include perturbations lower than 1$, so that delayed neutrons keep playing a role in the reactor, and others higher than 1$, that turn the reactor promt critical. Although the perturbations of 5$, and even 2$, are not realistic as they would never be expected to happen in an actual reactor, they represent extreme cases that are useful from a testing point of view, since they will allow to assess the numerical limits of the used methods and its precision when extremely large local perturbations are introduced in a reactor.

An important point of any Monte Carlo simulation is the nuclear data library used to run the simulations. In the slab models, the chosen library has been JEFF (Joint Evaluated Fission and Fusion) 3.3, [51], as it is the latest version of the JEFF library, which is used to perform calculations on the MYRRHA reactor at the time this thesis is written.

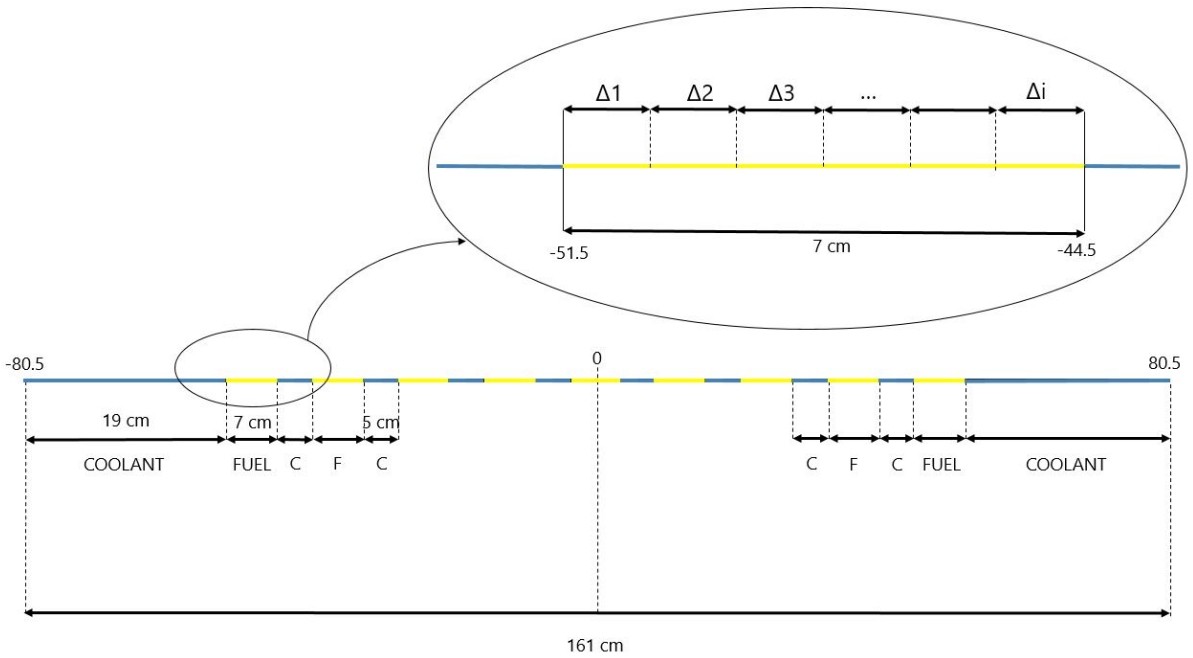


Figure 7: *Geometry and materials of the slab models in Dragon.*

Table 5: *Mesh sizes tested in the study with Dragon.*

	Top and bottom regions	**Intermediate regions**	**Calculation time**
Coarse mesh	20 divisions	5 divisions	5 minutes
Medium mesh	80 divisions	75 divisions	2 hours
Fine mesh	120 divisions	100 divisions	8 hours
Super mesh	175 divisions	150 divisions	21 hours

3.3.2 Characteristics of the Dragon model

To solve the transport problems, *Dragon* uses a modular approach, in which the algorithms are applied one after the other and use the output of one as the input of the following in order to obtain the desired parameters of the system. The first thing one has to do is to define the geometry of the model to be solved. In the case studied, which is a radially infinite slab, it has been defined as a 1-dimensional cartesian mesh with the same axial dimensions as the models defined previously in *Serpent* (see section 3.3.1). There are four aspects required for a 1-dimensional geometry to be correctly defined in *Dragon*. The first is the position of the interfaces between the materials in the axial plane. In the case of the slab, the total axial length is of 161 cm. As the slab is symmetrical, the origin of the axis is located in the centre of the geometry. Therefore, the start and end points of the geometry are located at -80.5 cm and 80.5 cm, respectively. Starting from the point at the bottom, i.e., at -80.5 cm, the first region of the model is composed by coolant, and has a length of 19 cm, so the first interface between the coolant and fuel regions is located

at point -51.5 cm. The next region is made of fuel, and has a length of 7 cm, which means that the second interface is located at point -44.5. From this point on, the regions of coolant (with a length of 5 cm) and fuel (of 7 cm each) are intercalated, so the following interfaces are at -39.5 cm, -32.5 cm, -27.5 cm, etc. A total of nine fuel regions are defined, as can be seen in figure 7, which represents a schematic view of the model defined in Dragon, with the same colour code as the one used for the Serpent model, i.e., blue for the coolant regions and yellow for the fuel regions.

Once the position of the interfaces has been defined, the next step is to determine the mesh size of the model. In *Dragon*, this is done by defining the number of divisions that are to be introduced in each of the regions defined by the interfaces. This is an important parameter to define, since the calculation and the final solution are dependant on the mesh used. If the mesh is too coarse, the solution obtained from the calculation might not converge. By convergence we mean a value that remains constant no matter how the mesh size is further refined. Therefore, the mesh needs to be fine enough to reach that convergence. Nevertheless, calculation time is also affected by the size of the mesh, so it is also interesting to find the size that optimises computation time while allowing to obtain a solution that is mesh-independent. With that objective, some tests have been run to find the optimal mesh size. The procedure followed has consisted in running calculations to obtain the increase of the effective multiplication factors, Δk_{eff}, between the unperturbed configuration and the six perturbed configurations that have been selected for the study in this thesis (see table 4).

The sizes of the meshes used are reported in table 5, and the results obtained with each mesh can be found in table 6. Seeing the results, it appears that the Δk_{eff} calculated with all mesh sizes tends to fluctuate around the same values. Nevertheless, the coarse mesh tends to disagree with the rest of the meshes in the positive perturbations, especially for the perturbations of 5$ and 2$, in which Δk_{eff} appears to be overestimated by a larger difference with respect to the fine and super meshes. For this reason, it has been decided that the coarse mesh would not be used, and instead the medium mesh has been selected due to the agreement between the results obtained with this mesh and the finer meshes, and the lower simulation time of the medium mesh compared with the fine and super meshes (the medium mesh takes approximately 2 hours for a calculation, while the fine needs 8 hours and the super mesh needs 21 hours).

Another aspect of the model that has to be defined is the multi-group energy structure that is used to define the libraries used to solve the neutron transport equation. As the multi-group approach is an approximation that is used in deterministic transport codes, the structure used affects the solutions calculated, so it needs to be adequate in order to minimise the error produced due to the approximation. Five different structures are readily available in the web page where the *Dragon* distribution is downloaded [2], which are made in the required format to be

[2] http://merlin.polymtl.ca/version5.htm

Table 6: Δk_{eff} *calculated for each mesh size.*

	Δk_{eff} **(pcm)**			
Perturbation	**Coarse mesh**	**Medium mesh**	**Fine mesh**	**Super mesh**
-0.8 $	-743	-731	-777	-726
-0.5 $	-442	-437	-479	-445
0.5 $	443	441	386	433
1 $	909	894	837	875
2 $	1803	1768	1703	1736
5 $	4366	4273	4185	4209

Table 7: Δk_{eff} *in a 1$ insertion calculated for each multi-group energy structure.*

	Δk_{eff} **(pcm)**	
Energy structure	**Thermal slab**	**Fast slab**
XMAS-172	894	1834
SHEM-281	873	1837
SHEM-295	879	1841
SHEM-315	908	1850
SHEM-361	916	1841

used with the code. These multi-group structures are XMAS 172-groups, SHEM 281-groups, SHEM 295-groups, SHEM 361-groups and SHEM 315-groups, and are applied to the main nuclear data libraries. Since the *Dragon* models have to be comparable to the models created in *Serpent*, the library used in both codes has to be the same, and in this case the library used is JEFF version 3.3. The selection of the multi-group structure has been made based on a comparison of the results obtained with the structures previously mentioned in the calculation of the Δk_{eff} for a 1$ perturbation. The results obtained in the two slab models for each of the energy structures can be seen in table 7.

From table 7, it can be seen that the selection of the energy structure does not have any significant effect on the result of the calculation. For the thermal slab, a disagreement of 43 pcm is obtained between the SHEM-281 and SHEM-361 structures, which are the ones that yield the most different Δk_{eff}. In the case of the fast slab model, the maximum difference is obtained between the XMAS-172 and SHEM-315 structures, in this case by 16 pcm. Given that the differences can be considered small enough to be negligible in both models, the criterion to select the energy structure is then to minimise the calculation time. The calculation time that has been

obtained does not depend on the model used, as it is approximately the same for the calculations with the thermal and fast slab models alike. It has been found that the lowest calculation time is obtained with the XMAS 172-group structure, and it is of approximately 2 hours and 15 minutes. It is followed by the SHEM 281-group structure, which completes the calculation in 2 hours and 45 minutes. The other three group structures require significantly longer times, between six and eight hours. Therefore, the selected energy structure to be used in both models is the XMAS 172-group structure, as it is the structure with which the calculations are completed with the lowest possible time.

After selecting the appropriate mesh size and multi-group energy structure that allow to obtain a solution that is converged and as independent as possible of those parameters, the models are fully defined and the calculation of the static reactivity using the various methods described in section 4.2 can be performed. The results obtained and the discussion can be found in section 5.1.4 for the thermal spectrum slab model and section 5.2.4 for the fast spectrum slab model.

3.3.3 Materials used

Thermal spectrum model

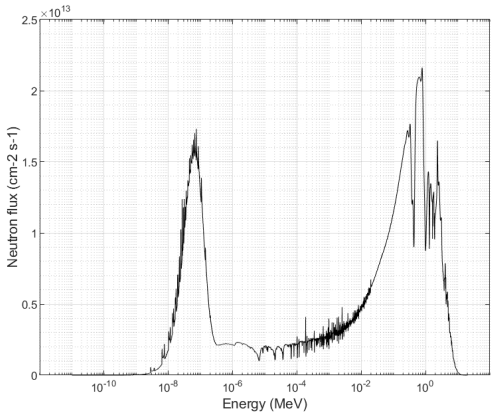
Table 8: *Composition of the materials used in the critical configuration of the thermal slab model.*

Fuel (UOX)		**Moderator (H_2O)**	
Isotope	**Mass fraction**	**Isotope**	**Mass fraction**
^{16}O	0.11848	^{16}O	0.88808
^{235}U	0.011786	^{1}H	0.11191
^{238}U	0.86973		

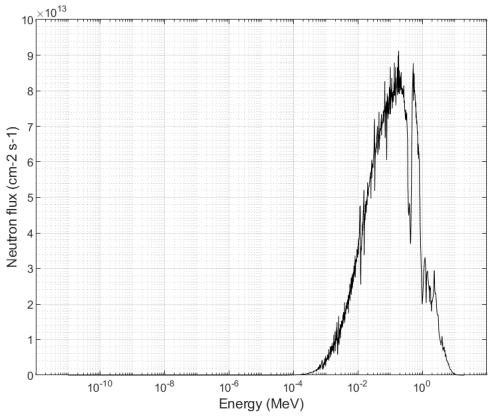
For the system to have a thermal spectrum it is necessary that the neutrons produced in fissions are moderated until they reach thermal energies. Therefore, the fuel regions need to be surrounded by a moderator material. Light water is the most common moderator used in commercial reactors around the world thanks to its great moderating power, as it contains hydrogen which can quickly thermalise neutrons due to its low atomic number. At the same time, water can be used as coolant, as it has a high specific heat capacity which allows it to remove the heat generated in the fuel as a result of fission reactions.

To make the system as close to reality as possible, the materials selected for the thermal slab are UOX fuel and light water as moderator, which can be found in many real-world reactor designs. The exact compositions of the materials can be found in table 8, in which the degree of enrichment of the UOX fuel has been selected so that the system is in a critical configuration. From this base composition, perturbations can be introduced in the system by modifying the fuel composition, either in all of the fuel regions to achieve a homogeneous perturbation or in

only one of the fuel regions, for a localised perturbation.



(a) *Spectral flux in the thermal slab model.*



(b) *Spectral flux in the fast slab model.*

Figure 8: *Thermal spectra of the slab models.*

The slab geometry shown in figure 5, together with the materials specified in table 8 produce a critical system with a thermal spectrum, as can be seen in figure 8a. The spectrum shows two well defined peaks, one at an energy of approximately 1 MeV, which corresponds to the neutrons being born from fission, and another at energies below 1 eV corresponding to the neutrons that have been thermalized in the water. Since the flux obtained with the model has the desired spectrum, the next step is to introduce perturbations in the fission cross section through variations of the enrichment of the fuel in the system.

The detailed compositions used for the UOX fuel in the different perturbations introduced to the thermal system can be seen in table 9. A total of six compositions have been considered, each introducing a different amount of reactivity, with a range from -0.8$ to 5$, to study the behaviour of the system in various scenarios. The magnitude of the perturbation is directly related to the degree of enrichment of the fuel: if the weight fraction of the fissile isotopes (in this case ^{235}U) is increased, the rate of fissions becomes higher and so the production rate of neutrons increases. At the same time, the removal rate of neutrons from the system remains approximately the same as in the unperturbed situation, which results in an increase in reactivity.

Fast spectrum model

Thanks to the materials defined for the fast slab model, the neutron flux has a clear fast spectrum, as shown in figure 8b. Compared with the thermal spectrum (figure 8a), the peak corresponding to the thermalised neutrons disappears, which reflects the differences in moderating power between the coolants used: in the case of water the hydrogen has a great contribution to the moderation of neutrons, while LBE, composed mainly of lead and bismuth, is essentially invisible to neutrons, so they can move through the coolant having a negligible loss of energy and reach the fuel regions at high energies and cause fast fissions.

Table 9: *UOX fuel compositions for the local reactivity insertions tested in the thermal slab model.*

0.5$		**1$**		**2$**	
Isotope	**Mass fraction**	**Isotope**	**Mass fraction**	**Isotope**	**Mass fraction**
^{16}O	0.11849	^{16}O	0.11850	^{16}O	0.11853
^{235}U	0.019481	^{235}U	0.028208	^{235}U	0.046718
^{238}U	0.86203	^{238}U	0.85329	^{238}U	0.83475
5$		**-0.5$**		**-0.8$**	
Isotope	**Mass fraction**	**Isotope**	**Mass fraction**	**Isotope**	**Mass fraction**
^{16}O	0.11862	^{16}O	0.11847	^{16}O	0.11846
^{235}U	0.10312	^{235}U	0.0044076	^{235}U	0.0
^{238}U	0.77826	^{238}U	0.87712	^{238}U	0.88154

Similarly to what has been done in the thermal spectrum case, the fuel configurations have been modified in order to introduce different reactivity insertions to the model. In the case of the fast slab model, this has been achieved by increasing the weight fraction of ^{239}Pu while decreasing the weight fraction of ^{238}U. The fraction of ^{16}O remains constant in all cases. Using this approach, the compositions of the fuels for reactivity insertions of 0.5$, 1$, 2$, 5$, -0.5$ and -0.8$ have been obtained. They are reported in table 11.

Table 10: *Composition of the materials used in the critical configuration.*

Fuel (MOX)		**Coolant (LBE)**	
Isotope	**Mass fraction**	**Isotope**	**Mass fraction**
^{16}O	0.12000	^{6}Li	2.27659E-08
^{239}Pu	0.065050	^{7}Li	2.77180E-07
^{238}U	0.81495	^{10}B	5.96892E-07
		^{11}B	2.40256E-06
		^{107}Aq	1.06769E-05
		^{109}Aq	9.91937E-06
		^{204}Pb	6.22985E-03
		^{206}Pb	1.07242E-01
		^{207}Pb	9.83427E-02
		^{208}Pb	2.33174E-01
		^{209}Bi	5.54987E-01

Table 11: *MOX fuel compositions for the local reactivity insertions.*

0.5$		**1$**		**2$**	
Isotope	**Mass fraction**	**Isotope**	**Mass fraction**	**Isotope**	**Mass fraction**
^{16}O	0.12	^{16}O	0.12	^{16}O	0.12
^{239}Pu	0.07510	^{239}Pu	0.08205	^{239}Pu	0.09510
^{238}U	0.80490	^{238}U	0.79795	^{238}U	0.78490
5$		**-0.5$**		**-0.8$**	
Isotope	**Mass fraction**	**Isotope**	**Mass fraction**	**Isotope**	**Mass fraction**
^{16}O	0.12	^{16}O	0.12	^{16}O	0.12
^{239}Pu	0.12010	^{239}Pu	0.05350	^{239}Pu	0.04510
^{238}U	0.75090	^{238}U	0.82650	^{238}U	0.83490

4 Methodology

In this section, the approaches followed to calculate the dynamic and static reactivity using the models presented in section 3 are presented. The general approach on how to obtain each value is discussed, together with more precise on how is has been applied in the codes *Serpent* and *Dragon* used in this thesis. The approach to obtain the dynamic reactivity is discussed in section 4.1, and the various approaches to obtain the static reactivity can be found in section 4.2.

4.1 Calculation of the dynamic reactivity and neutron population

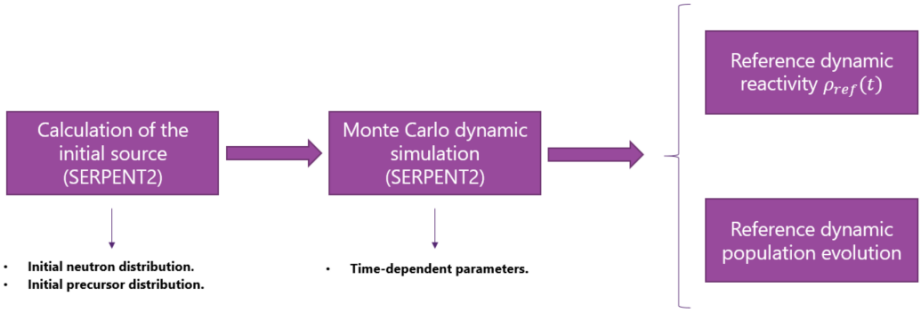


Figure 9: *Proposed approach to obtain the dynamic reactivity and neutron population.*

The reference reactivity is obtained from a dynamic Monte Carlo simulation of the perturbed model. As Monte Carlo simulations are a best estimate method to solve neutron transport problems, it can be considered that a time-dependent simulation of a transient yields the most realistic results of all the possible options to calculate the dynamic reactivity. For this reason, the reactivity obtained from the simulation can be considered a reference value with which the static reactivities that are calculated using static Monte Carlo k-eigenvalue simulations or the perturbation theory approach can be compared.

Following the scheme presented in figure 9, the first step is to run a static Monte Carlo simulation of the system in its initial critical state. With that simulation the initial spatial distributions of both the live neutrons and the precursors that will later produce the delayed neutrons are obtained. In order to obtain representative distributions that are it is important that the system is as close to criticality as possible, and also that a high enough number of particles and cycles are simulated. The main parameters of the calculations performed to generate the sources of the three models presented in section 3 are summarised in table 12.

Since the initial critical configuration is the same for all the cases studied, the calculations to obtain the initial distributions only need to be performed once in each model. The files generated are stored in order to be used in the input of the transient cases that are to be studied.

The next step in the procedure to obtain a dynamic reactivity consists in a second Monte Carlo simulation, which in this case is the proper time-dependent simulation in which the dynamic

Table 12: *Relevant parameters for the calculation of the initial source distributions.*

	Particles per cycle	**Active cycles**	**Skipped cycles**
Thermal slab model	1.5E+05	5000	500
Fast slab model	1.5E+05	5000	500
Heterogeneous MYRRHA model	1E+06	2000	300

evolution of the system is tallied. For that, the necessary parameters have to be set accordingly: they are the initial source distribution, the time structures to be used for the run itself and for the calculated tallies, and the variance reduction techniques that are to be used in the simulation. A more detailed description of the relevant parameters is made in the following sections.

4.1.1 Initial source distribution and batch size

The initial source distribution has been obtained in the first step of the procedure. It is composed of two different distributions, one being the live neutron distribution and the other the precursor distribution. The first gives the location of the prompt neutrons that are present in the system in the beginning of the simulation, which corresponds to the critical state of the system previous to the perturbation that initiates the transient. These neutrons are simulated in the first step of the run, and in turn generate more prompt neutrons and precursors that are simulated in the following time steps. The second distribution mentioned is composed of the precursors that, in time, decay to produce delayed neutrons. They are divided into eight precursor groups, depending on the decay constant.

In the beginning of the transient simulation a number of neutrons of the initial source distribution are sampled to start the run. This number of neutrons must be chosen by the user in the input file, and it defines the total number of particles to be sampled and the number of cycles (or batches, according to the nomenclature used by *Serpent*) are to be simulated. For example, if the input of a time-dependent simulation has a total of 9 million particles to be split into 100 batches, then a total of 90000 neutrons will be sampled from the initial source distribution in each cycle and will be used to run the transient simulation for the specified time, a process that will be repeated 100 times. Ideally, one would like to make the size of the batches as big as possible, because the statistics of the result are better when a large number of particles are simulated, but the memory available for storing the sampled particles and the secondary particles that are generated as the simulation advances is limited by the computational capacity of the cluster that is used for the calculations and by the complexity of the model itself. Therefore, the batch size of each of the models has been maximized in order to achieve the best possible statistics in the result. This has been done in a trial-and-error process, in which the size of the batches has been increased to the maximum number of particles allowed by the code before it fails due to lack of memory.

The batch sizes used in each model are reported in table 13. As can bee seen, this parameter is mainly dependant on the complexity of the model since all the models are run on the same cluster, so the available memory is the same for all of them. In the simplest of the models, which is the thermal slab, as it has a simple 2-D geometry and has only four isotopes in the material definition (^{235}U, ^{238}U, ^{16}O and ^{1}H), the memory usage to load the geometry and the nuclear data of the isotopes is the lowest of the four models, so the space available to store particles is the biggest possible, allowing for a large batch size. As the complexity of the model increases, due to the geometry being 3-dimensional and the material compositions, the size of the batches must be decreased due to a lack of memory.

Table 13: *Batch sizes used in the transient simulations.*

Thermal slab model	50000 particles/batch
Fast slab model	28000 particles/batch
Heterogeneous MYRRHA model	12000 particles/batch

Once the batch size is selected, one can play with the number of cycles to be simulated in order to reduce the final statistic uncertainty of the results. In the end of each cycle the particle buffers are cleared and a new batch is sampled from the initial source distribution to begin the next cycle. This means that the number of cycles run is not affected by the memory limitations: if there is enough memory to run one cycle, then one can run unlimited cycles because the memory available for particles is restored at the end of each cycle. Therefore, the amount of cycles run is only limited by the time required to make the calculation. As a consequence, an appropriate amount of cycles must be found that allows to have good statistics while keeping the running time as low as possible. This has been studied using the MYRRHA model, the results of which can be found in section 5.3.

4.1.2 Time structures

Another of the parameters that has to be set in a time-dependent simulation is the time structures. They define the total time that is going to be simulated, as well as the size of each time step. Three different time structures are defined, one for the run itself, that is used to define the total timespan of the run and how many time steps it will be divided into, and another two for the tallies, whose purpose is to determine the number of subdivisions in which the tallies will be divided. For example, if a tally of the population has a time structure that is defined with a timespan of 1 second divided into 5 time steps, the time will be divided into 5 subdivisions of 0.2 seconds each, and the output of the tally will be formed of five population values, corresponding to the population at the end of each time step, i.e., at time 0.2 s, 0.4 s, 0.6 s, etc.

It is worth noting that the time evolution of the system is not the same for all the perturbations studied. For instance, the neutron population evolution is not the same for an insertion of 1$ and an insertion of 5$: in the former case it will be significantly slower than in the latter because the

reactivity of the system is five times lower, and for this reason the amount of time that needs to be simulated to appreciate the evolution of the system must be larger. Moreover, the magnitude of the perturbation introduced effectively sets a limit to the length of time that can be simulated. When an open loop system, such as the ones studied in this thesis, is turned supercritical, its population grows indefinitely due to the lack of feedback effects. The bigger the population, the more histories need to be simulated, and as secondary particles are generated as a result of the various interactions that can take place, they have to be stored in buffers to wait to be simulated. Since the buffers are predefined in the beginning of the simulation, it is possible that they are overflowed if too many particles have to be stored. When it happens the simulation fails, so the total simulated time needs to be low enough to avoid it. Therefore, the time structures used in each case are different, selected with the goal of adapting to the time evolution of the system for each given reactivity insertion.

