UNIVERSITAT POLITÈCNICA DE VALÈNCIA

PhD Thesis

Integration Methods for the Time Dependent Neutron Diffusion Equation and other Approximations of the Neutron Transport Equation

March 2020

Author: Amanda María Carreño Sánchez

Supervisors: D. Damián Ginestar Peiró
D. Gumersindo J. Verdú Martín
D. Antoni Vidal Ferràndiz

Truth is much too complicated
to allow anything but approximations.

- John von Neumann

ABSTRACT

One of the most important targets in nuclear safety analyses is the fast and accurate computation of the power evolution inside of the reactor core. The distribution of neutrons can be described by the neutron transport Boltzmann equation. The solution of this equation for realistic nuclear reactors is not straightforward, and therefore, numerical approximations must be considered.

First, the thesis is focused on the attainment of the solution for several steady-state problems associated with neutron diffusion problem: the λ-modes, the γ-modes and the α-modes problems. A high order finite element method is used for the spatial discretization. Several characteristics of each type of spectral problem are compared and analyzed on different reactors.

Thereafter, several eigenvalue solvers and strategies are investigated to compute efficiently the algebraic eigenvalue problems obtained from the discretization. Most works devoted to solve the neutron diffusion equation are made for the approximation of two energy groups and without considering up-scattering. The main property of the proposed methodologies is that they depend on neither the reactor geometry, the type of eigenvalue problem nor the number of energy groups.

After that, the solution of the steady-state simplified spherical harmonics equations is obtained. The implementation of these equations has two main differences with respect to the neutron diffusion. First, the spatial discretization is made at level of pin. Thus, different meshes are studied. Second, the number of energy groups is commonly bigger than two. Therefore, block strategies are developed to optimize the computation of the algebraic eigenvalue problems associated.

Finally, an updated modal method is implemented to integrate the time-dependent neutron diffusion equation. Modal methods based on the expansion of the different spatial modes are presented and compared in several types of transients. Moreover, an adaptive time-step control is developed that avoids setting the time-step with a fixed value and it is adapted according to several error estimations.

RESUMEN

Uno de los objetivos más importantes en el análisis de la seguridad en el campo de la ingeniería nuclear es el cálculo, rápido y preciso, de la evolución de la potencia dentro del núcleo del reactor. La distribución de los neutrones se puede describir a través de la ecuación de transporte de Boltzmann. La solución de esta ecuación no puede obtenerse de manera sencilla para reactores realistas, y es por ello que se tienen que considerar aproximaciones numéricas.

En primer lugar, esta tesis se centra en obtener la solución para varios problemas estáticos asociados con la ecuación de difusión neutrónica: los modos λ, los modos γ y los modos α. Para la discretización espacial se ha utilizado un método de elementos finitos de alto orden. Diversas características de cada problema espectral se analizan y se comparan en diferentes reactores.

Después, se investigan varios métodos de cálculo para problemas de autovalores y estrategias para calcular los problemas algebraicos obtenidos a partir de la discretización espacial. La mayoría de los trabajos destinados a la resolución de la ecuación de difusión neutrónica están diseñados para la aproximación de dos grupos de energía, sin considerar dispersión de neutrones del grupo térmico al grupo rápido. La principal ventaja de la metodología que se propone es que no depende de la geometría del reactor, del tipo de problema de autovalores ni del número de grupos de energía del problema.

Tras esto, se obtiene la solución de las ecuaciones estacionarias de armónicos esféricos. La implementación de estas ecuaciones tiene dos principales diferencias respecto a la ecuación de difusión neutrónica. Primero, la discretización espacial se realiza a nivel de pin. Por tanto, se estudian diferentes tipos de mallas. Segundo, el número de grupos de energía es, generalmente, mayor que dos. De este modo, se desarrollan estrategias a bloques para optimizar el cálculo de los problemas algebraicos asociados.

Finalmente, se implementa un método modal actualizado para integrar la ecuación de difusión neutrónica dependiente del tiempo. Se presentan y comparan los métodos modales basados en desarrollos en función de los diferentes modos espaciales para varios tipos de transitorios. Además, también se desarrolla un control de paso de tiempo adaptativo, que evita la actualización de los modos de una manera fija y adapta el paso de tiempo en función de varias estimaciones del error.