It is not an easy task to predict which is the appropriate time to simulate each of the cases, as the exact evolution of the population is unknown. Doing it by trial-and-error is a cumbersome process, as time-dependent calculations are generally slow and memory-consuming, so instead what has been done is to approximately determine which is the appropriate time span for each of the cases using a simple analytical method. When a reactor is prompt critical, the neutron population increases exponentially as:

$$N_n = N_0 \cdot k_{eff}^n \tag{4.1}$$

Where N_n is the neutron population at generation n, N_0 is the initial neutron population and k_{eff} is the effective multiplication factor of the system, i.e., the ratio between the populations of two consecutive generations.

It was decided that the target simulation time for each case would be the amount of time required for the population of the system to be ten times the initial population. If we apply this condition to equation 4.1 it is possible to find the number of generations it would take the system to reach that target population, since the effective multiplication factor is known from the static Monte Carlo calculations of the perturbed systems:

$$10 \cdot N_0 = N_0 \cdot k_{eff}^n \tag{4.2}$$

$$10 = k_{eff}^n \tag{4.3}$$

And finally:

$$n = \frac{ln(10)}{ln(k_{eff})} \tag{4.4}$$

For the cases of a negative reactivity insertion, in which the population tends to decrease over time, the condition that was set is that the population becomes half of the initial population. Then, in a process analogous to the one shown in equations 4.2 to 4.4 but using $N_n = 0.5N_0$ instead, we find the expression:

$$n = \frac{ln(0.5)}{ln(k_{eff})} \tag{4.5}$$

Once the number of generations needed to reach the target population change is known, the last step to do in order to find the target time is to multiply it by the mean neutron generation time (Λ), which is the mean time period between two consecutive generations of neutrons are born. The target is then:

$$t = n \cdot \Lambda \tag{4.6}$$

The mean neutron generation time is considered to be independent of the state of the system, so it is a constant parameter independent of time and effective multiplication factor. Moreover, it is one of the output parameters that are calculated by *Serpent* in a static simulation, so it can be easily obtained. The mean neutron generation time of the models can be found in table 14.

Table 14: *Mean neutron generation time in the models studied.*

Thermal slab model	55.32 μs
Fast slab model	0.2857 μs
Heterogeneous MYRRHA model	0.9257 μs

Note that the mean neutron generation time of the thermal spectrum slab model is noticeably higher than for the rest of the models, which have fast spectra. This is because in a thermal system the neutrons need to be thermalised before they cause their own fissions, so this thermalisation process adds to the time that passes before the next generation of neutrons is born. The values of table 14 indicate that the time evolution of the population thermal system will be significantly slower than in the fast systems, so the dynamic simulations of the thermal system can be simulated for longer time spans.

Tables 15 and 16 show the parameters of the time structures used for the studied cases in the slab and MYRRHA models, respectively. The target simulated times have been calculated using equation 4.6. They are meant to be long enough to observe the population increase and its effect

in the flux of the system and its reactivity, but at the same time not so long that the memory buffers are overflowed. Using this reference times, a slightly longer time has been selected for each of the cases studied. Note that in the case of the null transient, in which no perturbation is introduced to the system, the population should remain constant ($k_{eff} = 1$), so the condition set for the target time is never reached.

Table 15: *Time structure used each of the studied cases of the slab models.*

	Thermal slab model		**Fast slab model**	
Reactivity insertion	**Target time**	**Simulated time**	**Target time**	**Simulated time**
-0.8 $	5.87 ms	20 ms	45.43 μs	150 μs
-0.5 $	10.37 ms	20 ms	79.82 μs	150 μs
0.5 $	31.69	200 ms	26.93 μs	400 μs
0 $	∞	200 ms	∞	400 μs
1 $	15.74 ms	30 ms	132.72 μs	250 μs
2 $	7.55 ms	15 ms	64.84 μs	200 μs
5 $	2.82 ms	5 ms	25.27 μs	50 μs

Table 16: *Time structure used each of the studied cases of the MYRRHA models.*

	Heterogeneous MYRRHA model	
Reactivity insertion	**Target time**	**Simulated time**
Unperturbed	∞	1 ms
Perturbed	10.37 ms	1 ms

As for the time structure of the tallies, they have been divided into two types of tallies. On one hand there are the tallies used in the calculation of the multiplication factor of the system as a function of time, i.e., the tallies that measure the production and loss of neutrons. For a more accurate representation of the evolution of these parameters in time, it is useful to obtain them at more regular times in the transient, and so the time steps for the calculation of these tallies need to be shorter. Therefore, the time structure for production and loss of neutrons has been defined with 20 time steps that divide the total simulation time. On the other hand, the neutron flux is also measured as a function of time. In this case, the interest is not to obtain an exact representation of the time evolution of the tally, but rather to obtain the representation of the flux at a number of instants in the transient. For this reason, a lower number of time steps, three. has been defined for the detectors that tally the neutron flux. As a result, the flux is obtained when the transient has been taking place for 1/3, 2/3 and the total simulated time, with the addition of the flux measured at the beginning of the transient that is calculated in a

static simulation of the critical system and that is common in all cases.

4.1.3 Variance reduction techniques

One of the problems that has been mentioned relative to time-dependent Monte Carlo calculations is the high variability of the fission chain length between histories [52], which might increase the variance of the results obtained. The branchless collision method offers a solution to this problem [53]. The algorithm avoids the variability introduced by the propagation of the several branches that might be generated in a fission event by uniting their contributions into a single history, which will have a weight proportional to that of all the branches. Using this allows to decrease the overall time of the simulation in comparison to a totally analog treatment of the neutron transport process, while keeping the variance of the result at sufficiently low levels.

4.1.4 Parameters tallied in the dynamic simulations

The objective of the dynamic simulations is to study the time-dependent behaviour of the system. With that in mind, the parameters of interest to be tallied are the neutron population, neutron spatial flux distribution and the dynamic reactivity. Obtaining the dynamic evolution of the neutron population and flux distribution is a straightforward process, as it is possible to directly define detectors in the *Serpent* input file that tally it from the simulation. As for the method used to calculate the dynamic reactivity itself from the results of the time-dependent simulation, it is slightly more complex than for a static simulation. While in the static case the effective multiplication factor is one of the parameters that is calculated in the simulation, in the dynamic case the k_{eff} obtained in the output is not a reliable value, and so another technique must be used to calculate it. The method used in this case is the one proposed in [54], which is based in a balance of neutron production over loss in each of the time steps, as follows:

$$k_{eff} = \frac{Production}{Loss} = \frac{P}{F + C - S + L} \tag{4.7}$$

Where P is the fission neutron production, F is the loss of neutrons due to fission reactions, C is the loss of neutrons due to capture reactions, S is the production of neutrons from scattering reactions and L is the loss of neutrons from leakage.

As the multiplication factor obtained from equation 4.7 results from a calculation with tallies that are the output of a Monte Carlo calculation, the mean value obtained does not provide enough information: it must be accompanied by an uncertainty which depends on the uncertainties of the respective tallies. The expression of the relative uncertainty of the multiplication factor $(\delta keff)$ is shown in equation 4.8 [55].

$$\delta_{k_{eff}} = \sqrt{(\delta_{Prod})^2 + \left(\frac{(\sigma_{Loss})^2}{Loss}\right)^2} \tag{4.8}$$

Where

$$\sigma_{Loss} = \sqrt{(C \cdot \delta_C)^2 + (F \cdot \delta_F)^2 + (S \cdot \delta_S)^2 + (L \cdot \delta_L)^2} \tag{4.9}$$

$$Loss = C + F - S + L \tag{4.10}$$

From the result of equation 4.8 it is possible to obtain the absolute uncertainty of the multiplication factor (in pcm) as shown in equation 4.11.

$$\sigma_{k_{eff}} = \left(k_{eff} \cdot \delta k_{eff}\right) \cdot 10^5 \tag{4.11}$$

Finally, since the uncertainty of the reactivity and the multiplication factor are equivalent, the reactivity of the system can be calculated from 4.7 and 4.11 as shown in equation 4.12.

$$\rho = \frac{k_{eff} - 1}{k_{eff}} \pm \sigma_{k_{eff}} \tag{4.12}$$

4.2 Calculation of the static reactivity and point kinetics equations

Although dynamic Monte Carlo simulations are a best estimate tool to calculate the reactivity of a system as a function of time, it is a recently introduced method and is still under development. The time and resources required to run such simulations make them inconvenient, so Monte Carlo is still not the most preferred method to make such calculations. The alternative to such calculations consists in approximating the time-dependent reactivity with a so-called "static reactivity", that measures the reactivity of the system in a specific point in time comparing the current perturbed state of the reactor with the initial critical configuration instead of measuring the dynamic evolution of reactivity with time, which is considerably more difficult.

One of the goals of this thesis is to assess the accuracy, conservativeness and applicability of the static reactivity approximation in comparison with the actual reactivity change that is induced in a reactor when it is perturbed. To do such study it is necessary to have a methodology to calculate the static reactivity, and in this thesis three alternative approaches are proposed: the first approach, which is referred to as $\Delta\lambda$ method, consists in calculating it from the eigenvalues obtained from k-eigenvalue static Monte Carlo simulations. The two alternative approaches

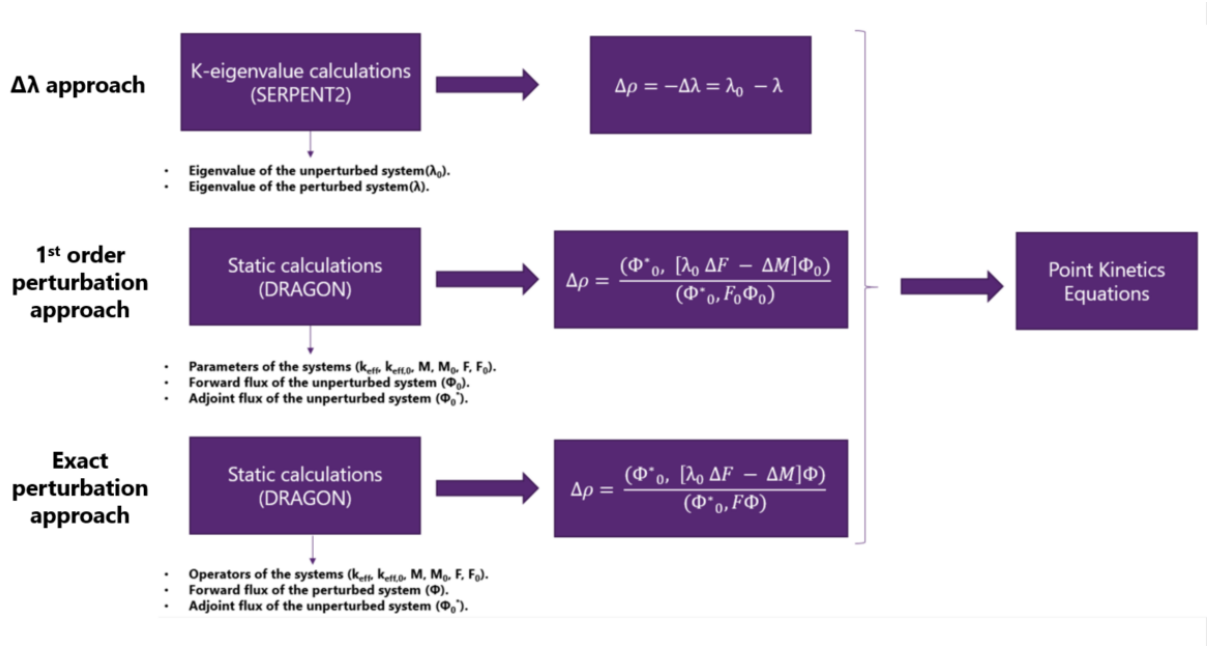


Figure 10: *Proposed approaches to obtain the static reactivity and insert it in the point kinetics equations.*

proposed are based on the application of perturbation theory to the studied system. The interest of the selected methods lies in the fact that the $\Delta\lambda$ method is the one which is currently being used at SCK CEN for the safety calculations that are part of the licensing process of MYRRHA. In the case of the methods derived from perturbation theory, the interest is that it can potentially improve the results of the $\Delta\lambda$ method and help complement its possible shortcomings. The calculated static reactivity is then used as a parameter in the point kinetics equations, which yields the time-dependent evolution of the neutron population.

The approach used to apply the proposed methods and the main considerations that are necessary for that objective are discussed in sections 4.2.1, 4.2.2 and 4.2.3.

4.2.1 $\Delta\lambda$ method

The $\Delta\lambda$ method for the calculation of the static reactivity of a system consists in approximating the change in reactivity to the difference in the eigenvalues obtained as a solution in the k-eigenvalue Monte Carlo simulations of the initial unperturbed system, which is in a critical state, and the final perturbed system. Static reactivity is calculated as:

$$\Delta\rho = -\Delta\lambda = \frac{1}{k_{eff,0}} - \frac{1}{k_{eff}} \tag{4.13}$$

Where $k_{eff,0}$ and k_{eff} are the effective multiplication factors of the unperturbed and perturbed systems, respectively. The results have been obtained performing static Monte Carlo simulations with the models and cases described in section 3, for which the parameters used are re-

Table 17: *Parameters used in the k-eigenvalue static simulations.*

Particles per cycle	1E+05
Active cycles	500
Skipped cycles	100

ported in table 17

Running the static Monte Carlo simulations with the selected parameters has yielded the effective multiplication factor with uncertainties lower than 10 pcm, which has been deemed a good enough statistic for the purposes of the calculations. As for the running time of the calculations, it has been found that it is only dependent on the model being considered for the calculation, and not on the specific perturbation being introduced. For instance, in the thermal slab model the biggest of the perturbation introduced, of 5$, has been found to require 10 minutes, which is the same time required to run the unperturbed case. For the fast slab model, the times required are lower, of just 5 minutes, due to the fact that neutrons of the fast system do not undergo as many scattering reactions as the neutrons in the thermal system do, since they have to be moderated in water. These simulations show that the necessary time for a k-eigenvalue Monte Carlo calculation to be completed with good statistics is convenient even for the most complex models, which is clearly a strong point of the Serpent code and the k-eigenvalue method used to calculate the static reactivity.

Once the eigenvalues of the unperturbed and perturbed systems have been calculated, the only step left to finish the calculation of the static reactivity is introducing the eigenvalues into equation 4.13. The results of the calculation of the static reactivity with the k-eigenvalue method for the different models and cases is reported in sections 5.1.4 and 5.2.4 for the thermal and fast slabs, respectively, together with a discussion of the results obtained.

4.2.2 Perturbation theory method

The second approach followed to calculate the static reactivity is the application of perturbation theory to the equations of the unperturbed and perturbed systems to obtain an approximation of the increment of the reactivity between the two states. As has been explained in section 2.3, this can be done as a first order approximation by applying equation 2.21, or as an "exact" approximation, as it is called in the literature, by applying equation 2.28.

The first order perturbation theory approach has the advantage that only the solution of the unperturbed system needs to be found and no information about the flux of the perturbed system, which is typically difficult to obtain, is required. As the unperturbed system is in a critical state, its solution can be accurately found with a k-eigenvalue Monte Carlo calculation or using deterministic methods, for example. Since the required parameters are not difficult to obtain,

the static reactivity is relatively easy to calculate. Nevertheless, the fact that the actual spatial flux distribution of the perturbed state is not taken into account means that the approximation can fail if the perturbation changes that distribution significantly in comparison with the unperturbed distribution.

The exact perturbation theory approach, on the other hand, has the advantage that it considers the flux spatial distribution of the perturbed system, and therefore it can approximate the change of reactivity with more precision than the first order approximation case, even for local perturbations that modify the shape of the flux distribution. The problem with this method is that the exact perturbed shape of the flux spatial distribution is difficult to obtain. A way to do so would be to obtain it by means of a dynamic Monte Carlo simulation, for example, which would represent a best estimate solution of the perturbed problem. Nevertheless, doing so would defeat the purpose of finding a static reactivity that approximates the real time-dependent reactivity, since a dynamic Monte Carlo simulation already allows to find the time-dependent reactivity. Instead, what can be done with equation 2.28 is to use a Φ that does not have the exact perturbed shape of the flux, but is similar enough to it that the $\Delta\rho$ calculated results in a good approximation, even in the case of local perturbations. This alternative Φ can be obtained with several approaches, for instance a k-eigenvalue static calculation of the perturbed system or, as has been done in this thesis, with a deterministic code.

The approach followed to obtain both the first order and the exact static reactivities has consisted in the use of the deterministic transport code *Dragon*. The main reason for using this code, and not the Monte Carlo code Serpent, is that the perturbation theory approach requires the adjoint flux distribution of the unperturbed system, Φ_0^*, as one of the input parameters, and Serpent does not offer a way to obtain it as an output of its simulations. Therefore, an alternative has had to be found for obtaining the adjoint flux, and *Dragon* has been chosen as it allows to obtain all the required parameters and includes a module that applies the perturbation theory approach.

The slab cases that have been used for the study in this thesis, which are radially infinite, have been modelled in *Dragon* as 1-dimensional geometries for the sake of simplicity in the calculations. The process followed for calculating the static reactivity consists in two main steps: first, the calculation of the solutions of the perturbed system and the unperturbed system, which yields the forward and adjoint fluxes, as well as the operators F and M, followed by the application of the perturbation theory equations 2.21 (first order approximation) and 2.28 (exact approximation). The first step is common in the two approaches followed to apply perturbation theory, while the second presents some differences between the the two approaches, which is explained later in this chapter.

Once the geometry and the material mixes have been defined in the input file of Dragon, the first thing to do is to call the tracking module. Its function is to analyse the geometry and to generate integration lines and tracking lengths of the system. As the slab models have been defined as 1-dimensional cartesian geometries, the most useful tracking module to use is the *SYBILT* module

[56], which uses the interface current method and is specially convenient for 1-dimensional plane geometries. This module generates a tracking data structure with the necessary surface and area vectors that is later used in the modules that are run after the tracking module.

Once the tracking data structure has been generated, the following step is to run the self-shielding module *SHI*. The function of the module is to apply the generalised Stamm'ler method [57] and modify the data from the microscopic cross sections library to adapt it to the current model, so the output of the model is a modified version of the library that was initially created when the definition of the material mixtures is made.

When the tracking structures have been created and the microscopic cross section libraries are modified with the self-shielding module, the assembly module, *ASM*, is called. It computes the collision probability matrices from the tracking file previously generated. This matrices are used in the final step, which is the solution of the fission source eigenvalue problem that yields the forward flux spatial distribution of the studied system. This calculation is performed by the module *FLU*.

This process is performed two times, first for the unperturbed system and then for the perturbed one. From these, the unperturbed and perturbed forward fluxes are calculated and stored. Next, the adjoint fluxes are calculated. The process is analogous to the calculation of the forward fluxes, but in this case the matrix of the cross sections used for the calculation of the flux is transposed. The reason is that, by definition, the adjoint matrix A^* of a real matrix A, equals the transposed matrix A^T. After applying this modification and solving the adjoint problem, the adjoint flux of the unperturbed system is obtained.

After the required fluxes have been obtained, all that remains is to calculate the static reactivity. One of the reasons to select Dragon to do this is that it includes the module *DUO*, which takes as the input the solutions of the eigenvalue problems of an unperturbed and a perturbed system and uses them to compute the exact approximation equation, i.e., equation 2.28). Nevertheless, the module does not include the calculation of the first order approximation (equation 2.21, so the source code of dragon has been modified in order to include this calculation.

To do so, the file of the *DUO* module that defines the computation of the exact approximation has been modified to include the lines required for the calculation of the first order approximation. Equations 2.28 and 2.21 only differ in the parameters they use, so the modification of the code has been easily introduced. With this modification it has been possible to calculate the static reactivity with the perturbation theory using the first order approximation approach and the exact approximation approach. The discussion of the results obtained can be found in section 5.1.4 for the thermal slab model and 5.2.4 for the fast slab model.

4.2.3 Point Kinetics Equations

Once the static reactivity has been obtained with either of three methods proposed, the neutron population evolution with time can be calculated by inserting the reactivity in the system of equations defined by equations 2.35 and 2.36. The rest of the kinetic parameters required for the calculation are those of the unperturbed system. They can be easily obtained by performing a k-eigenvalue static Monte Carlo simulation with *Serpent*, which yields the parameters of interest in the output file.

There are several numerical methods for to solve the system of equations, for instance the Padé approximations [58] or the generalised Runge-Kutta method [59]. In this case, thoguh, the solution of the system is found by using a publicly available *Matlab* script that solves the point kinetics equations with the use of piecewise constant approximations, whose main advantage is its simplicity [60]. With it, a solution of the point kinetics equations can be found, which yields the dynamic evolution of the neutron population in the system.

5 Results and discussion

This section contains the report of the results obtained in the studies performed on each of the models described, as well as the discussion of said results and an evaluation of their validity and shortcomings.

The first model evaluated is the slab model with thermal energy spectrum. In this model an evaluation of the effects of global and local perturbations has been performed. To do so, the spectral and spatial shift of the flux after the introduction of the perturbations has been studied by means of static Monte Carlo simulations. The dynamic behaviour of the model has also been studied, in this case by performing dynamic Monte Carlo simulations. The study includes the time-dependent evolution of the flux, divided into thermal and fast energy groups, as well as the neutron population evolution and the dynamic reactivity of the system. Finally, a calculation of the static reactivity has been performed using the different approaches described in section 4.2. The differences between the solutions found for local and global perturbations have been discussed in order to assess the usefulness of each approach for the different cases. The calculations have been done using the deterministic transport code Dragon. The static reactivities obtained have been used as input parameters of the point kinetics equations to obtain a time-dependent evolution of the population. The results obtained are discussed in this section. Moreover, the solutions found using static approaches are also compared to the reference solutions obtained with time-dependent Monte Carlo simulations to determine their conservativeness.

For the fast slab model, the study performed is the same as the one described for the thermal slab model, with the difference that in this case only local perturbations have been studied.

Finally, the heterogeneous MYRRHA model, described in section 5.3, is also included in the discussion. The aim of the study performed with this model is to determine the full extent of the computational capabilities of the cluster present at SCK CEN, called Newton. To do so, a study of the maximum number of particles that can be simulated in the cluster given the available memory has been done, as well as the uncertainty levels that can be reached and its dependence on the number of particles and cycles simulated. Finally, an assessment of the effect of the parameters used in the simulations on the total running time has also been done. The results are followed by a discussion of the shortcomings of the cluster and the requirements that would need to be fulfilled for the application of full-core time-dependent Monte Carlo simulations to the safety calculations of MYRRHA.

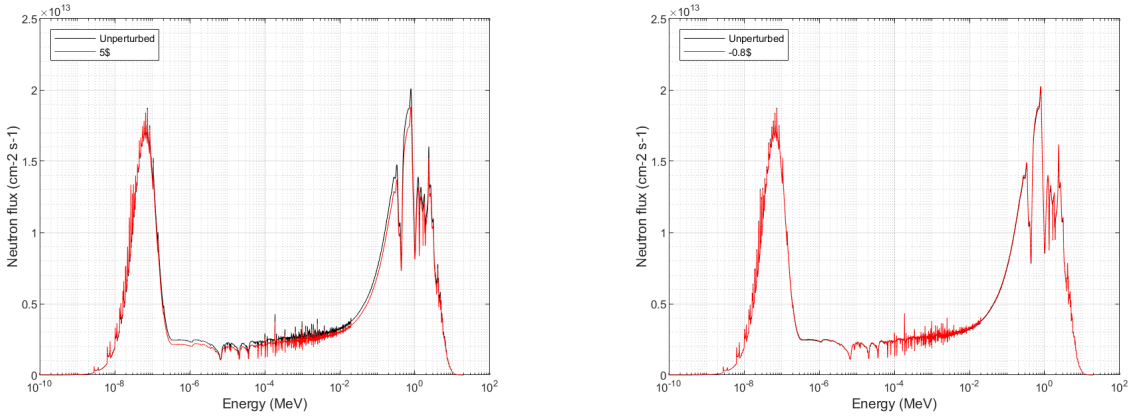
5.1 Thermal slab model

5.1.1 Spectral shift in the flux

As has been discussed in previous sections, one of the conditions for perturbation theory to be applied to calculate the static reactivity of a system is that the spectral distribution of the flux does not suffer a shift when the perturbation is introduced in the system. The first step

of the study is, therefore, to obtain the spectral distribution of the fluxes present in the system once they have been perturbed and compare it with the distributions of the unperturbed systems. This has been accomplished by performing static calculations of the slabs in their unperturbed states and the perturbed states described in section 3.3.1 and tallying the spectral distribution of the flux by defining an energy grid. The grid is formed by 3000 bins with constant lethargy-width from 10^{-11} MeV to 20 MeV, enough to cover all the energy spectrum of the neutron population inside the reactor.

The results obtained for the thermal spectrum slab are shown in figure 11. For the sake of clarity in the graphs, not all the perturbations have been included in the figure. Instead, only the shifts obtained from the biggest positive and negative perturbations are displayed (i.e., the 5$ and -0.8$ insertions). Since they are the most extreme cases studied, the shifts caused by all the other perturbations considered are not as pronounced, so they are not shown in the figures.



(a) Positive reactivity insertion.

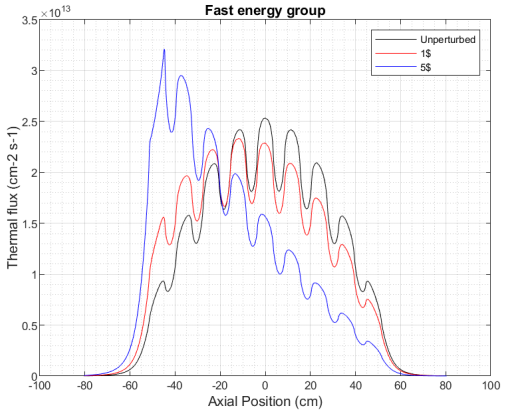
(b) Negative reactivity insertion.

Figure 11: Spectral flux shift in the thermal slab model.

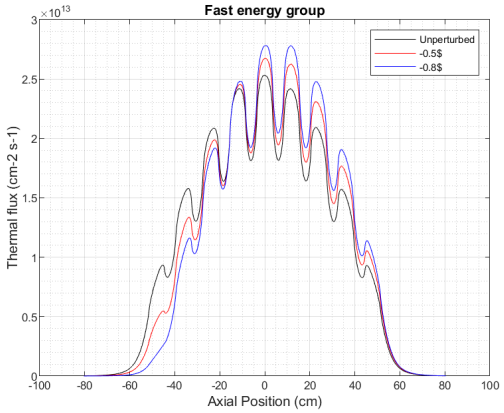
As can be seen in figure 11, the spectral flux does not suffer a noticeable shift when the local perturbations are introduced, regardless of the type of insertion, positive or negative. While it is true that in the case of the 5$ insertion the flux at energies above 10^{-6} MeV slightly decreases in comparison with the spectral flux obtained for the unperturbed system, the peaks obtained are located in the same energy bins, so the energy distribution is unchanged.. This difference in the absolute value of the fluxes could be attributed to the uncertainties associated to a stochastic calculation, or perhaps to the inexactitude of k-eigenvalue calculations in systems that are far away from criticality, as is clearly the case for the slab with a 5$ perturbation.

The results are encouraging, as they show that both models keep the same energy spectrum when they are perturbed, which is one of the necessary conditions for applying perturbation theory in order to calculate the static reactivity. Therefore, the methodology described in section 4 can be applied to the slab model.

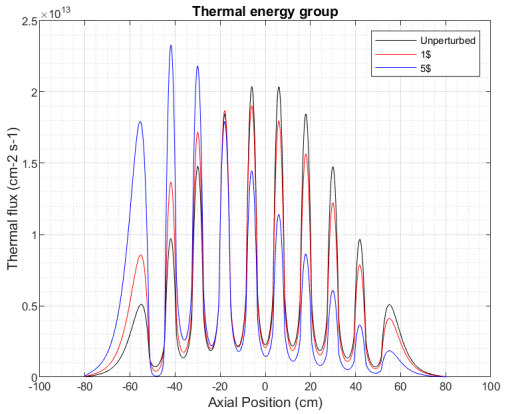
5.1.2 Spatial shift in the flux



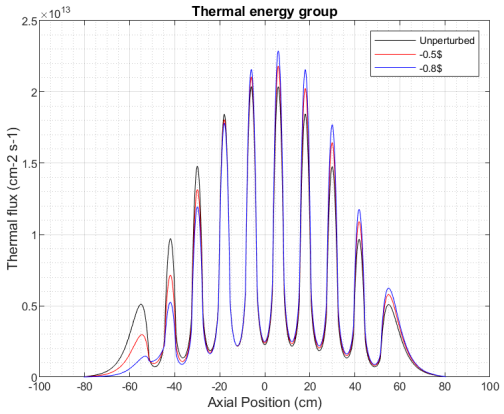
(a) Positive reactivity insertion.



(b) Negative reactivity insertion.



(c) Positive reactivity insertion.



(d) Negative reactivity insertion.

Figure 12: Spatial flux shifts in the thermal slab model after local perturbations.

Figure 12 shows the shifts in the spatial distribution of the neutron flux following local perturbations in the slab as explained in section 3.3.1, i.e, by modifying the fuel composition of the fuel region located at the bottom of the slab. As the slab has a thermal spectrum, it is useful to separate the flux into two energy groups, one for thermal neutrons and another for fast neutrons. The former group contains those neutrons with energy between 0 eV and 0.625 eV, while the latter contains the neutrons that have an energy above 0.625 eV. This energy cut-off is the most commonly used in thermal systems for which the spectrum is collapsed into a 2-group energy structure, and is conveniently included in the pre-defined energy structures available in *Serpent*.

The spatial distributions of the flux present in figure 12 clearly how how the flux of thermal and fast neutrons is distributed along the slab. The fast neutrons are mainly present in the fuel regions, as they are born there as a result of fission reactions between thermalised neutrons and ^{235}U, mostly, although there is also some contribution of ^{238}U. Therefore, the distributions of the neutron flux in the fast group (figures 12a and 12b) show peaks located in the areas where

the slab contains UOX fuel. In total, nine peaks can be observed, as there are nine fuel regions in the model (see figure 5). Outside of the fuel regions, where the coolant can be found, the fast neutrons that come from the fuel are moderated as a consequence of scattering reactions with hydrogen or oxygen atoms, so the presence of neutrons with high energy is not as high as in the fuel. Hence, the decrease observed in the coolant regions.

The flux distribution in the thermal energy group (figures 12c and 12d) can be interpreted in a similar way as the one in the fast energy group. In this case, the peaks can be observed in the regions that contain coolant, as the fast neutrons lose their energy when they interact with the water and eventually move to the thermal energy group. On the other hand, the thermal neutrons that enter the fuel are quickly absorbed by the ^{235}U thanks to its high fission cross section at thermal energies, so the amount of low energy neutrons that can be found in the fuel is much lower. While the maximum values observed in the flux of the two energy groups are in the same order of magnitude, in the case of the thermal spectrum the peaks in the flux distribution are significantly sharper than for the fast flux. This is due to the high absorption of thermal neutrons in the fuel that significantly decreases the low-energy neutron population in those regions, something that is not observed on the high-energy neutrons that are present in the water, as they are mostly scattered, so the flux in that region does not decrease as much.

Both the thermal and fast fluxes are symmetric when the slab is unperturbed, as both the geometry and the material compositions of the model are symmetric in the axial direction. Therefore, the maximum value of the flux is observed in the center of the geometry. Nevertheless, when a local perturbation is introduced the symmetry of the model is lost, as the material composition of one of the fuel regions (located between -51.5 cm and -44.5 cm) is modified. This induces a shift in the spatial flux distribution, which can be observed both in the thermal and fast energy groups. The direction of the shifts depends on the sign of the perturbation introduced, i.e, if the perturbation introduces a positive or a negative reactivity insertion. If the perturbation introduced is positive, the fission rate of the modified fuel region increases. As a consequence, the amount of fissions in that region is higher, so the fast neutron population in that area increases, as more of them are being born from fissions. This means that the fast neutron population tends to concentrate on the lower part of the slab (or the left side of the distributions of figure 12), and the flux is shifted towards that position. The behaviour of the thermal flux is similar, as an increase in the population of fast neutrons in the modified fuel region increases the amount of scattering reactions in the water that surrounds it, thermalising a bigger amount of fast neutrons and thus increasing the population of thermal neutrons, which can be seen in the higher peaks of the thermal flux the negative axial positions of figure 12c. Nevertheless, the higher fuel enrichment also increases the thermal neutron capture in the modified fuel, so the thermal neutron flux in that region decreases in comparison with the unperturbed case. As can be expected, the magnitude of the shift increases as the perturbation is higher, since the shift for the 5$ insertion is significantly greater than for the insertion of 1$.

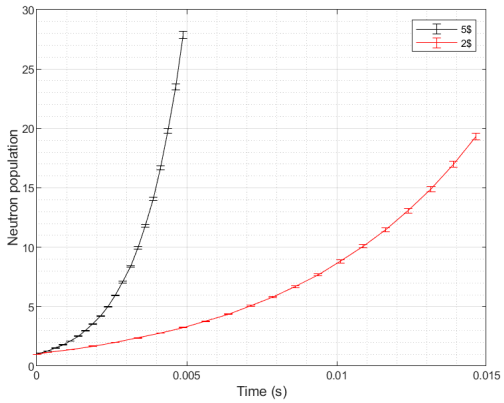
As for the negative reactivity insertions (figures 12b and 12d), the behaviour of the flux shift can be explained analogously to the positive reactivity insertions, but in the opposite direction: in this case, the decrease in the enrichment of the fuel decreases the fission rate of the modified fuel, so the neutron populations tends to decrease in the lower axial positions and increase in the upper regions. Therefore, a shift of the fast and thermal fluxes can be observed towards the positive axial regions. Again, this shift is bigger for stronger perturbations, similarly to what has been seen for the positive reactivity insertions.

5.1.3 Dynamic evolution of the system

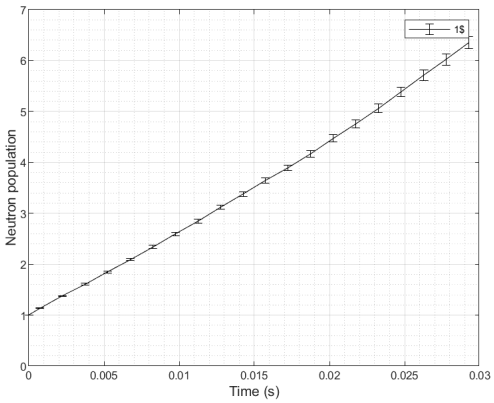
The time-dependent evolution of the system has been obtained from dynamic Monte Carlo simulations. The selection of the main parameters has been done with the objective of maximising the use of the available memory and minimising the uncertainty of the results. At the same time, a selection of the length of time simulated has been done so that it is long enough that the population evolution can be appreciated, but without being so long that the increase in neutron and precursor population overflows the predefined buffers. As will be seen in this section, this is of special importance for the bigger perturbations, i.e., greater than 1$, since they make the reactor prompt-critical and the population increases exponentially, so the buffers can be very quickly overflowed.

The main objective of the dynamic simulations with the thermal slab models is to obtain the evolution of the reactivity of the system with time in order to use it as a reference best estimate reactivity to benchmark the static reactivities for the very same system calculated as the difference between the eigenvalues obtained from solving the static k-eigenvalue problems of the unperturbed and the perturbed system, as is being currently done, and the alternative perturbation theory approach proposed in this thesis. It will be used to assess the precision of the calculated static reactivities, as well as their degree of uncertainty, important for the safety studies required to license MYRRHA. The other parameters of interest that have been obtained in the dynamic simulations are the evolution of the population of neutrons in the slab and the neutron flux along the length of the slab, which has been obtained differentiating the flux in two neutron energy groups, thermal and fast.

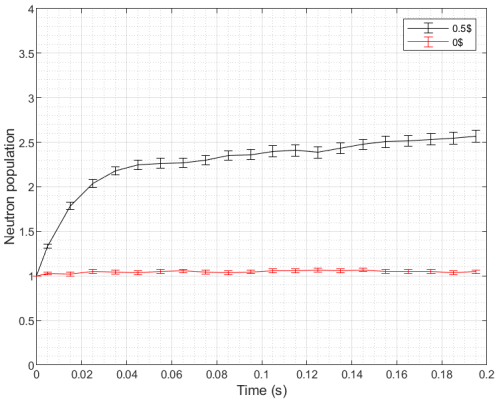
Neutron population. First, the neutron population evolution will be addressed, as the experience obtained in this thesis has shown that it is probably the most important parameter for the successful completion of any time-dependent simulation. The reason is that it essentially limits the length of time that can be simulated in a model for a given perturbation, as the neutrons and precursors that are present in the system have to be stored using part of the allocated memory to be simulated eventually. If the defined timespan of the simulation is too long, the increase of population that follows a positive reactivity insertion can result in too much memory being required to store particles, which causes the simulation to immediately fail. For this reason, this parameter has been used to define the length of the simulation of each insertion.



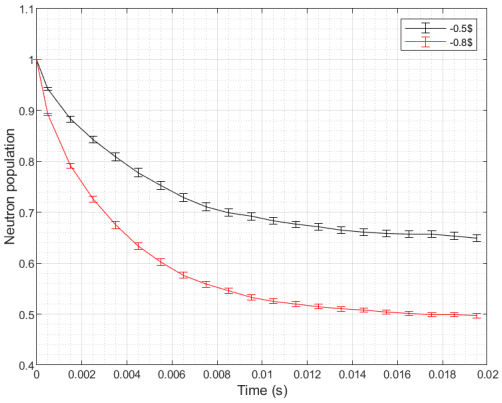
(a) *2$ and 5$ insertions.*



(b) *1$ insertion.*



(c) *0.5$ and 0$ insertion.*



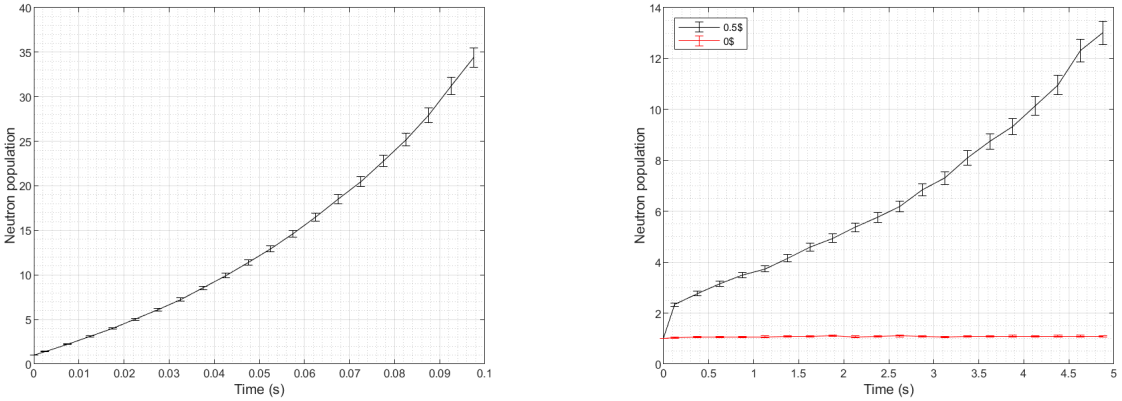
(d) *Negative reactivity insertions.*

Figure 13: *Dynamic population evolution of the thermal slab model.*

Figure 13 shows the evolutions simulated in each of the reactivity insertions studied. For the reasons explained, each of the insertions has been simulated for different time spans, so they have been grouped in various plots according to the length of the simulations to be observed more easily. The population evolution is plotted normalized to the initial population (i.e., the population of the unperturbed system, calculated in a static Monte Carlo simulation) and with the 2-σ uncertainties obtained in the simulations. Therefore, in all cases the population at the initial time point is set to unity, and the population at each instant can be interpreted as the factor by which the initial population has been multiplied. As all are normalised to the same initial value, the velocity in the increase of population can be easily compared between different insertions.

As can be seen in figure 13, the behaviour of the neutron population in the system depends greatly in the magnitude of the perturbation that is introduced. For big perturbations that make the system prompt critical, such as the insertions of 5$ and 2$, shown in figure 13a, the neutron population immediately starts increasing exponentially, since the system only requires the prompt neutrons generated in the fission reactions to maintain the fission chain, and therefore

the effect of the delayed neutrons cannot be observed. The velocity in the exponential increase in the neutron population depends on the k_{eff} of the system. It can be clearly seen in the comparison of the 5$ and 2$ insertions, where the population of the former case increases extremely quickly, increasing the initial population by a factor of 28 in only 50 ms. In the case of the 2$ insertion the exponential increase has a slightly slower evolution than the 5$ case due to the lower fission reaction rate, which generate fewer prompt neutrons, but still the population is multiplied by a factor of 20 in 15 ms. Although they do not represent realistic cases that would occur in actual reactors, as they are extremely big insertions, this serves as an example of the difficulties of controlling a prompt critical reactor, as the population of neutrons, which affects the neutron flux and the power output of a reactor, increases uncontrollably in an extremely short timespan making the control of such an excursion very challenging.



(a) *1$ perturbation.* (b) *0.5$ and 0$ perturbation.*

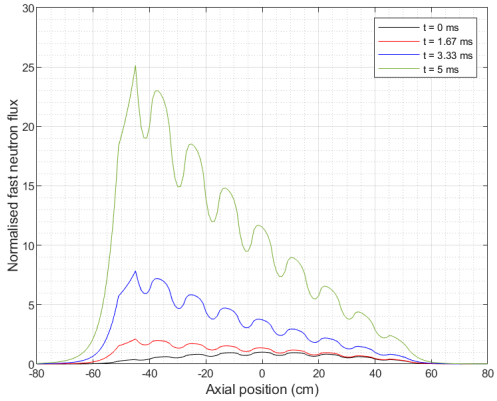
Figure 14: *Neutron population evolution of the 1$ and 0.5$ cases with longer time spans*

The case of the insertion of 1$ (figure 13b) still represents a case of a prompt critical system, but in contrast to the cases that appear in figure 13a the evolution appears to be linear instead of exponential, at least in the first 30 ms of the transient. As this insertion still makes the system prompt critical, the population would be expected to increase exponentially in a similar way as the other insertions greater than 1$. The reason is that the simulated timespan is too short to be able to appreciate the exponential shape of the transient, since the period of the system is larger than in the case of the bigger insertions, and so the population increase is not as fast. Indeed, when the same system is simulated for a longer timespan, in this case 100 ms, the exponential increase can be better appreciated (see figure 14a) and the neutron population further increases by a factor of 35.

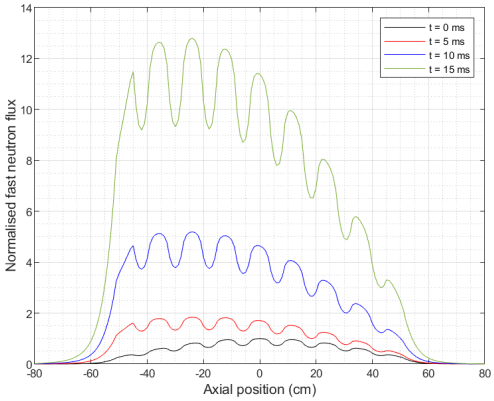
Figure 13c shows a different behaviour compared with the prompt critical cases. In this case, for a 0.5$ insertion, the neutron population has a sudden increase in the first 40 ms of the transient, increasing the initial population by a factor of 9. After this point, the population seems to stabilise, and it keeps increasing but in a seemingly linear way with a slight slope. 200 ms after the beginning of the transient, the population is 10 times as high as in the beginning. The

reason is that the reactivity insertion is lower than 1$, and so the reactor is prompt subcritical. Therefore, the prompt neutrons are not enough to maintain the fission chain, so the delayed neutrons are required. As the decay time of the precursors that generate the delayed neutrons is long in comparison with the time required for the prompt neutrons to create their own fissions, the increase in population is slower than in the prompt critical cases. The initial jump in neutron population is due to the effect of the fissions from the prompt neutrons, which quickly increase the population, but once they have caused fissions the system needs to "wait" for the delayed neutrons to be born and cause their own fissions to keep the chain reaction alive. To see the effects of the delayed neutrons it is necessary to simulate the transient with a longer timespan. In this case increasing the simulated time is not a problem for the memory usage, as the slower increase in neutron population means that the buffers where particles are stored do not overflow as quickly as in the prompt critical cases. A second run of the 0.5$ case has been made, this time with a timespan of 5 seconds, which is long enough for the precursors to decay and generate delayed neutrons. The result of the run can be seen in figure 14b, and shows that a longer simulated time results in the population increasing exponentially after the initial prompt jump, but in a much slower rate than in the prompt critical cases due to the effect of the delayed neutrons. After 5 seconds the population has only increased by a factor of 13, which serves as a practical demonstration that the length of time that can be simulated in a dynamic Monte Carlo simulation depends greatly on the type and magnitude of the insertion and how it affects the population of neutrons in the system.

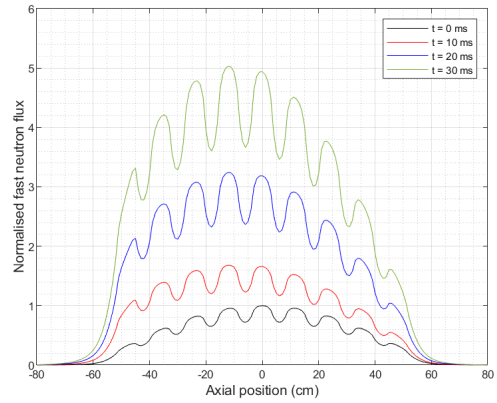
As for the population evolution for the negative reactivity insertions, they can be seen in figure 13d. Since with this insertions the system is made subcritical, the population decreases immediately after the beginning of the transient. Similarly to what has been seen for the positive 0.5$ insertion, a prompt jump is initially produced as the insertion is introduced to the system, followed by a stabilisation of the population for the rest of the transient. The decrease is significantly stronger for the larger perturbation, in the figure -0.8$, as the fission rate is decreased more than for the -0.5$ case, but in both cases the prompt jump has a similar duration in time, until approximately 120 ms. As for the point where the population stabilises, in the case of a -0.5$ insertion the population reaches a value of 65% of the initial population, and in the case of -0.8$, it reaches a vale of 50% of the initial population.



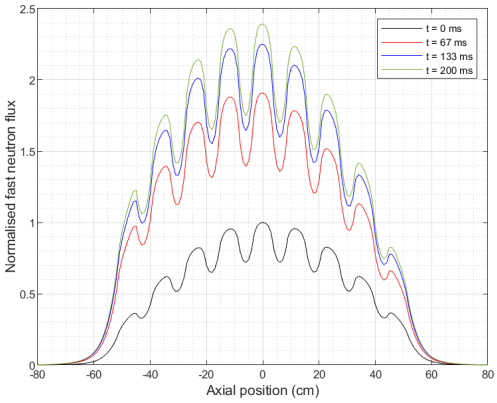
(a) *5$ insertion.*



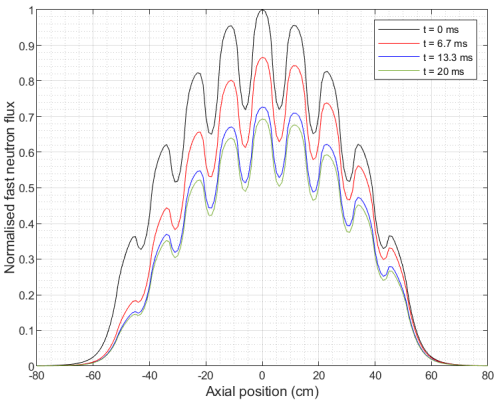
(b) *2$ insertion.*



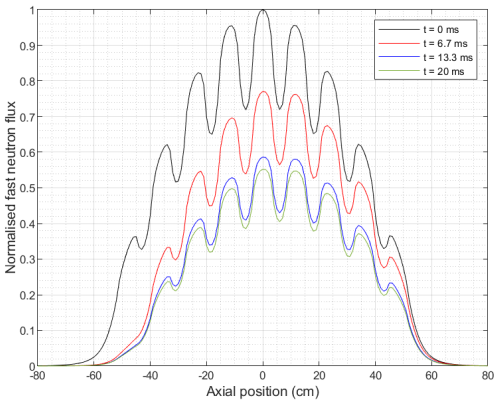
(c) *1$ insertion.*



(d) *0.5$ insertion.*



(e) *-0.5$ insertion.*



(f) *-0.8$ insertion.*

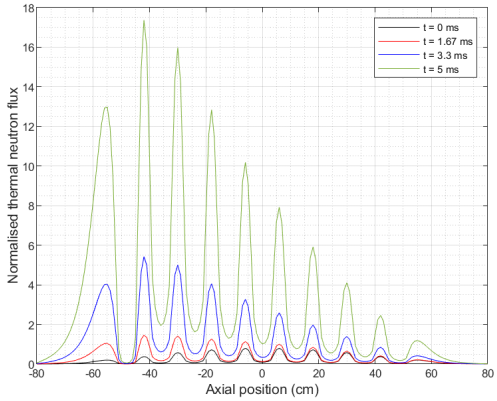
Figure 15: *Fast flux evolution over time in the thermal slab model.*

Spatial flux distribution. The second parameter that has been tallied in the dynamic simulations is the neutron flux along the axial direction of the slab, divided in two energy groups to differentiate between the neutrons in the thermal and fast energy ranges. Figure 15 shows the evolution of the fast flux over time for the six reactivity insertions that have been studied. It

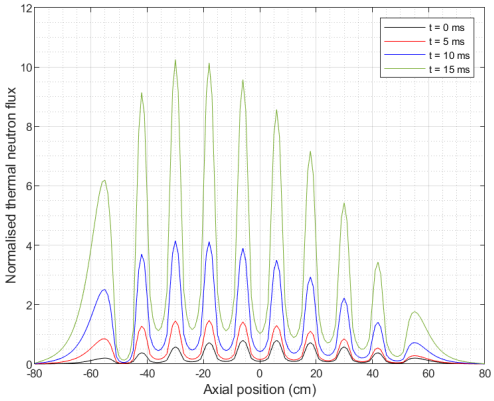
can be appreciated that the local perturbations induce a shift in the spatial distribution of the flux, which appears inmediately in the transient because the perturbation is applied as a step at time t = 0 s. The perturbation tends to displace the flux towards the negative positions in the axial direction of the slab (the left side of the figures) in the case of positive reactivity insertions, while in the negative reactivity insertions the flux tends to move towards the positive axial positions (to the right side of the figures). This behaviour is the same that has been obtained in the static Monte Carlo simulations previously performed with the thermal slab model, reported in figure 12, which shows that static simulations are a good tool to make qualitative predictions of such shifts in the flux when local perturbations take place. Nevertheless, dynamic Monte Carlo simulations offer more information because they also take into account delayed neutrons and the evolution of the neutron population over time, which means that the increase or decrease of the actual value of the flux can also be known. This can be seen in figure 15, where apart from the spatial shift also the change of magnitude (normalised to the flux of the unperturbed system) is displayed.

In the thermal group the behaviour of the flux is analogous to what is observed in the flux of the fast energy range. It can be seen that for the positive reactivity insertions the flux tends to shift to the left side of the plots and to increase in magnitude, while for the negative reactivity insertions it tends to have a displacement towards the right side of the plot and the magnitude of the flux decreases with time. Nevertheless, the variations can only be seen in the peaks of the graphs, which corresponds to the water regions of the slab. In the valleys of the plots, corresponding to the fuel regions, the flux has much smaller variations, probably due to the fact that although the thermal neutron population increases (in the case of the positive insertions), so does the fission rate in the fuel, meaning that more neutrons are absorbed so the production/loss balance in the fuel regions remains approximately constant over the transient. Similarly, for negative insertions the population decreases, but so does the fission rate which makes the balance remain more or less constant. In contrast, in the water regions only the change in the population takes place, but the neutron losses remain constant so the variations in the flux can be seen more clearly.

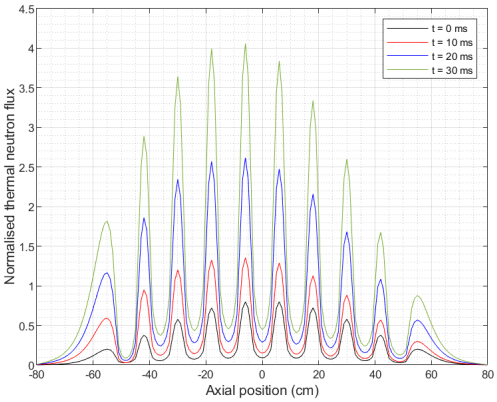
The evolution of the flux is closely related to the evolution of the neutron population. By comparing figures 13 and 15, it can be easily seen that in the cases where the population increases exponentially (5$ and 2$), the magnitude of flux has the same behaviour, with slight increases at lower times and bigger changes as time increases. For 1$, where the neutron population appears to increase linearly, the flux does the same, showing approximately constant increases as time increases. For the insertions lower than 1$, the behaviour observed in the population was an initial prompt jump followed by a stabilisation. This can also be seen in the flux, with a bigger change between the unperturbed flux and the first time step, followed by smaller changes when the population stabilises.



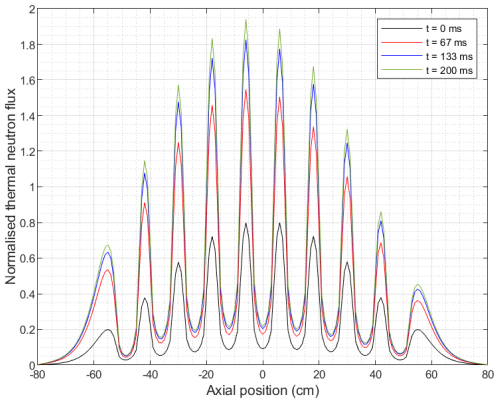
(a) *5$ insertion.*



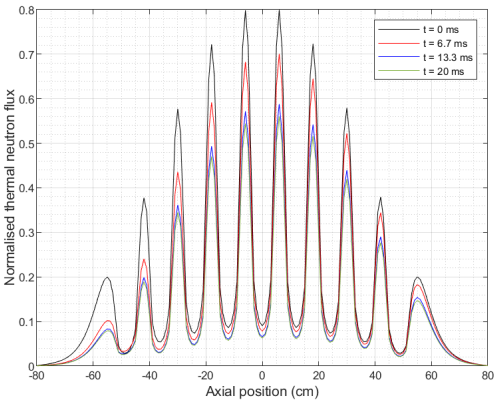
(b) *2$ insertion.*



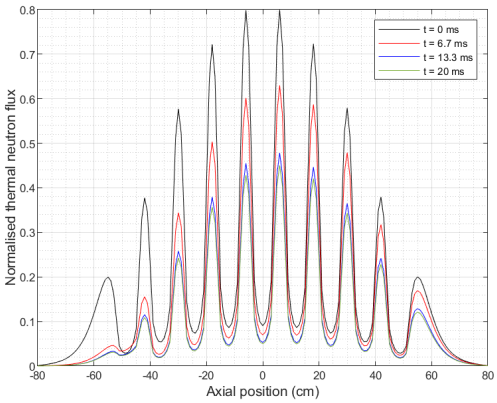
(c) *1$ insertion.*



(d) *0.5$ insertion.*



(e) *-0.5$ insertion.*

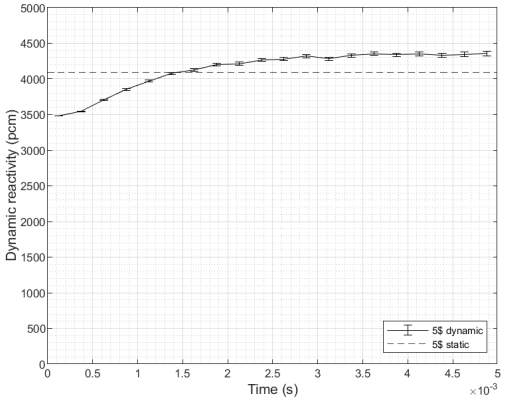


(f) *-0.8$ insertion.*

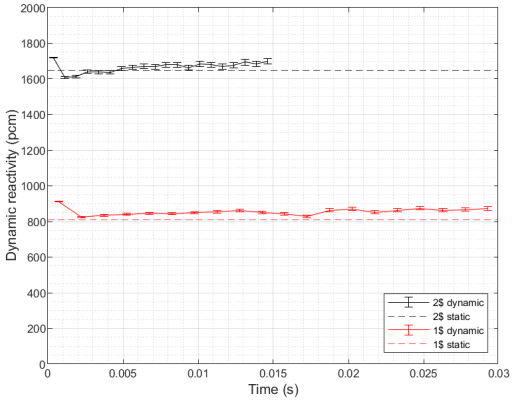
Figure 16: *Thermal flux evolution over time in the thermal slab model.*

Reactivity. Finally, the last dynamic parameter that has been tallied in the time-dependent calculations of the thermal slab model is the dynamic reactivity of the system. In contrast with static Monte Carlo calculations, where the multiplication factor (k_{eff}) of the system is calculated as an eigenvalue and is printed as an output parameter, it cannot be obtained from a dynamic

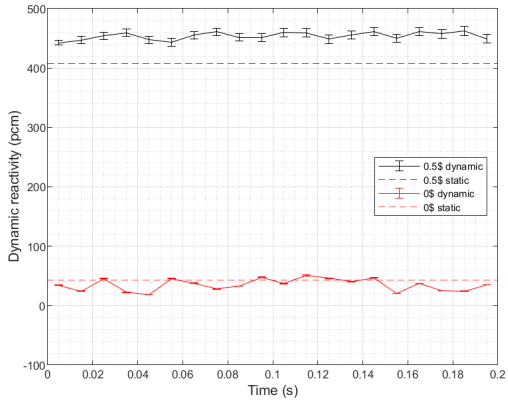
calculation as a time-dependent parameter. Therefore, k_{eff} is calculated indirectly as the ratio between the production and loss of neutrons in a time step (see section 4.1 for a more detailed explanation). The reactivity is then obtained from the multiplication factor as is shown in equation 4.12. The plots of the obtained reactivities as a function of time can be found in figure 17, according to the length of the simulated time.



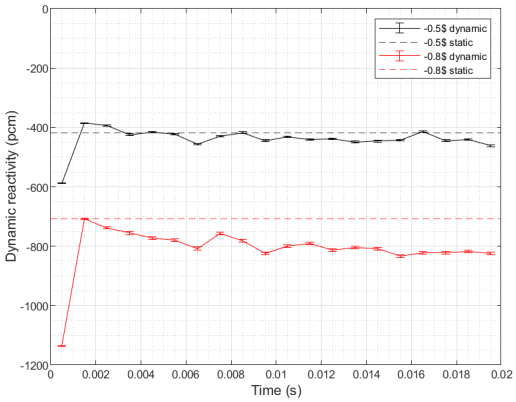
(a) *5$ insertion.*



(b) *1$ and 2$ insertions.*



(c) *0.5$ insertion.*



(d) *Negative reactivity insertions.*

Figure 17: *Dynamic evolution of the reactivity in the thermal slab model.*

As the insertions introduced are steps, the expected behaviour of the reactivity is that it immediately increases to from the critical value ($\rho_0 = 0$) to its perturbed value and remain constant for the whole transient, and that is what can be seen in the obtained reactivities. Except for the 0.5$ case, all the reactivities need a small period of time to converge to their final value, but once this value is reach the plots show that it remains constant, with slight variations that respond to the stochastic nature of the simulations. At a first glance, it might look like the 5$ case is the one that takes the most time to converge, but the reason for that is that it is the case in which the time span simulated is shorter, only 5 ms due to the extremely fast neutron population increase. In fact, all the cases shown appear to need between 2 ms and 3 ms to converge to its final value, as can be observed in figures 17b and 17d. The reason for the 0.5$ case to not show this behaviour

is that it is the longest simulated case, reaching 200 ms, and the first time step is too long to capture the first values of the reactivity before it converges.

Reactivity is the most valuable parameter obtained, as it is also used in the comparison between the methodologies to calculate the static reactivity, so a deeper evaluation of the results obtained in the dynamic Monte Carlo simulation, together with the results of the calculations of the static reactivity, is done in section 5.1.4.

5.1.4 Static reactivity

The reference dynamic reactivities calculated in section 5.1.3 have been used as a best estimate result to benchmark the static reactivities obtained from static Monte Carlo calculations and from the perturbation theory approach. This section includes the results of the different methods with the thermal slab model, together with an assessment of their advantages and shortcomings.

The static reactivities calculated using equations 2.21 and 2.28 have been obtained using the deterministic code *Dragon*. The results obtained for the six perturbations studied are reported in table 18, together with the reference dynamic reactivities simulated with Serpent.

Table 18: *$\Delta\rho$ obtained for the studied local perturbations in the thermal slab model in pcm.*

Perturbation	**Dynamic $\Delta\rho$**	**Static $\Delta\rho$**		
		- $\Delta\lambda$	**1st order pert.**	**Exact pert.**
-0.8 $	-798 $\pm$ 5	-707 $\pm$ 11	-6916	-708.93
-0.5 $	-461 $\pm$ 3	-418 $\pm$ 10	-1303	-415.97
0.5 $	420 $\pm$ 7	408 $\pm$ 11	1010	385.83
1 $	858 $\pm$ 10	809 $\pm$ 10	2696	796.57
2 $	1682 $\pm$ 16	1645 $\pm$ 11	6105	1590.49
5 $	4374 $\pm$ 31	4092 $\pm$ 16	12731	3815.65

Several observations can be made from the results displayed in table 18. From the three methods used to calculate the static reactivity, the one that approximates better the reference dynamic reactivity is the $\Delta\lambda$ method, and not the perturbation theory approach, either the first order or exact. Even though it is closer to the reference result, the $\Delta\lambda$ method is still not able to closely approximate the real reactivity change in the system after the local perturbation has been introduced, especially for the larger perturbations. This serves to show the weaknesses of the k-eigenvalue simulations to solve eigenvalue problems of systems that are not critical, as the eigenvalues calculated with this method are accurate only for systems close to criticality. For this reason, the reactivity of the unperturbed system, which is critical, can be accurately obtained with a k-eigenvalue simulation, but not the reactivity of the perturbed systems. This

was already an expected outcome before performing the calculations, so these results confirm the initial hypothesis regarding the accuracy of the $\Delta\lambda$ method.

Another aspect of this method that the study wanted to assess is if, even though the results are not accurate when compared to the dynamic reactivity, they are still conservative. The results for the perturbations studied show two different behaviours of the method: on the one hand, when the perturbation induces a positive reactivity insertion, the static reactivities calculated appear to underestimate the actual reactivity variation in the system, as they are consistently lower than the reference reactivity, which is a clear weakness of the method if it is to be used in the safety assessment and the commissioning of a design. On the other hand, though, the results obtained for negative reactivity insertions are conservative in comparison with the reference reactivities, as they seem to overestimate it (the calculated static reactivity is not as negative as the reference reactivity). This is a positive aspect of the $\Delta\lambda$ method, since it would yield conservative results, as long as the considered perturbation turns the reactor subcritical, and at the same time the computational requirements to compute it are considerably lower than for dynamic simulations, which is a clear advantage of the static reactivity approximations.

The results obtained for the first order perturbation theory approach appear to fail to approximate the reference reactivity in all the cases studied. In general the value obtained with this method differs from the reference reactivity by a factor between 2 and 4. In the worst case, which is the -0.8$ insertion, the difference is even larger, and the static reactivity misses the reference reactivity by a factor of 8.6. In contrast to what has been observed in the $\Delta\lambda$ method, it appears that the perturbation theory method is very conservative for the positive reactivity insertions, which it tends to greatly overestimate, but this conservativeness is not maintained when the perturbations are negative.

With these results, it cannot be said the 1st order perturbation theory approach represents an appropriate method to approximate the dynamic evolution of the reactivity in a system, as the static reactivity calculations, although conservative in some cases, fail to approximate the results of the reference calculations. The appropriate question to ask when seeing these results is what can be the cause of the differences between the reference calculation and the first order perturbation theory approximation. The reason can be found in the fact that this approximation only uses the information relative to the unperturbed system state, and ignores the changes that might occur in the system after the perturbation has taken place. As stated in section 2.3.1, this can have a negative effect in the accuracy of the approximation when the perturbation induces a significant change in the spatial neutron flux distribution of the system, as is the case of the thermal slab model. This can be clearly seen in figures 15 and 16, where noticeably shifts can be appreciated, especially for the bigger perturbations, although even the 0.5$ and -0.5$ perturbations provoke smaller but still noticeable shifts.

This hypothesis is confirmed by looking at the results obtained with the exact perturbation method. In this case, the perturbation theory approach has been applied by using the fluxes of

the perturbed configurations of the system in static calculations. As can be seen, the solutions of the calculation are considerably closer to the reference results than the first order approximation, since the perturbed flux contains the information of the shift introduced in the spatial flux, which is especially important when the local perturbation is not quickly propagated throughout the system and has a localised effect, as happens in the thermal slab model.

The conclusion regarding the first order approximation method that can be reached with the results obtained is that it appears to be useful when a perturbation turns a system supercritical, as the approximations it produces yield conservative results when compared with a reference time-dependent Monte Carlo calculation. It has a major advantage, and it is that it only requires a static calculation of the unperturbed system to determine the required fluxes to introduce to equation 2.21, so the imprecisions associated with finding the solution of perturbed system can be avoided. Nevertheless, it appears that when the perturbations introduced to the system are local and the spatial flux distribution is shifted as a consequence of that, the first order approximation fails to approximate the value of the dynamic reactivity with enough precision.

From this conclusion, a new question arose: if the disagreement obtained with the first order approximation is due to the spatial shift of the flux, is the first order approximation more accurate if the perturbation is homogeneous along the whole system and not a local one, so that the spatial flux is not shifted?

5.1.5 Point kinetics equations

The static reactivities presented in section 5.1.4 have been used as input parameters in the point kinetics equations (see equations 2.35 and 2.36). The rest of the kinetic parameters introduced correspond to the critical configuration of the system, and can be found in table 1. In this way, the dynamic evolution of the population in the system has been obtained using the point kinetics approach and has been compared with the dynamic evolution obtained with the time-dependent Monte Carlo approach, the results of which are presented in 5.1.3. Figure 18 reports the solutions found which each of the methods. For the six configurations of the thermal slab system that have been studied, three dynamic populations are plotted. One is the solution of the time-dependent Monte Carlo approach, which displays the population obtained in a true dynamic calculation, while the other two are the result of solving the point kinetics equations using a static reactivity as the input parameter. The reactivities that have been used are the ones obtained with the $\Delta\rho$ method and exact perturbation method. The static reactivity that was obtained with the first order perturbation method, on the other hand, has been left out of the calculations because, although the results obtained are clearly conservative, they are significantly different from the other approaches in magnitude.

The results show a good agreement between the dynamic Monte Carlo and the point kinetics approaches in the beginning of the transient, although in all cases it can be seen that as the transient progresses the discrepancies between the methods tend to increase. In the configurations

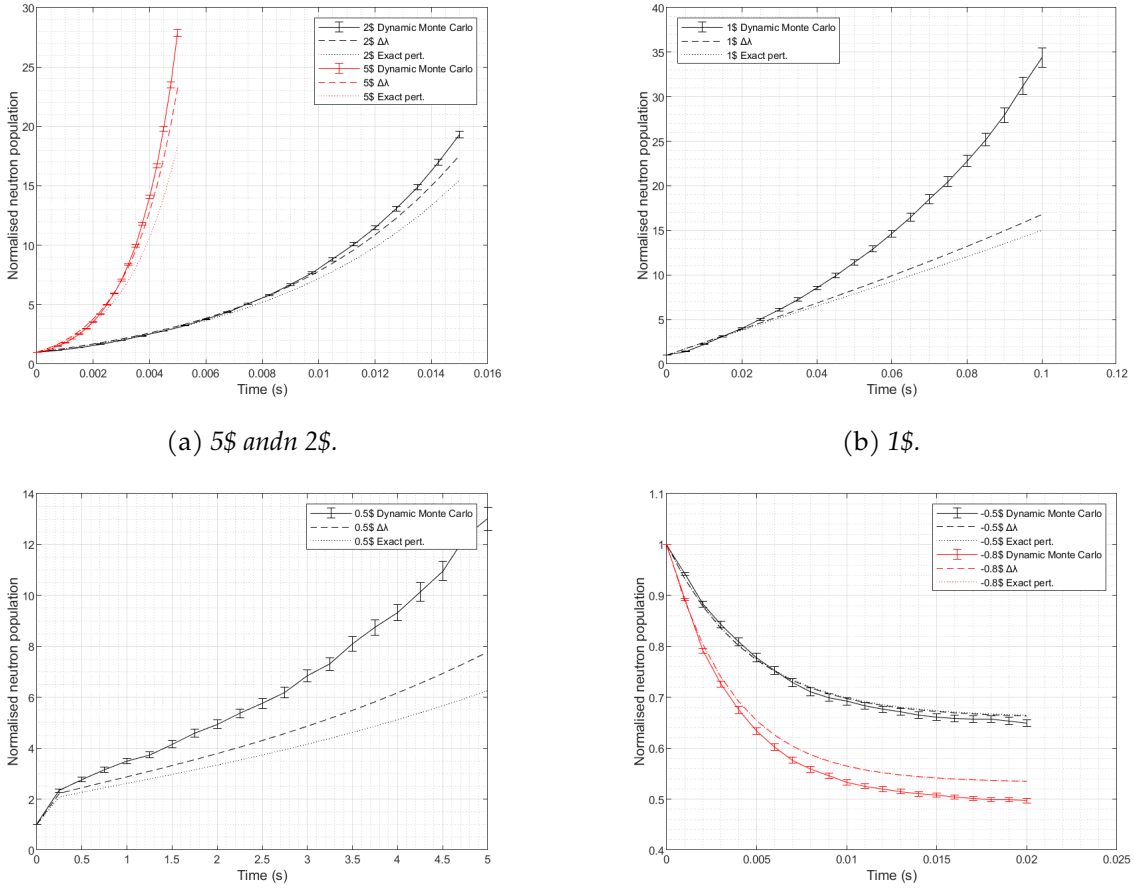
of 2$ and 5$ (figure 18a), that make the system prompt critical, it can be seen that the point kinetics approach captures correctly the exponential increase in the population that is observed in the Monte Carlo simulation, but nevertheless the progression of the dynamic approach is faster that for the point kinetics approach. At the end of the transients, the dynamic approach and the exact perturbation approach differ by a factor of 1.5 in the 5$ case, and a factor of 1.27 in the 2$, which is not an extremely high difference, although as time progresses its tendency is to increase. The case of the 1$ insertion yields interesting results, as the progression of the dynamic Monte Carlo solution is exponential, but the solutions of the point kinetics equations are linear instead. The reason is that the system becomes prompt critical at a reactivity of 814 pcm. The dynamic reactivity obtained in this configuration is of 858 pcm (see table 18) and so the population evolves exponentially, as in a prompt critical system, but the solutions of the static reactivities are lower, specifically of 809 pcm in the $\Delta\lambda$ approach and 796 pcm in the exact perturbation approach, which means that the case solved by the point kinetics equations is a non-prompt critical reactor. This case illustrates the importance of the approximations being conservative, because if a transient of this characteristics were to take place in an actual situation, the real evolution of the population (represented by the dynamic Monte Carlo simulation) would be significantly faster than expected.

The case of the 0.5$ insertion shows how the discrepancy between the approaches tends to increase for longer transients. In this case, the duration of the transient is of 5 seconds. While the initial prompt jump of the population is correctly captured by both point kinetics calculation, the increase that follows once the effect of the delayed neutrons begins is different for the three approaches, to the point that they differ by a factor of 2.2 at the end of the transient. The best agreement is found in the cases with negative reactivity insertions (figure 18d). After 20 ms the population seems to have stabilised in both cases and the discrepancies between methods are small, especially in the -0.5$ case.

In conclusion, it can be said that the point kinetics approach is able to correctly capture the neutron population evolution from a qualitative point of view in all cases except for the 1$ insertion, in which it has been seen that for the dynamic Monte Carlo approach the system is prompt critical, while for the point kinetics approach it is not, what results in different evolutions being obtained. As for the quantitative results, it appears that the point kinetics approximations yield conservative results only for negative reactivity insertions. For positive reactivity insertions, instead, the population of the time-dependent Monte Carlo approach is consistently larger. Finally, it is seen that the discrepancies tend to increase with time, so for longer transients the divergence is larger than for shorter ones.

5.1.6 Global perturbation

Three cases of global perturbations have been modelled: a 2$ insertion, which makes the system prompt critical, a 0.5$ insertion, and a -0.5$ insertion. The slab model has been modified so that all nine fuel regions have the same change in the composition for the perturbation to have a



(a) *5$ andn 2$.*

(b) *1$.*

(c) *0.5$.*

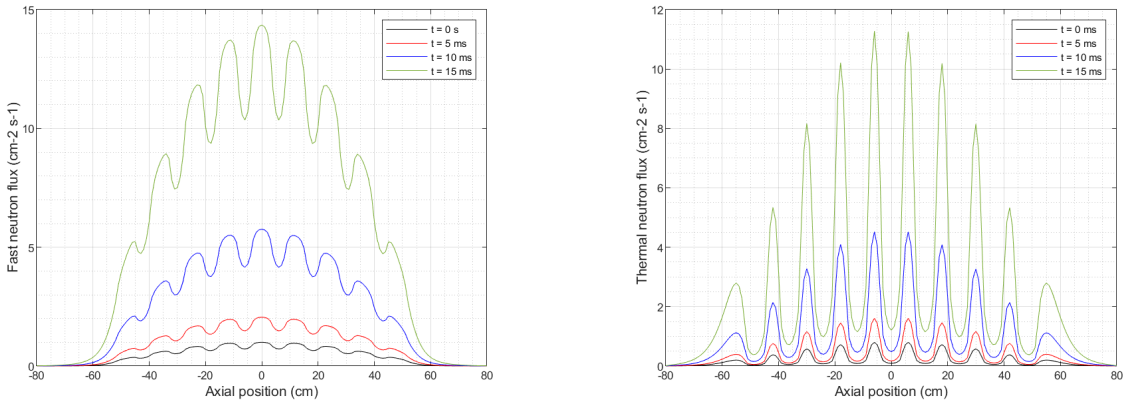
(d) *-0.5$ and -0.8$.*

Figure 18: *Dynamic evolution of the neutron population with the time-dependent Monte Carlo and point kinetics approaches.*

global effect. The enrichments of the fuel to reach the desired perturbations are 1.16% for the -0.5$ perturbation, 1.19% for the 0.5$ perturbation and 1.25$ for the 2$ perturbation.

With these parameters the dynamic evolution of the system has been simulated by means of time-dependent Monte Carlo simulations. The 2$ insertion has been simulated for a time of 15 ms, whereas the 0.5$ and -0.5$ insertions have been simulated for 2 s. The global perturbations introduced in the system have an effect in the flux spatial distribution, but only in its magnitude. As can be seen in figure 19, for the 2$ insertion the spatial flux of both the thermal and fast energy groups does not suffer any spatial shift. Instead, only the value of the flux changes, as it increases over time due to the reactivity being positive. The behaviour of the flux observed in the other two cases is the same.

The reactivity of each perturbation has been calculated using the same methods as in the local perturbations, so that four different values have been obtained for each case: the dynamic reactivity, obtained from the time-dependent Monte Carlo simulation, and three static reactivities



(a) *Fast flux distribution.* (b) *Thermal flux distribution.*

Figure 19: *Dynamic evolution of the spatial flux distribution for a 2$ global perturbation.*

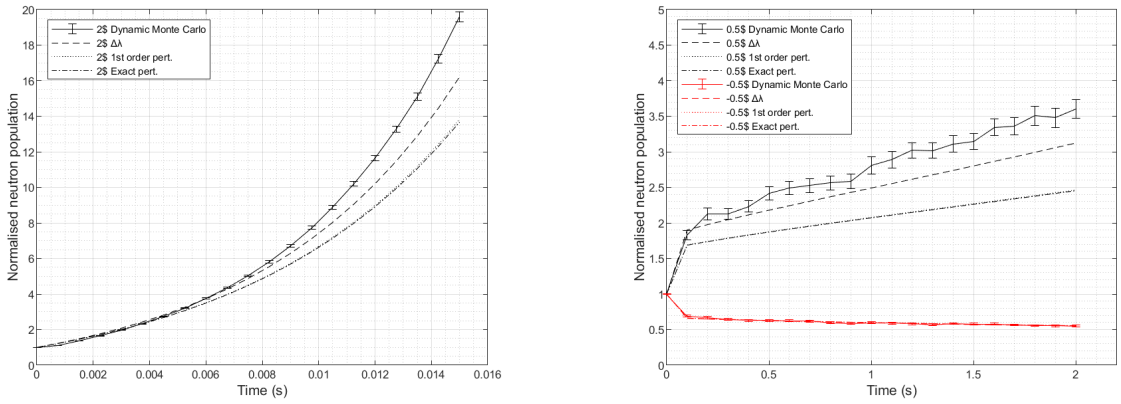
obtained with the $\Delta\lambda$ method, the first order perturbation method and the exact perturbation method. The results obtained are displayed in table 19. In it it can be seen that the three static reactivities have a good agreement between them, and that they all approximate the dynamic reactivity correctly. The best approximation has been obtained with the $\Delta\lambda$ method, which is calculated using Monte Carlo simulations. In comparison, the two static reactivities that are obtained with perturbation approaches show a slightly larger discrepancy with the dynamic reactivity. Since this results are obtained using the code *Dragon*, the disagreement can be attributed to the error introduced by the approximations that are necessary to solve the static problem using deterministic methods. In general, the first order approximation has a significant improvement since the spatial distribution of the flux is not shifted when the perturbation is introduced. Moreover, since the flux spatial distribution is the same for the unperturbed and perturbed systems, the first order perturbation approximation and the exact perturbation approximation are approximately equivalent. The differences observed are due to the fission operator **F** being modified. The first order approximation uses the unperturbed operator (F_0), while the exact approximation uses the perturbed operator $(\mathbf{F})$, which are slightly different since the perturbation is introduced by modifying the enrichment of the fuel.

Table 19: $\Delta\rho$ *(in pcm) obtained for the studied global perturbations in the thermal slab model.*

Perturbation	**Dynamic** $\Delta\rho$	**Static** $\Delta\rho$		
		- $\Delta\lambda$	**1st order pert.**	**Exact pert.**
-0.5 $	-415 $\pm$ 5	-404 $\pm$ 8	-397.99	-398.27
0.5 $	400 $\pm$ 9	372 $\pm$ 7	319.69	319.47
2 $	1652 $\pm$ 15	1610 $\pm$ 8	1539.32	1534.97

Following the same approach as in the local perturbations, the neutron population has been

tracked over time. The time-dependent Monte Carlo simulation has yielded a neutron population as a function of time, and the three static reactivities have been used as inputs of the point kinetics equations. The results can be found in figure 20. The 2$ has an exponential progression in all the approaches since the system is prompt critical. The discrepancy between the solutions is small in the beginning of the transient, but it increases as the time advances. At the end of the transient, after 15 ms, the dynamic Monte Carlo approach shows the largest population, which is approximately 20 times as high as the initial population. Both the 1st order and exact perturbation approaches have very similar progressions, as the static reactivities obtained are very similar. They have a disagreement of approximately 40% with the dynamic Monte Carlo solution. The $\Delta\rho$ method yields a better approximation with a disagreement of 25%. The behaviour of the 0.5$ case is very similar, with the three point kinetics solutions underestimating the population evolution of the dynamic Monte Carlo simulation. In this case the disagreement appears to remain constant after the initial prompt jump, being of 40% in the perturbation approaches and of 10% in the $\Delta\rho$ approach. The best agreement is found in the -0.5$ case, in which the three point kinetics calculations appear to capture the dynamic solution with good accuracy during the 2 second transient.



(a) *2$.* (b) *0.5$ and -0.5$.*

Figure 20: *Dynamic evolution of the neutron population for global perturbations in the thermal slab system.*

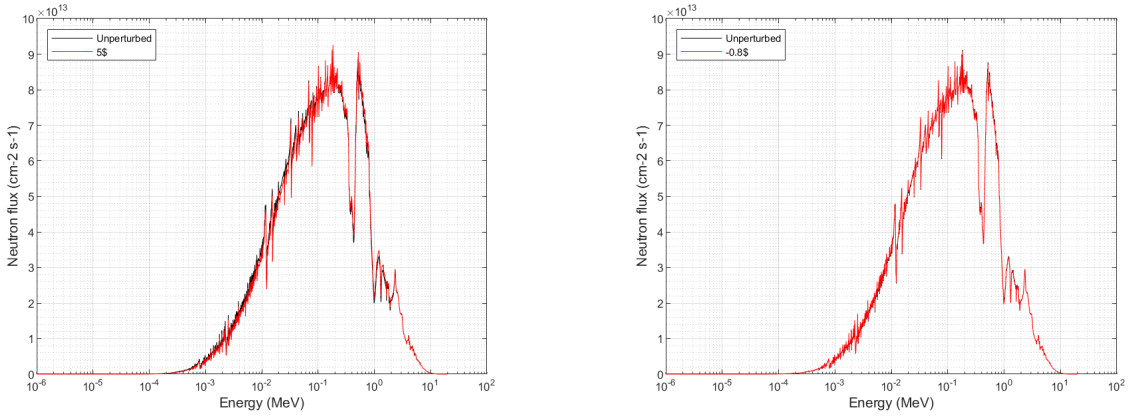
5.2 Fast slab model

5.2.1 Spectral shift in the flux

The behaviour of the fast spectrum slab is very similar to what has been observed in the thermal spectrum slab. The perturbations introduced do not significantly modify the spectral distribution of the flux with respect to the unperturbed case. The comparison between the unperturbed and the largest positive and negative perturbations studied is shown in figure 21. As can be seen in the figure, for the perturbation of -0.8$, there is no noticeable shift in the spectral flux distribution, whereas for the 5$ perturbation there is a very slight shift towards the higher energies of the plot, as the perturbed flux in the range between 10^{-3} MeV and 10^{-1} MeV tends

to decrease, and the flux in the range between 10^{-1} MeV and 10 MeV increases instead. Nevertheless, the shift is very small indeed and the peaks observed in the unperturbed flux are in the same energies and have the same magnitude as the peaks in the fluxes of all the perturbed cases.

The conclusion is that, in the same way that has been observed in the thermal slab model (see section 5.1.1, local perturbations do not have a significant effect on the spectral flux distribution of the model. The behaviour is therefore the same regardless of the energy spectrum present in the model. This lack of spectral shift means that the condition to apply perturbation theory for the calculation of the static reactivity is fulfilled, so the study can be performed in this model.



(a) *Positive reactivity insertion.* (b) *Negative reactivity insertion.*

Figure 21: *Spectral flux shift in the fast slab model.*

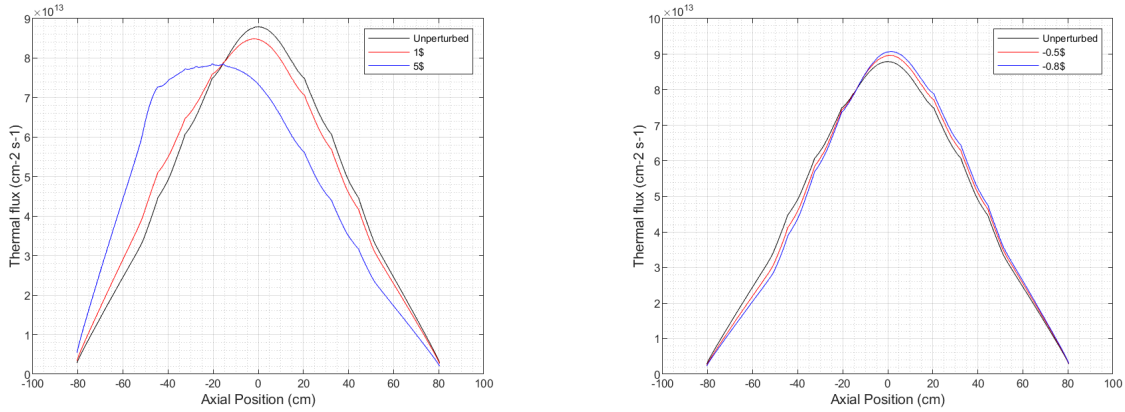
5.2.2 Spatial shift in the flux

The spatial distribution of the flux present in the fast slab model presents some noticeable differences with respect to what has been observed in the other slab model. To begin with, since there are no neutrons present in the thermal energy range there is no need to make a distinction between energy groups. Instead, all the neutrons have been grouped in a single energy group, so in the results presented in this section make no distinction between fast and thermal fluxes. The spatial flux distribution itself present differences between the thermal and fast spectra slabs. Whereas in the thermal slab the moderation in the water created a very noticeable distinction between the fuel and coolant regions, in which the neutrons in the fast energy group dominated the flux in the fuel regions and the ones in the thermal energy group dominated in the regions with coolant creating sharp peaks and valleys in the spatial distribution, in the fast slab model the lack of moderation in the coolant means that the spatial flux distribution is significantly smoother, presenting a near cosine shape with the flux being close to zero in the axial top and bottom limits of the slab and the maximum in the center in the unperturbed case (see figure 22, as the geometry and materials distribution of the materials make it symmetrical.

Another aspect of the results presented in figure 22 that is noticeable is the small shifts in the

spatial flux distributions when the local perturbations have been introduced in the fast slab system. While all the perturbations modify the spatial flux distribution of the system, the only perturbation that produces a significant shift is the 5$ perturbation. In the rest of the cases, the shift is not as important. For instance, for a 1$ perturbation a shift towards the negative axial positions is produced, and the peak of the flux is displaced to the left of figure 22a as a result of the increase in the fission rate in the bottom fuel region produced by the increase in ^{239}Pu concentration. Nevertheless, the shift does not modify the distribution significantly. In a similar way, the negative reactivity insertions (figure 22b generate a shift toward the right side of the plot, corresponding to the positive axial positions. Nevertheless, the shape remains very similar to the unperturbed case and the maximum flux remains approximately in the same value.

In contrast, the thermal slab system presented big shifts in the spatial distribution of the flux in both energy groups, but specially in the fast group, and as it has been seen in section 5.1.4, this shifts have resulted in the first order approximation of the static reactivity being very poor. This is a very interesting observation, since the small spatial shift in the flux means that the first order approximation of the perturbation theory approach might represent a good approximation to the real dynamic reactivity of the system. If this is the case for the fast slab model, there is a possibility that it can also be extrapolated to other fast spectrum reactors, such as MYRRHA, which means that the first order approximation can potentially be useful for the safety assessment of MYRRHA.



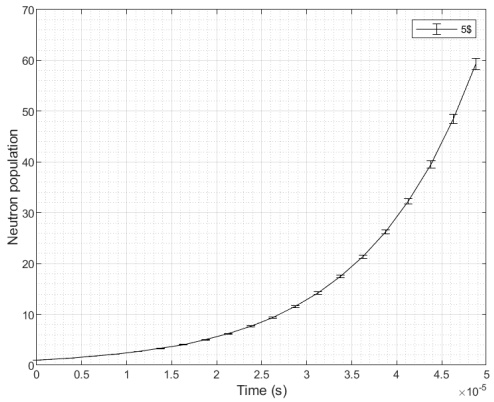
(a) *Positive reactivity insertion.* (b) *Negative reactivity insertion.*

Figure 22: *Spatial flux shifts in the fast slab model.*

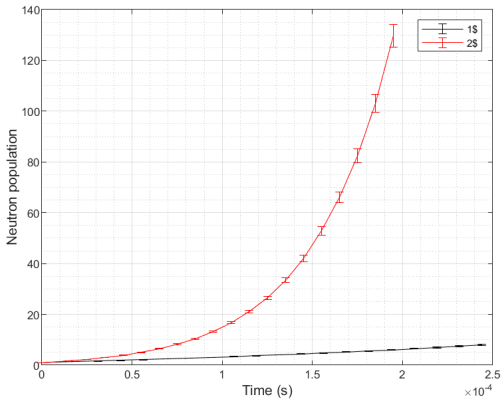
5.2.3 Dynamic evolution of the system

This section is structured in the same way as section 5.1.3, in which the results of the dynamic Monte Carlo simulations made with the thermal slab model are reported: first, a study of the simulated neutron population in the system over time is made. In it, the behaviour of the time-dependent evolution of the population is analysed, and the differences and similarities between the perturbations tested are commented. Following this, the spatial flux distribution evolution

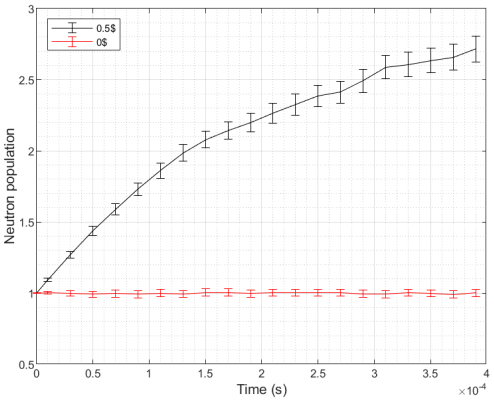
over the duration of the transient is presented, and finally the dynamic evolution of the reactivity in the system is reported, together with a comparison with the static reactivities that have been obtained in static k-eigenvalue simulations of the perturbed systems.



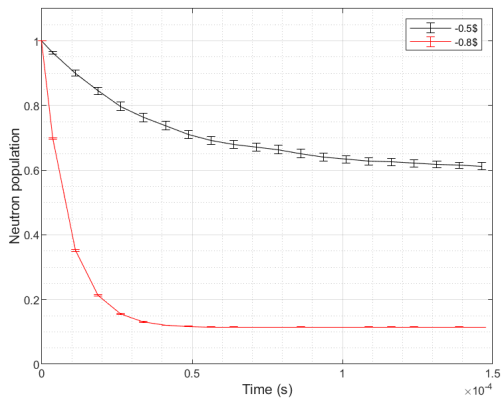
(a) *5$.*



(b) *1$ and 2$.*



(c) *0.5$ and 0$.*



(d) *-0.5$ and -0.8$.*

Figure 23: *Dynamic evolution of the population in the fast spectrum slab.*

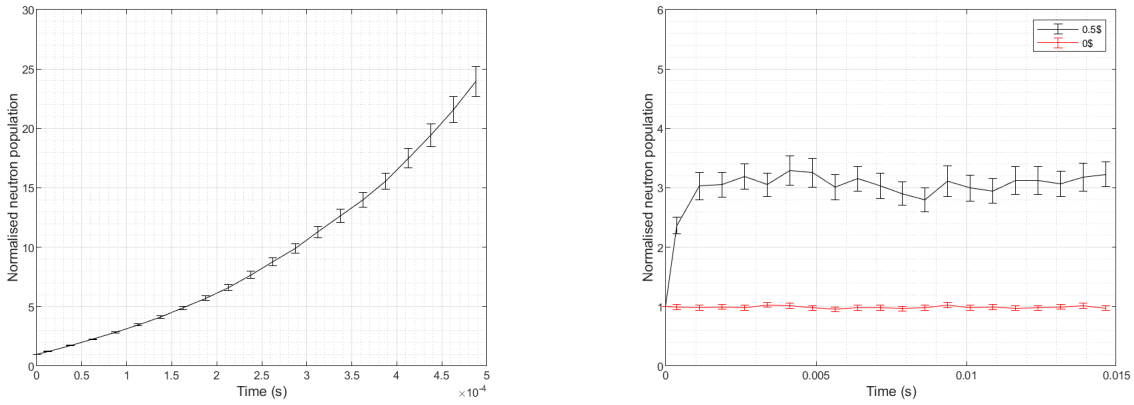
Neutron population. The low mean generation time of the fast slab system in comparison with the thermal spectrum slab indicated that the dynamic evolution of the fast system would be significantly faster, making it more difficult to control a possible reactivity excursion as the population of neutrons, and therefore the flux and the power output of the reactor, would increase much faster. This has been further confirmed in the study of the dynamic evolution of the system, as the time-dependent Monte Carlo simulations show that, indeed, the evolution of the system is much faster. While the thermal slab allowed to simulate prompt critical transients in the order of milliseconds, in the case of the fast slab, these same transients can only be simulated for time spans of the order of microseconds before the population increases to the point where the simulation fails.

The results of the dynamic neutron population are reported in figure 23, where the tallies in each time step are plotted with a 2-σ uncertainty. They have been grouped according to the simulated

time of each of the insertions, which have been selected with the objective of correctly seeing the population evolution. As expected, the fastest evolution corresponds to the 5$ insertion (figure 23a), as it is the insertion that increases the fission rate in the system the most. It makes the reactor prompt critical, and so the population increases exponentially from the beginning of the transient because the prompt neutrons born from fission are enough to maintain the fission chain. In just 50 μs the population has increased by a factor of 60 from its initial value, which is an extremely fast evolution. For comparison, in the thermal slab it took 5 ms for the population to increase by a factor of 25 (see figure 13a, which contains the evolution of the 5$ perturbation), so the increase it more than 100 times slower in the thermal slab than in the fast slab for an equivalent reactivity insertion.

Figure 23b contains the other two cases studied that make the reactor prompt critical, which are perturbations of 1$ and 2$. Although the system is still prompt critical, this insertions are significantly smaller than in the 5$ case, which make the dynamic evolution slower. This means that the transient can be run for a longer time. Indeed, the 1$ insertion has been run for a period of 250 μs and the 2$ insertion for 200 μs. In this time it can be clearly appreciated that the 2$ case has an exponential increase that brings the neutron population to increase by a factor of 130. It has been tried to simulate the transient for a longer time, specifically 250 μs, but it has not been possible as the simulation fails due to an overflow of the population. If we compare this evolution with the 5$ case, we can see that for a 2$ perturbation the time required to multiply the initial population by 60 is of approximately 170 μs, while in the case of the 5$ perturbation the time was of 50 μs. The evolution is therefore three times slower in the 5$ case. The evolution in the 1$ case is even slower, and after 250 μs it has barely reached an increase factor of 10. The time evolution appears to be linear which can be due to the fact that the simulated time is not large enough to appreciate the exponential shape that is seen in the other prompt critical cases, as has also happened in the thermal spectrum slab. To confirm it, a longer run has been performed with a time of 500 μs, which is shown in figure 24a. In it, the exponential evolution of the population can be more clearly seen, confirming that the insertion makes the reactor prompt critical, although its evolution is much slower than in the 2$ and 5$ cases, as not as many prompt neutrons are generated per unit time due to the lower fission rate in the system that results from the lower concentration of ^{239}Pu in the system.

Two cases have also been studied in which the reactor is not turned prompt critical: a null transient, in which the critical configuration is kept throughout the transient, and a 0.5$ insertion. The need for the delayed neutrons to be generated from the decay of precursors means that in this cases the dynamic evolution is slower than in the prompt critical cases, since the time scale of the decay of the precursors is larger than the one of the prompt neutrons. Figure 23c contains the results of these cases. In it, it can be seen how the population of the system in the 5$ case has an initial increase, the so-called prompt jump, caused by the increased fission rate in the system. Nevertheless, the prompt neutrons alone are not able to maintain the fission chain since the reactivity inserted is lower than the system's delayed neutron fraction (β_{eff}),



(a) *1$.* (b) *0.5$ and 0$.*

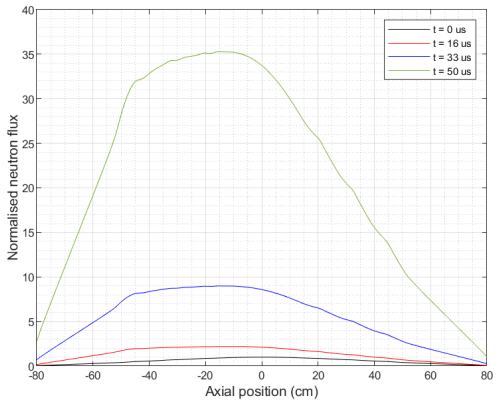
Figure 24: *Dynamic evolution of the population in the fast spectrum slab with extended simulated times.*

so the delayed neutrons are required to maintain the fission chain. Therefore, once the prompt neutrons that are initially created have induced their fissions, originating the prompt jump that takes place in the first instants of the transient, the increase in the neutron population is stopped and it stabilises. This full evolution cannot be fully seen in the transient of 400 μs that appears in figure 23c, so a second run was made with an increased time span, in this case of 15 ms with the objective of better seeing the dynamic evolution of the system. The new run can be seen in figure 24b, which shows how the population initially increases until it triplicates the initial value after approximately 1 ms, and it later remains constant until the end of the transient. Although faster, the general behaviour is the same as the one observed in the thermal spectrum slab (figure 13c), in which the population also stabilises after the initial prompt jump. The shorter time scale in the fast slab is due to the prompt neutrons not moderating before they cause fissions, which makes the prompt jump appear more quickly. In the fast slab it has also been seen that the population eventually increases after its stable period due to the decay of the precursors after approximately 200 to 300 ms. Since the precursors are are the same in both cases, the population in the fast slab should also begin to increase again after this period of time, but it has not been possible to run such a long transient due to time and availability constraints, as the running time of the simulation would have been too long, of around three weeks, and the cluster was required for other calculations. For these reasons it has not been possible to confirm the expected trend in the population, but nevertheless the results of the time length that has been simulated agree with the results of the same case in the thermal slab.

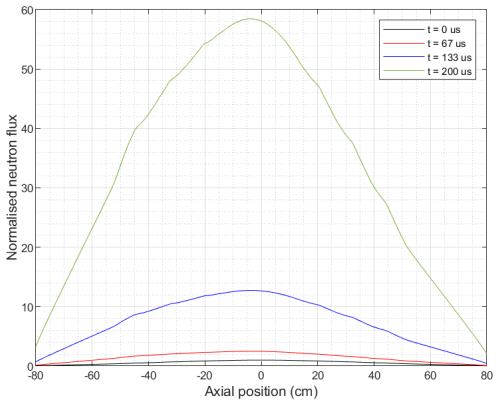
In the null transient, since the system remains critical for the transient, the neutron population is kept constant for the length of the transient, as was expected since the neutron production rate is equal to the loss rate due to absorptions, scattering and leakage. For further verification, a second run of 15 ms has also been made to confirm that the trend is maintained even for larger time spans, and the results, shown in figure 24b, are in the same line of the ones obtained in the run of figure 23c, which shows that the population remains stable.

Finally, in the two cases in which negative reactivity insertions have been introduced the neutron population tends to decrease (see figure 23d. In both cases the general trend is to have a sudden decrease in the population during the first microseconds of the transient, followed by a stabilisation of the population that is maintained for the rest of the transient. The initial prompt jump is due to the decreased fission rate that is induced in the system as a result of the lower enrichment of the fuel, and is analogous to the prompt jump observed in the 0.5$ case, but in the opposite direction: in this case, the decrease in the fission rate causes the number of prompt neutrons present in the system to decrease, as not as many of them are generated from fissions. This causes the population of neutrons to suffer a sudden decrease. Eventually, though, the precursors decay and the delayed neutrons are generated and induce the fissions necessary to stabilise the population in a new value, which remains constant. The magnitude of the perturbation that is introduced determines which is this new population value: in the case of the -0.5$ perturbation, the prompt jump is not as sharp as in the -0.8$ case, as the decrease in the fission rate is not as big. Therefore, the population decreases until it becomes 60% of the initial value, at around 150 μs and at this point it appears to stabilise. The evolution of the -0.8% case is faster, as the prompt jump rapidly decreases the population to a value of just 10% of the initial population, which is reached just after 50 μs from the beginning of the transient, and remains constant from this point on.

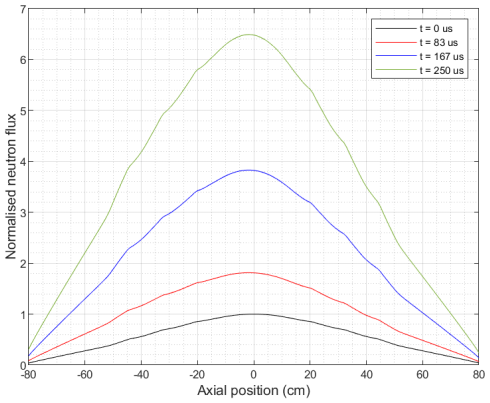
Spatial flux distribution. The spatial flux distribution of the fast slab model has only been obtained for one energy group, as there is no need to make a distinction between fast and thermal neutrons since there are none in the thermal energy range (see figure 21). The results for each of the perturbations induced can be seen in figure 25, which reports the dynamic evolution of the spatial flux distribution as a function of time. The fluxes are shown as relative fluxes, normalised to the maximum value of the unperturbed flux distribution.



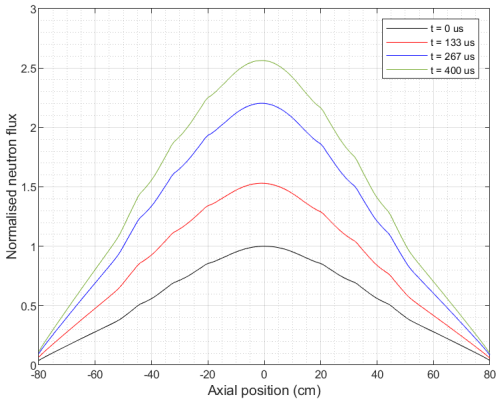
(a) *5$.*



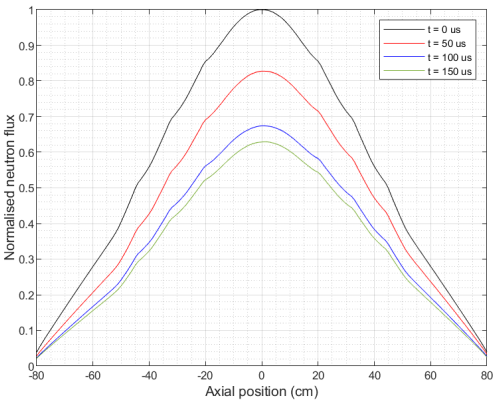
(b) *2$.*



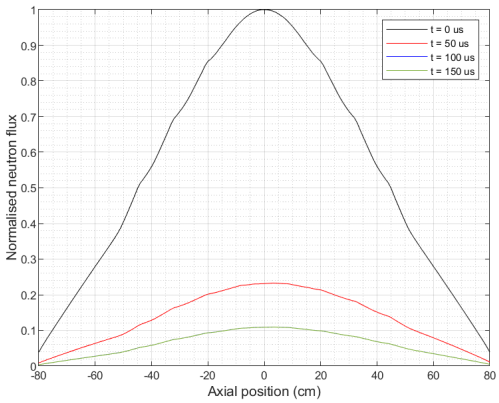
(c) *1$.*



(d) *0.5$.*



(e) *-0.5$.*



(f) *-0.8$.*

Figure 25: *Dynamic evolution of the spatial flux distribution in the fast slab system.*

In the 5$ and 2$ cases (figures 25a and 25b) it can be seen how the exponential evolution of the neutron population previously observed in figures 23a and 23b affects the magnitude of the neutron flux, as the increase in the magnitude is small in the first time step, and it becomes larger in the last, following the trend observed in the population. As for the shift in the flux,

it can be appreciated that for the 5$ case the spatial distribution of the flux has a noticeable change from time t = 0 s, in which the flux has the distribution of the unperturbed system, to the rest of the times displayed, which have the shape of the flux in the perturbed system. The shift is produced immediately after the beginning of the transient, as the introduction of the perturbation is modelled as an instantaneous step, instead of a more progressive way. The shift in the 2$ case is also produced, but it is not as noticeable as in the 5$ case since the perturbation that is introduced is not as big.

In the case of the 1$ insertion, it has been seen that when only 250 μs are simulated, the evolution of the population is approximately linear, which is further confirmed when the dynamic evolution of the flux in observed (figure 25c), since the increase in the magnitude of the flux is constant throughout the transient. As for the shift in the spatial distribution, although the perturbation is strong enough to turns the reactor prompt critical by just modifying the configuration of the system locally, it appears to propagate quick enough to have an almost homogeneous effect in the system, which results in a very slight shift towards the positive axial direction.

In the 0.5$ case, it has been seen that the population begins the transient with an approximately linear increase, which has a slope that eventually becomes smaller, although at the time point of 400 μs the population is still increasing. The evolution of the flux agrees with that progression, showing that the increase in the intervals between 0 - 133 μs and 133 - 267 μs are approximately constant, while the increase in the last time step between 267 - 400 μs is smaller, indicating that the population increases more slowly.

Finally, in the negative reactivity insertions (figures 25e and 25f) the evolution of the flux is contrary to the positive insertion cases, as the flux tends to decrease following the trend of the population. For the smaller perturbation, of -0.5$, the decrease in the magnitude of the neutron flux in the first and second time steps reflect the initial prompt jump, in which the population has an initial decrease, which is later stabilised thanks to the decay of precursors. Indeed, in the last time step, which ends at a time of 150 μs, the decrease in the flux is considerably smaller than in the previous time step. The prompt jump is more noticeable in the biggest perturbation, of -0.8$, since the prompt jump takes places faster, in the beginning of the transient, and then the neutron population stabilises in a value of 10% of the initial population. In the flux, the first time step has a big decrease in the magnitude of the flux, while in the second and third time steps the magnitude stabilises. At the end of the second time step, at 100 μs, the population has already reached its final value, so the flux remains constant in all the third time step. In both cases, similarly to the 1$ and 0.5$ perturbations, the spatial shift in the flux is small, which appears to indicate that the local perturbations are propagated quick enough throughout the system to have an almost homogeneous effect.

Reactivity. While, generally, a good agreement has been found in the thermal slab system between the dynamic reactivity calculated with time-dependent Monte Carlo simulations and the static reactivity calculated with k-eigenvalue simulations, in the case of the fast slab system

the results appear to be in disagreement. The results of the fast slab, displayed in figure 26, show that the reactivities obtained with the two methods have significant differences, especially for the biggest positive and negative perturbations, of 5$ and -0.8$.

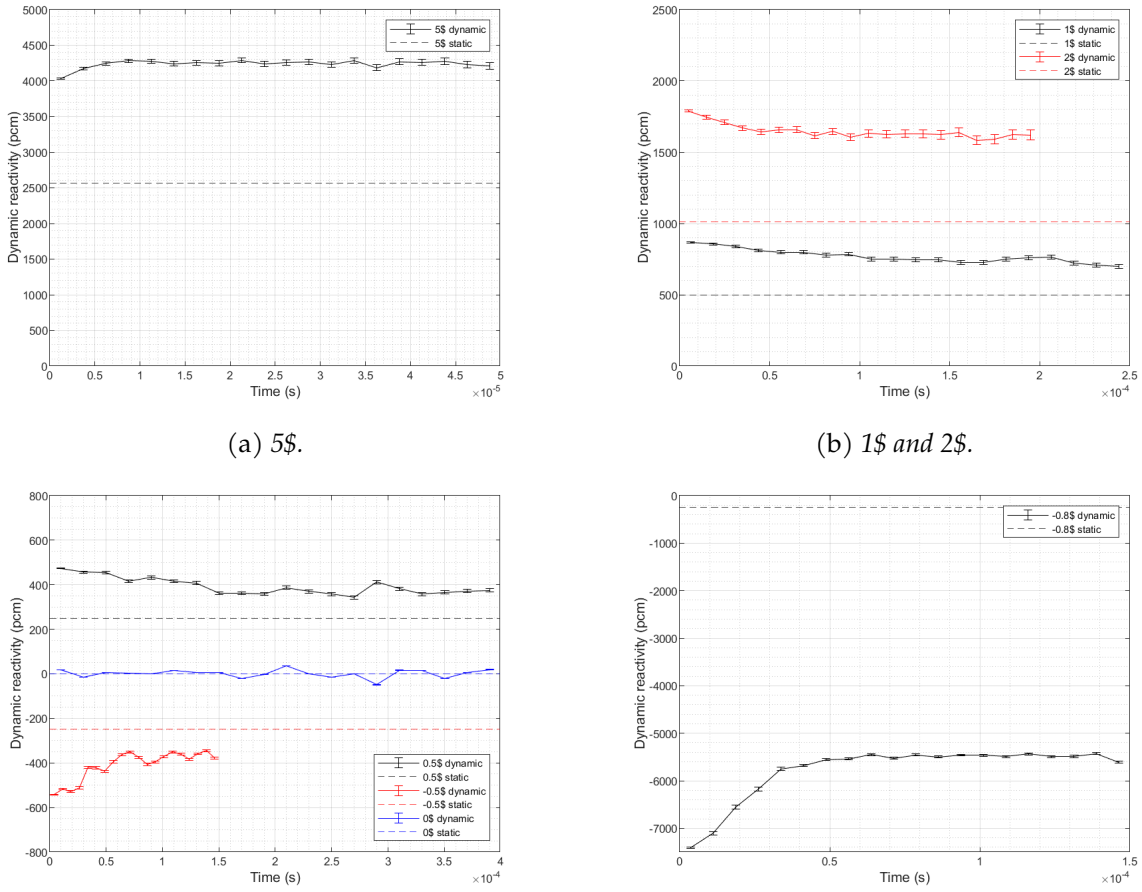
In the case of the 5$ perturbation (see figure 26a), it has been seen that the dynamic reactivity converges at a value of approximately 4200 pcm, whereas the static reactivity that has been obtained using a k-eigenvalue simulation of the perturbed system yielded a reactivity just below 2600 pcm. This difference cannot be attributed to statistical error, since the uncertainty of the static calculation is of 6 pcm, and the one of the dynamic reactivity, although higher, never surpasses 50 pcm, which is far less than the disagreement observed between the mean values obtained. It has to be said that this disagreement has also been observed in the thermal slab model, but the differences between static and dynamic reactivity are not as big as in the fast slab model.

The biggest disagreement can be found in the -0.8$ case, in which the dynamic reactivity converges at a value of -5500 pcm, which is far less than the static reactivity calculated with the same configuration of the fast slab in a k-eigenvalue static simulation, which has yielded a result of -439 pcm.

In the rest of the cases the disagreement between the dynamic reactivity and the static reactivity is not as big. For a perturbation of 2$, it has been found that the dynamic reactivity after a period of 200 μs is of 1600 pcm, whereas the static reactivity is of 1025 pcm, which represents a disagreement of 56%. In the 1$ case the reactivity stabilises at 700 pcm, whereas the static reactivity yields a value of 498 pcm, so the disagreement is of 40%.

The lowest disagreements (in absolute terms) in the perturbed cases have been found in the 0.5$ and -0.5$ perturbations, which correspond to the cases that are closest to criticality. The 0.5$ and -0.5$ have yielded symmetric results: for the positive 0.5$ case, the dynamic reactivity obtained is of 380 pcm, and the static reactivity is of 249 pcm, whereas in the negative -0.5$ case the reactivity values are the same but with opposite sign. Therefore, in both cases the relative disagreement between static and dynamic reactivity is of 52%, so it is approximately in the same level as in the 1$ and 2$ cases. In the case of the null transient, however, the dynamic and static reactivities show a very good agreement, as both results converge at approximately 0 pcm.

The observed trend of the dynamic reactivities in comparison with the static ones is the same that has been obtained previously in the thermal slab model: the static reactivity coming from a k-eigenvalue calculation tends to underestimate the reactivity when the perturbations introduce positive reactivity changes, as the static reactivity is consistently lower that the dynamic reactivity. On the other hand, when the perturbations introduce negative reactivity changes, the static reactivity tends to overestimate the actual reactivity change, represented by the dynamic reactivity coming from a time-dependent calculation. In this sense, the results of both the thermal and fast slab systems agree, pointing in the direction that approximating the ac-



(a) *5$.*

(b) *1$ and 2$.*

(c) *0.5$, -0.5$ and 0$.*

(d) *-0.8$.*

Figure 26: *Dynamic evolution of the reactivity in the fast slab system.*

tual reactivity change with a k-eigenvalue calculation yields conservative results in the case of negative reactivity insertions, and therefore is appropriate for safety applications, but when the reactivity insertions are positive the trend of the k-eigenvalue calculations is to underestimate the reactivity, which means a more conservative method should be found.

Although the observed qualitative behaviour of the dynamic reactivity of the fast slab model is consistent with the results obtained in the thermal slab model, the significant differences between the dynamic reactivity and the static reactivity are a cause of concern, especially in the case of -0.8$, in which the disagreement is significantly bigger than in the other cases. It appears that, in this perturbation, the neutron population decreases too much, so the number of histories that can be simulated in the transient is too low for the calculation to produce accurate results. It is possible that this numerical problem can be solved by increasing the amount of particles simulated per history, but it is not possible due to the limitations of the cluster used for the calculations. For this reason, the results obtained in the -0.8$ case will be ignored, as no meaningful conclusion can be reached and the problem cannot be solved given the computational tools available.

5.2.4 Static reactivity

Table 20: $\Delta\rho$ *(in pcm) obtained for the studied local perturbations in the fast slab model.*

Perturbation	**Dynamic $\Delta\rho$**	**Static $\Delta\rho$**		
		- $\Delta\lambda$	**1st order pert.**	**Exact pert.**
-0.8 $	-5613 $\pm$ 22	-413 $\pm$ 8	-866.81	-1174.69
-0.5 $	-378 $\pm$ 4	-250 $\pm$ 6	-676.15	-948.44
0.5 $	374 $\pm$ 7	247 $\pm$ 5	1001.38	948.44
1 $	698 $\pm$ 14	498 $\pm$ 8	1933.70	1797.18
2 $	1621 $\pm$ 34	1014 $\pm$ 8	4199.32	3857.51
5 $	4211 $\pm$ 48	2564 $\pm$ 10	9932.94	9201.60

The comparison of the results obtained for the fast slab using the different methods proposed are displayed in table 20. For each of the perturbations introduced, the result obtained in the dynamic Monte Carlo simulation corresponding to the reactivity and uncertainty of the last simulated time step is displayed in the second column. The third column contains the static reactivity calculated with equation 4.13 using the solution of the k-eigenvalue static Monte Carlo simulations. Finally, the fourth and fifth columns contain the static reactivity calculated using the first order and exact perturbation theory approaches, i.e., with equations 2.21 and 2.28, respectively.

As mentioned in section 5.2.3 when the dynamic reactivity is discussed, in the case of the fast spectrum slab there is a noticeable disagreement between the dynamic reactivity and the static reactivity calculated in Serpent with a k-eigenvalue simulation, which is in the third column of the table. The relative difference is in the range between 40% and 60% for the all cases, except for the -0.8$ perturbation in which the dynamic reactivity obtained is of -5613 pcm, while the k-eigenvalue static reactivity from Serpent is of -348 pcm. The conclusion reached for this case is that the result of the dynamic reactivity is not correct, which might be due to a lack of simulated histories resulting from the sudden decrease in neutron population, which could have caused the simulation to miscalculate the magnitude of the reactivity insertion. For the rest of the cases, though, the dynamic reactivity appears to behave in the expected manner, so it appears that the static reactivity tends to underestimate the dynamic reactivity for the positive insertions, while it overestimates it in the case of negative reactivity insertions. This behaviour agrees with the results of the thermal slab model, so it could be said from the results that the reactivity calculated with the k-eigenvalue method in a static Monte Carlo simulation is conservative with respect to the reference dynamic reactivity, but only in the case of perturbations that make the reactor subcritical.

The results of the calculations performed with *Dragon* show some interesting behaviour. The

first thing that must be said is that there is significant disagreement between the results obtained for the $\Delta\rho$ in *Serpent* and in *Dragon*. It appears that the $\Delta\rho$ obtained with the perturbation theory approaches, which are calculated with *Dragon* are always larger, in absolute terms, in comparison with the one obtained from *Serpent*, which is a somehow surprising result, as it has been previously seen that for the thermal slab model the two codes had a general good agreement in the calculation of this parameter (at least between the exact perturbation and the $\Delta\lambda$ approaches), so the source of this major disagreement in the fast slab model needs to be discussed. The reason lies in the fact that *Dragon* is a code that is optimised for thermal spectrum models. Therefore, the libraries that are available to download from the web page where the distribution of *Dragon* can be found [3] use multi-group energy structures that are prepared specially to be used in models that have a thermal energy spectrum, which is obviously not the case of the fast slab model. There are five possible energy meshes available in the web page, and are based on the XMAS and SHEM [61] energy group structures, both of them optimised to minimise the error coming from self shielding calculations in the thermal and epithermal energy ranges. As they are not optimised for fast spectra, when used in the fast slab model the errors that originate from the multi-group approximation made to solve the neutron transport problem lead to miscalculations of the static reactivity.

The hypothesis to explain the disagreement in the static reactivity of the *Dragon* calculations would still need to be proved, though. An approach that could be followed to do so would consist in generating libraries with an energy group structure appropriate for fast spectra, like the one present in the fast slab model. Candidate structures to be used could be the SCALE 245- or 302-group structure, which are optimised for sodium-cooled fast reactors with MOX fuels [62], or LANL 187-group or 80-group structures, for example. The new libraries should be generated by the user in the draglib little-endian format, which is the one used in *Dragon*, using a nuclear data processing code such as *NJOY* [63]. Recalculating the Δk_{eff} with this new library a new set of solutions would be obtained, which would confirm the validity of the hypothesis or not. Another approach would be to use a different code to do the calculation that is optimised for fast systems, such as *Grace* [64], which is a multi-group fast spectrum code that solves the neutron transport equation in slab geometries, such as the ones discussed in this part of the thesis. Nevertheless the modification of libraries, the use of NJOY or the use of another neutron transport code are not part of the scope of the project, and there is not enough time available to include them. Although interesting, this exercise could not be included in the thesis and has been left as future work.

Even though it appears that the calculations in *Dragon* fail to calculate correctly the $\Delta\rho$, there are some interesting conclusions that can be reached by looking at the static reactivity calculated with the first order perturbation theory approach. If the two results obtained with *Dragon* are compared, it can be appreciated that there is good agreement between the first order per-

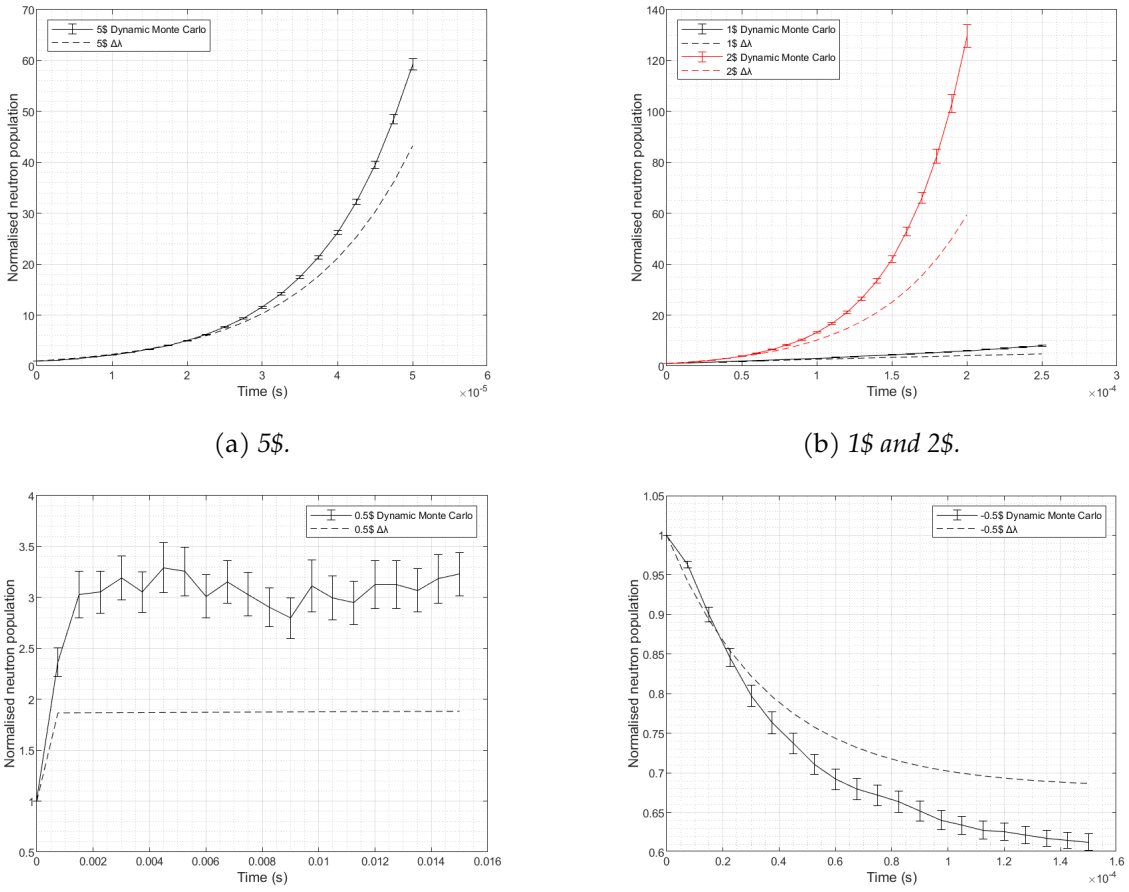
[3] http://merlin.polymtl.ca/

turbation and exact perturbation approaches (columns 4 and 5 of table 20, respectively). This agreement is specially good for the 0.5$ case, in which the difference between methods is of 53 pcm. The difference increases as the perturbations become bigger. In the case of the 1$ insertion it is still good, of 136 pcm, and for the 2$ case it is of 341 pcm. For the biggest perturbation, of 5$, the difference is of 713 pcm. In the negative reactivity insertions there are 272 pcm of difference for the -0.5$ case and 307 pcm for the -0.8$ case. When compared with the differences observed in the thermal slab model, it can be seen that they are significantly smaller in the fast slab. The reason can be found in the spatial shifts that the flux distributions have in each of the models. While the shifts in the thermal flux distribution are noticeable, even for the smallest perturbations (see figures 15 and 16), the shifts that appear in the fast slab for the same perturbations are much smaller (see figure 22). Since the perturbation approach to calculate the dynamic reactivity is dependant on the flux distribution that is introduced in equation 2.21, if the unperturbed flux distribution is used and the perturbation introduces a big shift in the distribution, then the first order approximation fails to approximate correctly the static reactivity. This is perfectly seen in the thermal slab, where there is a big shift in the flux that results in big disagreements in the static reactivity results, and also in the fast slab, where the smaller shifts mean that the first order approximation yields better results, especially in the positive reactivity insertions, in which the first order approximation appears to be more conservative than the Δk_{eff} approach currently used at SCK CEN.

5.2.5 Point kinetics equations

The results obtained with *Dragon* have shown a significant disagreement with the Monte Carlo solution, which has been attributed to the inappropriate energy mesh structures used for the calculations, as they are not designed for fast spectrum systems. For this reason, in the case of the fast slab the static reactivity calculated with the $\Delta\lambda$ obtained with *Dragon* has been left out of the calculations with the point kinetics equations. Figure 27 shows the neutron population evolution in the fast slab calculated with two approaches: dynamic Monte Carlo simulations and point kinetics, using the static reactivity obtained from Serpent static calculations. The -0.8$ case has not been included in the figure due to the major disagreement observed between dynamic and static results.

It can be seen for the figure that for most cases the solutions obtained with the two methods show a good agreement in the beginning of the transients, but as time advances the two solutions increasingly disagree. It can be clearly seen in figure 27b, where the solutions of the 2$ case begin with a good agreement in the first 50 μs, but after this point they diverge, with the solution of the dynamic Monte Carlo simulation increasing faster than the solution of the point kinetics equations. As both have an exponential increase, since the system is prompt critical, the difference between the solutions also increase exponentially. This same trend is also appreciated for the 5$ and 1$ cases, although the length of the transient calculated is not long enough to see the divergence with the same clarity as in the 2$ case. The cases that are not prompt crit-



(a) *5$.*

(b) *1$ and 2$.*

(c) *0.5$.*

(d) *-0.5$.*

Figure 27: *Dynamic evolution of the neutron population with the time-dependent Monte Carlo and point kinetics approaches.*

ical, which are the 0.5$ and -0.5$ insertions (figures 27c and 27d, respectively), have a different behaviour. In these cases, the populations have an initial increase followed by a stabilisation, which the point kinetics approach is able to capture. Nevertheless, the level at which the population stabilises in the point kinetics case differs from the solution of the time-dependent Monte Carlo solution. In the positive reactivity insertion the population is underestimated, while in the negative reactivity insertion it tends to be overestimated, which makes the solution found conservative.

In general, it can be said that the point kinetics equations correctly capture the trend in the dynamic evolution of the population, but there is a magnitude disagreement between the methods used, which can be attributed to the differences in the dynamic and static reactivities obtained. This disagreement can be interpreted as an example of the weaknesses of the k-eigenvalue method for the calculation of the reactivity in systems that are not critical.

5.3 Heterogeneous MYRRHA core model

This section contains a discussion of the dynamic Monte Carlo simulations that have been performed with Serpent in the heterogeneous model of the MYRRHA core. Since this is the most complex and realistic model among the ones that have been used in this thesis, the memory usage is the most intensive one. For this reason, together with the homogeneous MYRRHA model, it has been deemed as an appropriate model for a study of the capabilities of the Newton cluster present at SCK CEN to perform time-dependent calculations, as well as the capabilities, advantages and shortcomings of the Monte Carlo code itself.

With this objective in mind, two configurations of the reactor have been defined to perform the study: first, an unperturbed configuration in which the reactor is in a critical state. For that the control rod assemblies have been partially introduced in the active region of the core so that the reactivity reaches a value of $\rho = 0$, i.e., the reactor is critical. The second configuration used is one in which a local perturbation is introduced in the system. Starting from the critical configuration, the perturbation is introduced by completely removing one of the six control rod assemblies from the reactor. This introduces a local positive reactivity insertion in the system. A more detailed explanation of the two configurations, together with figures to better visualise it, can be found in section 3.2.2.

The following subsections contain the results obtained for the model. First, the variations induced by the local perturbation in the flux distribution, both spectral and spatial are reported. Then, the results of the study of the effects of the parameters used in the dynamic simulation are reported. This study consists in assessing the uncertainties obtained in the tallied dynamic reactivity and population of the system as a function of the number of cycles run in order to see if the uncertainty of the results saturates at a given number of cycles. The uncertainty values have also been assessed as a function of the number of time steps in which the transient is divided. The objective is to find an optimal configuration of the input parameters that minimises the uncertainty of the tallies. Finally, this configuration has been used to run a transient simulation of the perturbed system, the results of which are also reported.

5.3.1 Spectral shift in the flux

MYRRHA is a lead-cooled reactor, so the neutron flux spectrum present in the reactor is fast. This is reflected in the results obtained for the heterogeneous model in its critical configuration, which can be seen in figure 28. The figure shows the spectral flux normalised to its maximum value. It can be seen that the energy spectrum of the neutrons present in the core region is mainly in the range between 0.01 MeV and 10 MeV, which is the energy region in which fission neutron are born. Since there is practically no moderation in the reactor (except in the thermal IPS, where the flux shows a thermal spectrum), the neutrons in the reactor remain at the energy in which they are born, although some of them lose a part of it in their interactions before they disappear. The case of the thermal IPS is different, as it can be seen how a peak corresponding

to the thermal neutrons appears.

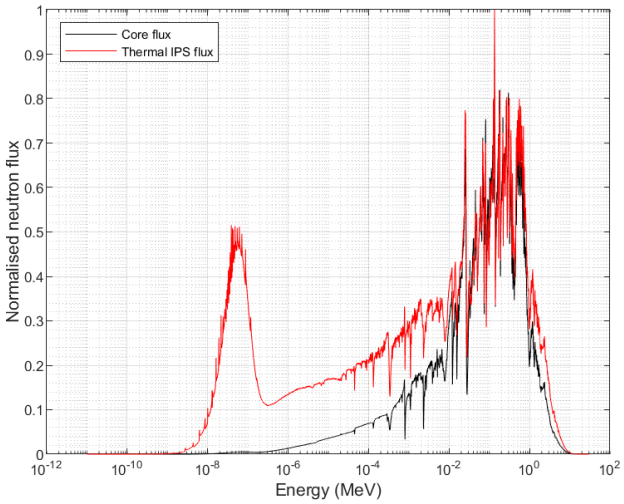


Figure 28: *Spectral flux distribution in the critical configuration of the hetoregenous model.*

Once the perturbation is introduced to the system by extracting one of the control rods from the active region of the core, it has been observed that the spectral distribution of the flux does not suffer any significant variation. It can be seen in figure 29, where the spectral distributions of the unperturbed and the perturbed systems are plotted together and no relevant shift in the spectrum can be seen. This behaviour was already expected, as it is the same that has been observed previously in the two slab models studied.

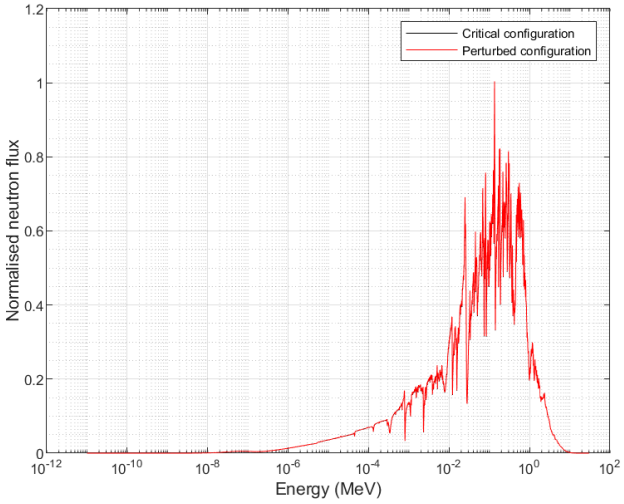
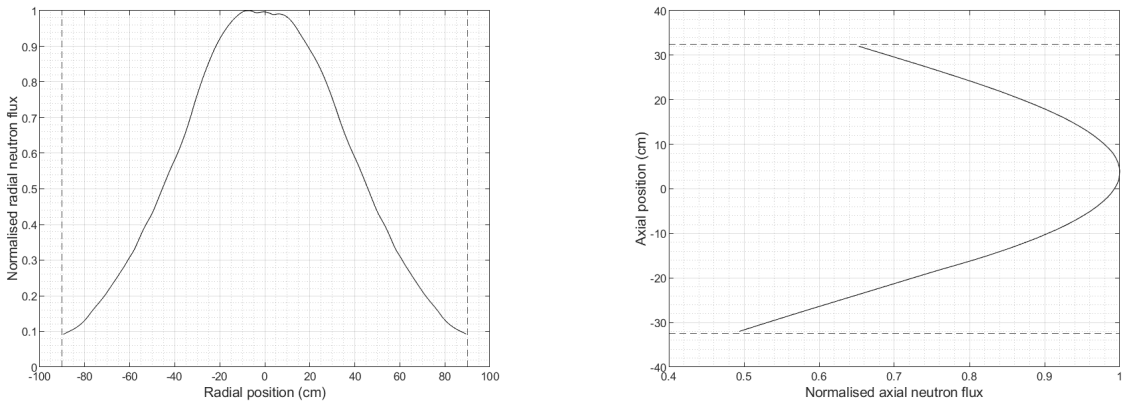


Figure 29: *Spectral flux distribution in the perturbed configuration of the hetoregenous model.*

5.3.2 Spatial shift in the flux

As it has been seen in section 5.3.1, the perturbation does not change the speactral distribution of the neutron flux significantly. Nevertheless, when the perturbation is introduced it is possible to see a spatial change in the flux distribution. Unlike the slab models previously studied, the MYRRHA model is 3-dimensional, in this case the shifts studied are not only in the axial direction, but also in the radial direction. For the critical case (figure 30), it can be seen that in the radial direction the flux follows an almost symmetric distribution, as the configuration in the axis that has been tallied is symmetrical. This is not the case of the spatial distribution in the axial direction along the active region of the core, between -32.5 cm and 32.5 cm, which shows a shift towards the upper region of the core. The reason for this shift is that in order to reach criticality it is necessary to introduce the control rods in the core. To do so, all six of the control rod assemblies are introduced a total of 15 cm inside the active region from the bottom (see figure 4a for a cross sectional view of the positions of the control rod assemblies in the critical configuration). Since the introduction of the control rods is homogeneous in the radial direction, the radial spatial distribution of the flux remains symmetrical, as has been observed in figure 30, but in the axial direction the absorption rate of neutrons is higher in the bottom region of the core than in the top region due to the presence of boron. As a result, the neutron flux in the bottom is lower than in the top, and the peak of the flux is shifted upwards, to a position of 4 cm.

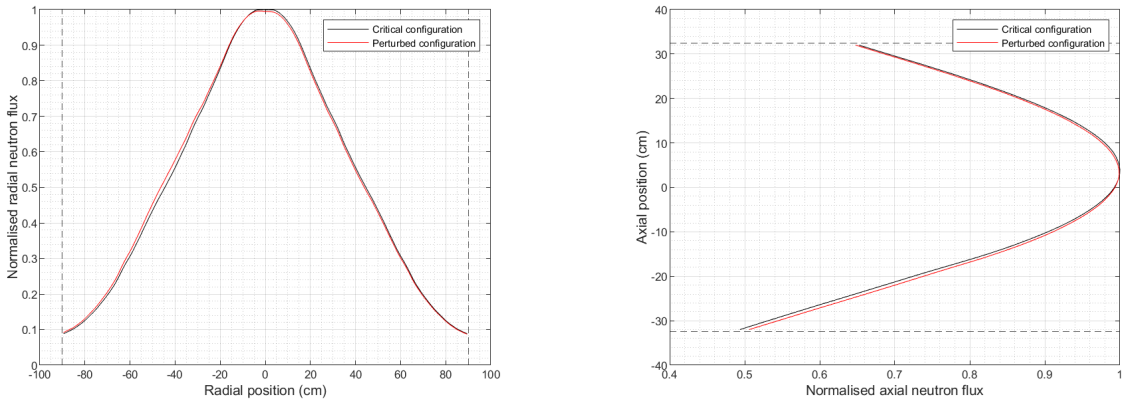


(a) *Radial flux distribution.* (b) *Axial flux distribution.*

Figure 30: *Spatial distributions of the neutron flux in the critical configuration of the heterogeneous model.*

When the perturbation is included, the behaviour of the system is quite similar to the behaviour observed in the fast slab model, as a spatial shift can be observed in the axial and radial distributions, but the local perturbation does not appear to have a significant effect, or at least it is not as pronounced as the ones observed for the thermal slab model. Nevertheless, it can be noted that the removal of the control rod assembly located at the position marked with a star in figure 2 has a slight effect on the spatial distribution of the flux. In the radial distribution it can be seen

that the flux in the negative Y axis (left region of figure 31a) the flux suffers a small increase due to the control rod in the region being removed, which reduces the capture reaction rate, although the effect is not strong since the capture cross section of the boron in the control rods is not very high in a fast energy spectrum. For the axial distribution the effect is similar. In this case, since the removed rod was inserted in the bottom region of the core, the perturbed system shows a slight increase in the flux in the region, also due to the reduced capture reaction rate from removing the control rod.



(a) *Radial flux distribution.* (b) *Axial flux distribution.*

Figure 31: *Spatial distributions of the neutron flux in the critical configuration of the heterogeneous model.*

The small shifts observed in the perturbed situation suggest that 1st order perturbation might be able to correctly approximate the reactivity change induced by the perturbation, as it is required that the spatial flux distribution of the system does not suffer a significant shift for the method to be accurate. Nevertheless, it has not been tested since it was not possible to obtain a reference dynamic reactivity from a time-dependent Monte Carlo calculation to compare with the results of the perturbation theory approach, and therefore it was decided that it was preferable to use the simpler slab models for that study. Instead, this model has been used for a study of the capabilities of the time-dependent Monte Carlo simulations with Serpent, which is reported in sections 5.3.3 5.3.4.

5.3.3 Optimisation of the dynamic simulations

It has been seen in section 5.3 that the usage of the Serpent model of the heterogeneous MYRRHA core has a high memory cost, since the geometry and the nuclear data of the materials present in the model require a significant fraction of the available memory in the nodes of the cluster. For this reason, it has been found that the number of histories per cycle that can be run is limited. While all the models used have a certain limit to that parameter, the limit found for this model is the lowest, of only 12000 particles per cycle. If one tries to increase this parameter, the simulation raises an error and fails, since the memory available for storing particle structures is

overflowed. Therefore, this parameter is fixed, and no more improvement of the uncertainty of the result can be done by increasing it.

Nevertheless, the fact that the number of particles per cycle is limited does not mean that the simulation cannot be optimised, as other parameters can be modified to improve the results of a given simulation. One of them is the number of cycles that can be run. Since the results are calculated as a mean of the tallied parameter in all cycles, if more cycles are run the result is closer to the real value and the uncertainty decreases. The second aspect that can improve the magnitude of the uncertainty is the number of time steps used in the simulation, as dividing the simulation in shorter periods of time effectively increases the number of histories that are run because the sample of particles to run takes place at the beginning of every time step. Therefore, if a simulation is set to have 12000 particles per cycle and the simulation is divided into 2 time steps, the total number of particles sampled is 24000.

The study has been divided into two parts: first, a study of the optimal number of cycles and secondly, a study of the optimal number of time steps to find the combination of parameters that minimises the uncertainty and the running time.

With the amount of 12000 particles per cycle that has been found to be the maximum allowed by the cluster, a series of simulations have been run using different amounts of cycles, and the uncertainties in the reactivity and the population of the system at the end of the transient simulation are measured have been displayed as a function of the cycles run. The simulations have been run in two different transients: a null transient, in which the reactor is kept in its unperturbed configuration, and a perturbed transient, in which one of the control rod assemblies is taken out of the reactor in a single step. In all cases the transient has been simulated for a total of 1 ms, divided into 3 time steps. The number of cycles run and the time required for the simulation to be completed in each case are displayed in figure 32. As can be seen, the running time is directly dependent on the number of cycles run, and it increases linearly with the amount of cycles. In the case of the null transient the linear increase has a constant slope in the whole range of cycles studied, whereas in the rod ejection transient the time appears to increase faster when more than 250 cycles are run, as the plot shows an increase in the slope of the running time. This might be explained by the fact that the duration of each individual cycle is different, as it depends on random events, so when more cycles are run the chances of having cycles take longer than usual increase, and that results in an increase of the overall simulation time.

Having seen that the time of the simulations increase approximately linearly with the number of cycles, the next step is to observe how the uncertainty of the results behave depending on the number of cycles that are run. For that, the dynamic reactivity of the system, as well as the dynamic population evolution, have been tallied in simulations that differ in the number of cycles run. These simulations have been carried out for the two configurations previously presented, the null transient and the rod ejection transient. The results obtained can be seen in figures 33a and 33b for the relative uncertainty in the dynamic reactivity and the neutron population,

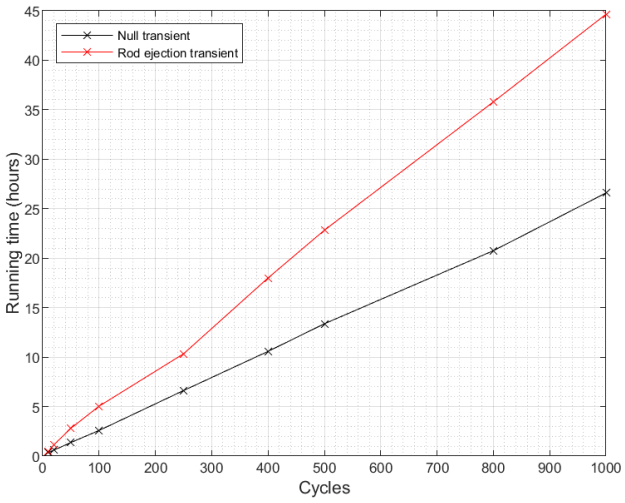
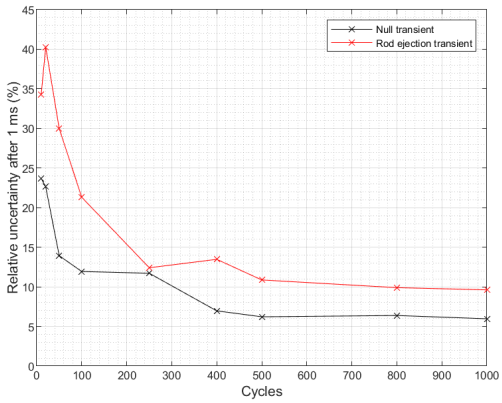


Figure 32: *Running time with respect to the number of cycles run in the MYRRHA model.*

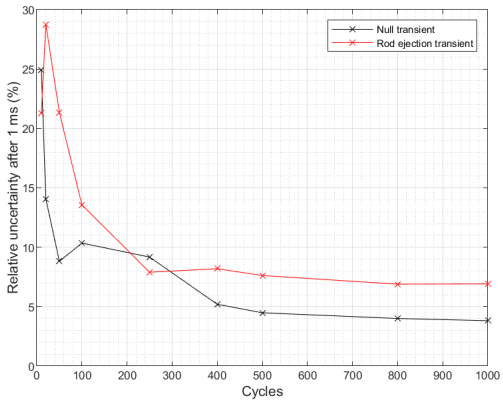
respectively. The results show how the relative uncertainty follow a similar trend in the two parameters tallied. For a low number of cycles the uncertainties have high values, in the range between 20% to 40% for both the null transient and the rod ejection transient, but as the cycles increase the uncertainty rapidly decreases until it eventually converges to a final value that remains constant regardless of how much the amount of cycles is increased. Both transients seem to converge in the range of 500 cycles, although apparently the rod ejection transient has already reached its final uncertainty values at 250 cycles, contrary to the null transient, which requires a larger number of cycles to reach its final value. Regarding the magnitude of the uncertainties, the null transient consistently shows lower values than the rod ejection transient, which is a reasonable result since in the null transient the system remains in its critical state, so no perturbation is introduced that can have an effect on the uncertainty. Another interesting observation that can be done is that the relative uncertainty in the reactivity is consistently higher than in the population. The reason is that the dynamic reactivity cannot be tallied directly in Serpent. Instead, it has to be indirectly calculated from the results of five other tallies that measure the rate of neutron production over losses (as in equation 4.7), so the uncertainty of the calculated reactivity combines the uncertainties of these five individual tallies, and therefore it is larger. On the other hand, the dynamic evolution of the population can be directly tallied in Serpent, yielding lower uncertainties.

The final uncertainties that have been reached in the dynamic reactivity are of 6% in the null transient and of 10% in the rod ejection transient. For the population, they are of 4% in the null transient and of 7% in the rod ejection transient.

The other approach that can be followed to optimise the simulation is to modify the number of time steps in which the transient is divided. To observe the effect that this parameter has in a simulation, a number of runs have been simulated with fixed parameters, the only difference



(a) *Dynamic reactivity.*



(b) *Neutron population.*

Figure 33: *Relative uncertainty in the tallied parameters in the MYRRHA model.*

among them being the number of time steps. For each of the two configurations available, the null transient and the rod ejection transient, a total of five simulations have been run with one to five time steps. The simulations have been run with 12000 particles per cycle, which is the maximum that has been possible to reach given the memory available computational power in the cluster, and 800 cycles, since the results obtained in figure 33b show that further increasing this parameter does not improve the final uncertainties of the calculation. The parameters tracked in the runs are the calculation time as a function of the amount of time steps, displayed in figure 34 and the final uncertainty of the reactivity and the neutron population, which appears in figure 35.

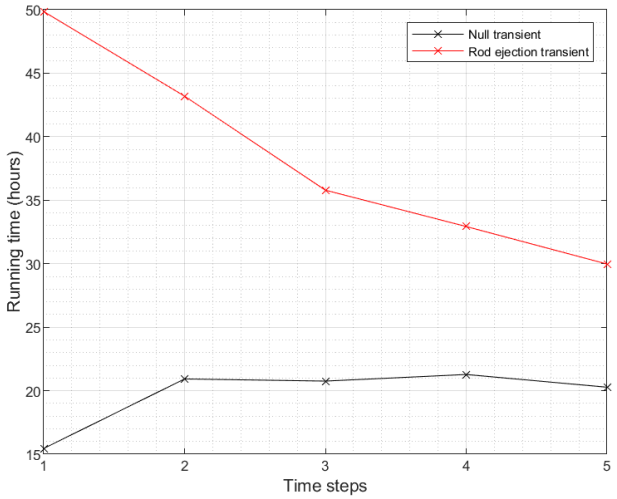
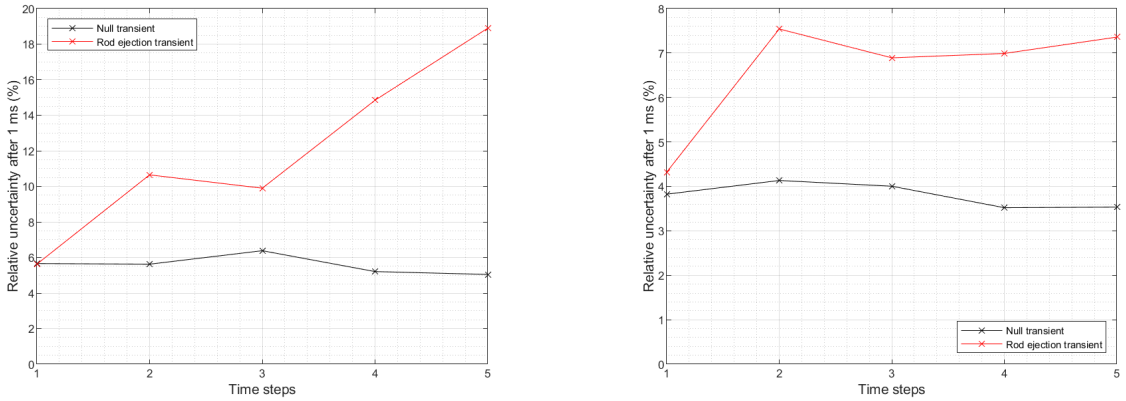


Figure 34: *Running time with respect to the number of time steps in the MYRRHA model.*

In the case of the null transient, the time required to run the simulation remains approximately constant regardless of the number of time steps defined, whereas in the rod ejection transient a

decrease in the time as the number of time steps is increased. The rod ejection transient has the reactor in a supercritical state, so the neutron population tends to increase with time. Therefore, at the end of a time step the population is larger than in the beginning, so the neutron population has to be reduced in the population control process. This population control has the effect of reducing the number of histories simulated, which means that as the amount of time steps is increased, the total simulated histories is reduced, so the time required to run the calculation is reduced. In the null transient, instead, the reactor is critical throughout the simulation, so the neutron population remains constant. Since there is no increase or decrease in the population, the population control that takes place in the interface between time steps has no effect in the simulation because the amount of particles simulated is the same. It can be seen that when there is no population control in the null transient case, i.e., only one time step is defined, the simulation time is slightly lower than in the rest of the cases, which can be due to the fact that the process of sampling live neutrons and forcing the decay of precursors (see section 2.2.3) is not performed.



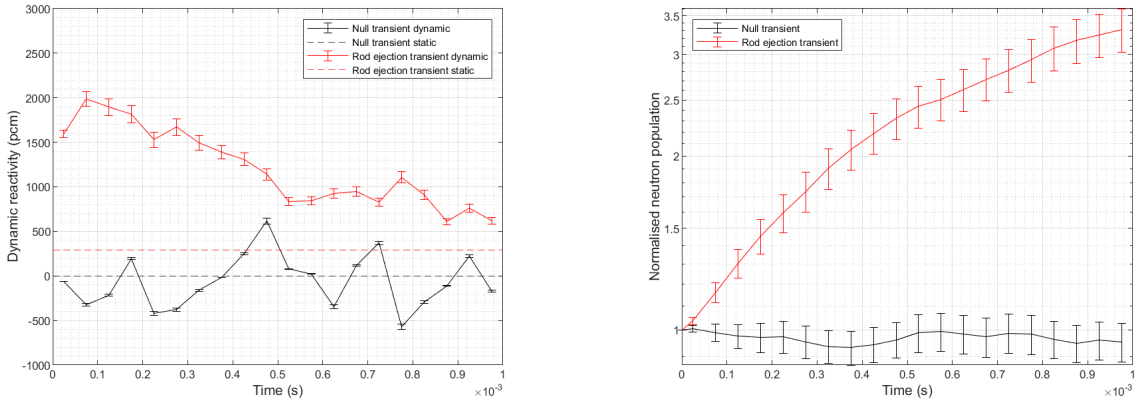
(a) *Dynamic reactivity.* (b) *Neutron population.*

Figure 35: *Relative uncertainty with respect to the number of time steps in the MYRRHA model.*

Figure 35 shows that in the null transient the uncertainty does not seem to be affected by the number of time steps defined in the simulation. The fact that the population remains constant during the transient means that the population control has no effect in the number of particles present in the system, so the histories simulated are the same regardless of the time steps used. Therefore, the uncertainty obtained is approximately the same in all cases, although some variability is observed due to the stochastic nature of the calculation. In contrast, for the rod ejection transient the uncertainty in both the population and the reactivity appears to increase as more time steps are included in the simulation. The effect of the population control can be seen, as it effectively decreases the number of histories simulated by killing part of the neutrons present in the system at the beginning of every time step. While it has a positive effect on the simulating time, which has been seen to decrease when more time steps are simulated, the effect on the uncertainty is negative since fewer histories are simulated.

5.3.4 Dynamic evolution of the system

From the results obtained in section 5.3.3 it has been decided to run the time-dependent simulations of the system by using 800 cycles and just one time step, as it has yielded the best results in terms of uncertainty. A transient of 1 ms has been has been run for the unperturbed configuration and the rod ejection configuration of the system, the results of which can be seen in this section.



(a) *Dynamic reactivity.* (b) *Neutron population.*

Figure 36: *Dynamic evolution of the reactivity and the neutron population in the MYRRHA model.*

Figure 36 shows the time-dependent evolution of the reactivity and the neutron population in the MYRRHA model for the null transient and the rod ejection transient. The dynamic reactivity obtained in figure 36a shows great variability. In the null transient, it oscillates around the value of 0 pcm that has been obtained in the static simulation of the system, showing values that range between 500 pcm and -500 pcm, which shows that, although the uncertainty in the null transient is reasonably low, the little amount of particles that can be simulated per cycle due to the memory available limits the precision of the result. In the rod ejection transient it can be seen how the reactivity at the beginning of the transient overestimates the static solution found by a large margin, reaching a value of 2000 pcm. As the transient advances, it appears that the static and dynamic solutions converge to the value of 300 pcm, although the dynamic solution is always higher than the static one, a trend that has already been observed in the two slab models.

The dynamic evolution of the population (figure 36b) shows that in the null transient the population remains close to its starting value for the whole duration of the transient. The values obtained are slightly lower than unity, but it can be attributed to the statistical uncertainty, as the 2-σ error bars contain the value of 1, corresponding to the initial population. In the rod ejection transient, it can be seen how the perturbation introduced produces an increase in the neutron population, which has the trend of decreasing its slope as the transient advances.

The results reached show that by optimising the simulation through the number of cycles and

time steps defined in the input file it is possible to reduce the the uncertainty of the tallied parameters. The reactivity has been calculated with a 6% uncertainty, while the neutron population has been obtained with uncertainties below 5%. Nevertheless, it is also clear that the low number of histories per cycle causes big oscillations in the calculated reactivity during the transient.

6 Timeline of the project

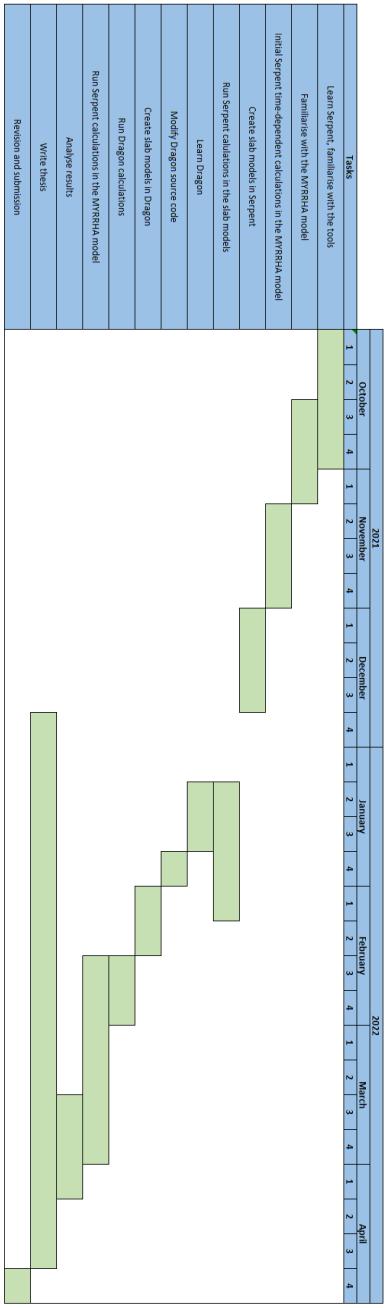


Figure 37: *Gantt diagram of the tasks performed during the project.*

Conclusions

This thesis had the goal of exploring the capabilities of the Monte Carlo code *Serpent* to perform time-dependent calculations, which has been implemented in the last few years. The results obtained were meant to be used as best-estimate solutions of the dynamic evolution of parameters such as the reactivity and the neutron population in a heterogeneous system, which would later be compared to calculations of these same parameters obtained from static approximations. The objective of such comparison was to assess the precision with which each of the static approaches can approximate the actual dynamic evolution of the system (obtained from the time-dependent Monte Carlo simulations), and how conservative the approximations are depending on aspects such as the type of insertion, e.g. if it is a homogeneous perturbation over the whole volume of the system or if it is a localised perturbation in a specific region, or the magnitude of the perturbation.

The original aim was to perform the study using the heterogeneous model of the MYRRHA core, as it would have served as a realistic case of study in a full 3-dimensional core and the results would have been directly applicable to the safety studies of the reactor being performed at SCK CEN. Nevertheless, it was found early in the project that the cluster that had to be used to perform the calculations did not have the computational power required to properly run transient simulations of the full core heterogeneous model due to its complexity in the geometry and materials used. While it was possible to run transients, the results did not achieve the desired uncertainty values, since the amount of histories simulated was limited by the available memory in the cluster. Seeing that, it was then decided that the study would be divided between two different types of model in order to achieve the desired objectives: on one side, a slab geometry with lower computational requirements would be used to perform the comparison between the dynamic evolution of the system and the static approaches to approximate it. Two slab models would be used, one with a thermal spectrum and another with a fast spectrum. On the other side, the MYRRHA model would be used to study the full capabilities of the available tools to perform time-dependent Monte Carlo simulations, i.e., understand how the input parameters affect the final results in terms of uncertainty and calculation time, in order to see the main weaknesses of the cluster currently used and the tools that would be required to solve them.

Even though the thesis was meant to be focused on the use of Monte Carlo codes, more specifically *Serpent*, the need to compute certain parameters that could not be obtained from this code, such as the adjoint neutron flux, which is necessary for the static reactivity using the perturbation theory approach, it was required to search for alternatives. This has resulted in a more diverse thesis, since a number of different codes have been used, not only Monte Carlo codes but also deterministic. *Dragon* and *Matlab* are good examples, as they appear in the body of the thesis, but other codes such as *TART* and *CORESIM* have also been used, although they have been discarded for different reasons.

Regarding the results obtained for the slab models, it has been seen that, generally, the dynamic behaviour of the systems was the expected, as the step perturbations introduced in the systems have an instantaneous effect in the reactivity, which converges around a final magnitude and remains approximately constant during the transient, showing some variability due to the stochastic nature of the calculation. The population also evolves as expected in a qualitative way, as it can be seen that it has an exponential progression when the system is made prompt critical, and the effect of the delayed neutrons is appreciated for non-prompt critical configurations. In the static reactivity calculations, the weaknesses of the $\Delta\lambda$ method to estimate the reactivity have been shown, as it fails when the system is brought away from criticality. When the perturbation makes the system supercritical, the method consistently underestimates the dynamic reactivity, whereas when the system is subcritical, it overestimates it. The conclusion reached regarding the $\Delta\lambda$ method is that for negative reactivity insertions it is conservative with comparison with the time-dependent calculations, but it is not conservative for positive reactivity insertions, so an alternative approach should be used.

The alternative approach proposed has consisted in the use of perturbation theory to derive an expression for the static reactivity. Two different expressions have been used, the 1st order perturbation equation, which relies on the information of the unperturbed system, and the "exact" perturbation equation, which uses the perturbed system's information. In the thermal system, it has been seen that the 1st order approximation yields conservative results for the positive reactivity insertions, although the large spatial shift observed in the flux distribution when local perturbations have been applied cause the approximation to fail. As a result, the static reactivity differs from the dynamic reactivity by a factor of 3, making the approximation useless. Nevertheless, when the perturbation introduced in the system is homogeneous, so that the flux of the perturbed system maintains the same shape as the unperturbed one, the solution of the first order approximation has been observed to be conservative with respect to the dynamic solution, showing that it is a good alternative as long as the shift in the flux is not large.

As for the exact perturbation results, they appear to better approximate the dynamic reactivity because the perturbed flux is used in the calculations, but it appears to have the same shortcomings as the $\Delta\lambda$ method, since it also underestimates the dynamic results. In the fast system, the results obtained for the perturbation approaches have been considered incorrect, since the code used for the calculation does not have multi-group energy structures optimised for fast spectrum systems, so a comparison with the dynamic results is pointless. Nevertheless, a useful conclusion that has been obtained is that, when the spatial shift in the flux is small, as is the case of the fast slab model, the solutions of the 1st order and exact approximations have a significantly better agreement, something that can potentially be extrapolated to MYRRHA.

The work performed using the MYRRHA model has been useful to define the main limitations of the cluster used to run simulations. By varying the values of the number of cycles run in the simulation and the amount of time steps in which the transient is divided it has been possible

to achieve uncertainties lower than 6% in the reactivity and 5% in the neutron population. The uncertainty has been found to converge to a given value as the number of time steps is increased, which is reached after approximately 500 cycles, although a final value of 800 cycles was chosen to ensure the convergence is reached. The time required to run the transient appears to scale linearly with the number of cycles. The conclusion reached regarding optimal number of time steps to be used in the simulations is that, when possible, it is better to not perform population control, i.e., use only one time step for the entire duration of the transient, as the uncertainty obtained is the lowest. It has been found that reducing the number of time steps has a negative effect in the calculation time, as it is higher. This behaviour has been observed only for the rod ejection transient, while in the case of the null transient it appears that the uncertainty and the running time is not affected by the time steps defined, due to the population control having no effect in the system. Nevertheless, the tallied reactivity and population have shown that reducing the uncertainty does not improve the variability of the tallied parameters, as it has been seen that, while the population has the expected behaviour, the reactivity shows great fluctuations during the transient, that result in disagreements with the solutions of the static simulations. The conclusion reached is that it is possible that the variability is due to the small amount of particles simulated in each cycle, and that increasing this number can lead to an improvement of the results. This hypothesis should have to be tested in detail first, though, something that has not been possible to do in this thesis due to the lack of a more powerful computer. For this reason, it remains as a work to be performed in the future.

The results presented in this thesis have pointed out some interesting conclusions, but they are part of a process that will be continued in the future. They have helped identify aspects that should be improved in order to reach more solid conclusions. It would be interesting to reproduce the results obtained in the time-dependent calculations with *Serpent* using other codes, such as *OpenMC*, in order to produce a benchmark and make sure that the current results are a correct solution of the system, which could also help find aspects of the *Serpent* implementation of the time-dependent calculations that need to be corrected by the developers. Additionally, the code *Dragon* could be modified so that an arbitrary flux can be introduced by the user to perform the perturbation theory calculations. In this way, it could be possible to play with the inputs of the perturbation theory equation in order to use it to obtain conservative solutions for the static reactivity that could be used in the safety assessments of new designs of reactors. For the time-dependent simulations themselves, using a more powerful computer would allow to improve the results of the heterogeneous MYRRHA model, as having more memory available would allow to simulate a larger amount of histories, reducing the uncertainty of the results and the calculation time. Therefore, it would be interesting for SCK CEN researchers to have access to supercomputers like the one present in the Flemish Supercomputing Center to progress further in this line of research.

Acknowledgements

Firstly, I would like to express my deepest gratitude to Augusto and Pablo for being my mentors. Without their help and support it would not have been possible to carry out this project. Even at difficult times, when problems appeared and the project had to be modified, their dedication and commitment kept me motivated to continue the work. I leave convinced that I have met not only great scientists, but also friends.

I would also like to extend my gratitude to the entirety of the ANS group and SCK CEN as a whole, for taking me as an intern and allowing me to participate in a project as exciting as MYRRHA.

I wish to thank my supervisor, Prof. de Blas, for accepting to direct this thesis from Barcelona and playing an important role in my beginnings in the nuclear world. His teachings and support have been very helpful not only in this thesis, but also during the duration of the Master and even the Bachelor.

Last, but not least, I would like to wholeheartedly thank my parents and my entire family for helping and supporting me in all my decisions, and providing me with an education that has brought me where I am today.

References

[1] Anurag Gupta and Suhail Ahmad Khan. Methods of solving neutron transport equation. In *Physics of Nuclear Reactors*, pages 263–329. Elsevier, 2021.

[2] Sedat Goluoglu. *A deterministic method for transient, three-dimensional neutron transport*. The University of Tennessee, 1997.

[3] Paul K Romano, Nicholas E Horelik, Bryan R Herman, Adam G Nelson, Benoit Forget, and Kord Smith. Openmc: A state-of-the-art monte carlo code for research and development. *Annals of Nuclear Energy*, 82:90–97, 2015.

[4] Jaime Romero-Barrientos. Time-dependent monte carlo in fissile systems with beta-delayed neutron precursors. *arXiv preprint arXiv:2101.09338*, 2021.

[5] Antonios G Mylonakis, M Varvayanni, DGE Grigoriadis, and N Catsaros. Developing and investigating a pure monte-carlo module for transient neutron transport analysis. *Annals of Nuclear Energy*, 104:103–112, 2017.

[6] Jaakko Leppänen et al. Development of a dynamic simulation mode in serpent 2 monte carlo code. *Proceedings of M&C*, pages 5–9, 2013.

[7] Emil Fridman and Xingkai Huo. Dynamic simulation of the cefr control rod drop experiments with the monte carlo code serpent. *Annals of Nuclear Energy*, 148:107707, 2020.

[8] Ville Valtavirta, Mohammad Hessan, Jaakko Leppänen, and Hans-Josef Allelein. Time-dependent modelling of the jsi triga with serpent 2. *NENE-2018. Portoroz, Slovenia*, 2018.

[9] JB Yasinsky and AF Henry. Some numerical experiments concerning space-time reactor kinetics behavior. *Nuclear Science and Engineering*, 22(2):171–181, 1965.

[10] JB Yasinsky. On the use of point kinetics for the analysis of rod-ejection accidents. *Nuclear Science and Engineering*, 39(2):241–256, 1970.

[11] Jaakko Leppänen et al. Serpent–a continuous-energy monte carlo reactor physics burnup calculation code. *VTT Technical Research Centre of Finland*, 4, 2013.

[12] Jaakko Leppänen. On the use of the continuous-energy monte carlo method for lattice physics applications. 2009.

[13] Hamid Aït Abderrahim, Peter Baeten, Didier De Bruyn, Jan Heyse, Paul Schuurmans, and Jan Wagemans. Myrrha, a multipurpose hybrid research reactor for high-end applications. *Nuclear Physics News*, 20(1):24–28, 2010.

[14] Didier De Bruyn, Hamid Aït Abderrahim, Peter Baeten, and Carmen Angulo Pérez. The

belgian myrrha ads programme. part 1: The new phased implementation plan. In *2018 International Congress on Advances in Nuclear Power Plants (ICAPP 2018)*, pages 1066–1073, April 2018. Score=3; 2018 International Congress on Advances in Nuclear Power Plants ; Conference date: 08-04-2018 Through 11-04-2018.

[15] Hervé Nifenecker, Olivier Meplan, and Sylvain David. *Accelerator driven subcritical reactors*. CRC Press, 2003.

[16] Hamid Aït Abderrahim, Didier De Bruyn, Gert Van den Eynde, and Rafaël Fernandez. Accelerator driven subcritical systems. In Ehud Greenspan, editor, *Encyclopedia of Nuclear Energy*, pages 191–202. Elsevier, Oxford, 2021.

[17] Igor Pioro. *Handbook of generation IV nuclear reactors*. Woodhead Publishing, 2016.

[18] Gert Van den Eynde, Edouard Malambu, Alexey Stankovskiy, Rafaël Fernandez, and Peter Baeten. An updated core design for the multi-purpose irradiation facility myrrha. *Journal of Nuclear science and Technology*, 52(7-8):1053–1057, 2015.

[19] Hamid Aït Abderrahim, Didier De Bruyn, and Michel Giot. From myrrha to xt-ads. development of pb-bi cooled ads and perspective of implementation in europe. 2007.

[20] CSK CEN. https://www.myrrha.be/about-myrrha, 2022.

[21] Dan Gabriel Cacuci. *Handbook of Nuclear Engineering: Vol. 1: Nuclear Engineering Fundamentals*, volume 1. Springer Science & Business Media, 2010.

[22] Alireza Haghighat. *Monte Carlo methods for particle transport*. Crc Press, 2020.

[23] Vito Vitali, Florent Chevallier, Alexis Jinaphanh, Andrea Zoia, and Patrick Blaise. Comparison of direct and adjoint k and α-eigenfunctions. *Journal of Nuclear Engineering*, 2(2):132–151, 2021.

[24] Dermott E Cullen, Christopher J Clouse, Richard Procassini, and Robert C Little. Static and dynamic criticality: are they different? Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2003.

[25] Colin James Josey. General improvements to the mcnp alpha-eigenvalue solver. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2018.

[26] Benjamin R Betzler, Brian C Kiedrowski, Forrest B Brown, and William R Martin. Calculating alpha eigenvalues in a continuous-energy infinite medium with monte carlo. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2012.

[27] Karl Otto Ott and Robert J Neuhold. *Introductory nuclear reactor dynamics*. American nuclear

society La Grange Park, Illinois, 1985.

[28] Lei Zhu et al. *Discrete generalized multigroup theory and applications*. PhD thesis, Massachusetts Institute of Technology, 2012.

[29] Edward Larsen. The suppression of energy discretization errors in multigroup transport calculations. Technical report, Univ. of Michigan, 2013.

[30] Christophe Demaziere. Core sim: A multi-purpose neutronic tool for research and education. *Annals of Nuclear Energy*, 38(12):2698–2718, 2011.

[31] Christophe Demazière, Victor Dykin, Augusto Hernández-Solís, and Viktor Boman. Development of three-dimensional capabilities for modelling stationary fluctuations in nuclear reactor cores. *Annals of Nuclear Energy*, 84:19–30, 2015.

[32] Weston M Stacey. *Nuclear reactor physics*. John Wiley & Sons, 2018.

[33] H Brockmann. Improved treatment of two-dimensional neutral particle transport through voids within the discrete ordinates method by use of generalized view factors. Technical report, Forschungszentrum Jülich GmbH, 1992.

[34] JY Moller and JJ Lautard. Minaret, a deterministic neutron transport solver for nuclear core calculations. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2011.

[35] Guy Marleau, Alain Hébert, and Robert Roy. A user guide for dragon version 4. *Institute of Genius Nuclear, Department of Genius Mechanical, School Polytechnic of Montreal*, 2011.

[36] Bart L Sjenitzer and J Eduard Hoogenboom. Dynamic monte carlo method for nuclear reactor kinetics calculations. *Nuclear Science and Engineering*, 175(1):94–107, 2013.

[37] Bart L Sjenitzer, J Eduard Hoogenboom, et al. A monte carlo method for calculation of the dynamic behaviour of nuclear reactors. *Progress in Nuclear Science and Technology*, 2:716–721, 2011.

[38] Ville Valtavirta, Mohammad Hessan, and Jaakko Leppänen. Delayed neutron emission model for time dependent simulations with the serpent 2 monte carlo code- first results. In *Proceedings of PHYSOR*, pages 1–5, 2016.

[39] Bart L Sjenitzer and J Eduard Hoogenboom. Variance reduction for fixed-source monte carlo calculations in multiplying systems by improving chain-length statistics. *Annals of Nuclear Energy*, 38(10):2195–2203, 2011.

[40] Steven C Van der Marck and R Klein Meulekamp. Calculating the effective delayed neu-

tron fraction using monte carlo techniques. In *Proc. Int. Conf. PHYSOR 2004*, pages 25–29, 2004.

[41] MJH Khan, MM Sarker, and SMA Islam. Analysis of kinetic parameters of 3 mw triga mark-ii research reactor using the srac2006 code system. *Annals of Nuclear Energy*, 60:181–186, 2013.

[42] Jaakko Leppänen, Manuele Aufiero, Emil Fridman, Reuven Rachamin, and Steven van der Marck. Calculation of effective point kinetics parameters in the serpent 2 monte carlo code. *Annals of Nuclear Energy*, 65:272–279, 2014.

[43] S Michalek, J Hascik, and G Farkas. Possibilities of delayed neutron fraction ({beta}{sub eff}) calculation and measurement. 2008.

[44] AL Nichols, M Verpelli, and DL Aldama. *Handbook of nuclear data for safeguards: database extensions, August 2008*. IAEA, 2008.

[45] E Malambu and A Stankovskiy. Revised core design for myrrha– rev. 1.6. *SCK• CEN/3958903*, 2014.

[46] Luca Fiorito, Rafael Fernandez, and Gert Van den Eynde. Myrrha core design – version 1.8. *5th International Workshop on ACCELERATOR-DRIVEN SUB-CRITICAL SYSTEMS & THORIUM UTILIZATION*, 2019.

[47] Augusto Hernandez-Solis, Yohannes Molla, Edoaurd Malambu, Alexey Stankovskiy, and Gert Van den Eynde. Verification of the openmc homogenized myrrha-1.6 core model. In *EPJ Web of Conferences*, volume 247, page 04002. EDP Sciences, 2021.

[48] Yohannes Molla. Openmc modeling of the critical myrrha configuration: an emphasis in cross section homogenization. 2019.

[49] Heddy Barale. *Application of the SERPENT code to neutronic analyses of the MYRRHA core: a sensitivity and uncertainty approach*. PhD thesis, Politecnico di Torino, 2020.

[50] Filip Osuskỳ, Branislav Vrban, Peter Ballo, Štefan Čerba, Jakub Lüley, and Gabriel Farkas. Numerical multigroup transient analysis of slab nuclear reactor with thermal feedback. *Acta Polytechnica CTU Proceedings*, 4:62–67, 2016.

[51] Arjan JM Plompen, O Cabellos, C De Saint Jean, M Fleming, A Algora, M Angelone, P Archier, E Bauge, O Bersillon, A Blokhin, et al. The joint evaluated fission and fusion nuclear data library, jeff-3.3. *The European Physical Journal A*, 56(7):1–108, 2020.

[52] Bart L Sjenitzer. The dynamic monte carlo method for transient analysis of nuclear reactors. 2013.

[53] Margaux Faucher, Davide Mancusi, and Andrea Zoia. Variance-reduction methods for monte carlo kinetic simulations. In *MandC 2019*, 2019.

[54] Ville Valtavirta. Coupled calculations with serpent 2.1.29. Technical report, VTT, 2017.

[55] M Palmer. Propagation of uncertainty through mathematical operations. *Massachusetts Institute of*, 2003.

[56] Dušan Ćalić, Andrej Trkov, and Marjan Kromar. Use of lattice code dragon in reactor calculations. 2013.

[57] A Hébert and G Marleau. Generalization of the stamm'ler method for the self-shielding of resonant isotopes in arbitrary geometries. *Nuclear science and engineering*, 108(3):230–239, 1991.

[58] AE Aboanber and AA Nahla. Solution of the point kinetics equations in the presence of newtonian temperature feedback by padé approximations via the analytical inversion method. *Journal of Physics A: Mathematical and General*, 35(45):9609, 2002.

[59] Raghvendra Upadhyay, Goutham Dutta, Mohit Kumar Singh, Dalip Bera, and Neelesh Srivastava. A numerical solution of the point kinetics equations and its validation. *Mathematical Sciences International Research Journal*, 3:993–1005, 2014.

[60] Paul Romano. Analytical solution of the point kinetics equations, 2011. URL: https://github.com/paulromano/kinetics.

[61] A Santamarina and N Hfaiedh. The shem energy mesh for accurate fuel depletion and buc calculations. In *Proc. Int. Conf. Nuclear Criticality-Safety ICNC2007, St Petersburg*, 2007.

[62] Friederike Bostelmann, Bradley T Rearden, Winfried Zwermann, and Andreas Pautz. Scale/ampx multigroup libraries for sodium-cooled fast reactor systems. *Annals of Nuclear Energy*, 140:107102, 2020.

[63] Robert Macfarlane, Douglas W Muir, RM Boicourt, Albert Comstock Kahler III, and Jeremy Lloyd Conlin. The njoy nuclear data processing system, version 2016. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2017.

[64] Zoltán Szatmáry and János Valkó. *GRACE-a multigroup fast neutron spectrum code*. Hungarian Academy of Sciences, Central Research Institute for Physics, 1970.