RESUM

Un dels objectius més importants per a l'anàlisi de la seguretat en el camp de l'enginyeria nuclear és el càlcul, ràpid i precís, de l'evolució de la potència dins del nucli d'un reactor. La distribució dels neutrons pot modelar-se mitjançant l'equació del transport de Boltzmann. La solució d'aquesta equació per a un reactor realístic no pot obtenir's de manera senzilla. És per això que han de considerar-se aproximacions numèriques.

En primer lloc, la tesi se centra en l'obtenció de la solució per a diversos problemes estàtics associats amb l'equació de difusió neutrònica: els modes λ, els modes γ i els modes α. Per a la discretització espacial s'ha utilitzat un mètode d'elements finits d'alt ordre. Algunes de les característiques dels problemes espectrals s'analitzaran i es compararan per a diferents reactors.

Tanmateix, diversos solucionadors de problemes d'autovalors i estratègies es desenvolupen per a calcular els problemes obtinguts de la discretització espacial. La majoria dels treballs per a resoldre l'equació de difusió neutrònica estan dissenyats per a l'aproximació de dos grups d'energia i sense considerar dispersió de neutrons del grup tèrmic al grup ràpid. El principal avantatge de la metodologia exposada és que no depèn de la geometria del reactor, del tipus de problema d'autovalors ni del nombre de grups d'energia del problema.

Seguidament, s'obté la solució de les equacions estacionàries d'harmònics esfèrics. La implementació d'aquestes equacions té dues principals diferències respecte a l'equació de difusió. Primer, la discretització espacial es realitza a nivell de pin a partir de l'estudi de diferents malles. Segon, el nombre de grups d'energia és, generalment, major que dos. D'aquesta forma, es desenvolupen estratègies a blocs per a optimitzar el càlcul dels problemes algebraics associats.

Finalment, s'implementa un mètode modal amb actualitzacions dels modes per a integrar l'equació de difusió neutrònica dependent del temps. Es presenten i es comparen els mètodes modals basats en l'expansió dels diferents modes espacials per a diversos tipus de transitoris. A més a més, un control de pas de temps adaptatiu es desenvolupa, evitant l'actualització dels modes d'una manera fixa i adaptant el pas de temps en funció de vàries estimacions de l'error.

AGRADECIMIENTOS

En primer lugar me gustaría dar las gracias a Gúmer y Damián por confiar en mí y darme la oportunidad de empezar esta aventura. En particular, a Damián por su paciencia, estar siempre dispuesto a ayudarme y enseñarme la mayor parte matemática que incluye esta tesis. También a Gúmer, por encontrar mil huecos para formarme en cuestiones de ingeniería nuclear. De la misma manera, quisiera agradecer a Toni que haya compartido conmigo sus consejos y conocimientos, especialmente en la parte de programación. Sin la guía y el apoyo de ellos no habría podido realizar esta tesis.

En segundo lugar, a todo el Departamento de Ingeniería Química y Nuclear por hacerme sentir como una más desde el primer día. En especial, me gustaría destacar a Aina, Anto, Consu, Javi, María, Sergio y Sete por los buenos momentos que hemos compartido en esta etapa, y que en ocasiones han sido muy necesarios.

Gracias también al Departamento de Matemática Aplicada, al grupo de profesores de matemáticas de la Escuela Técnica y Superior de Diseño, por guiarme y permitirme impartir docencia en la universidad. En particular, quería expresar mi gratitud a Esther por toda su ayuda.

Por último, y no por ello menos importante, quería mencionar a mi familia y amigos que han estado siempre a mi lado. A mis padres, por su constante apoyo y cariño, les debo lo que soy. A mi hermano, por sacarme en toda circunstancia de la vida una sonrisa. Y por supuesto, a Sergio, por regalarme su tiempo y amor día tras día.

Amanda

CONTENTS

LIST OF SYMBOLS

Acronyms

ANM	Analytical Nodal Method
AVG	Average pin power per cent error
BIFPAM	Block Inverse-Free Preconditioned Arnoldi Method
BWR	Boiling Water Reactor
CPU	Central Processing Unit
FEM	Finite Element Method
GC	Gradiend Conjugate Method
GDM	Generalized Davidson Method
GMG	Geometric Multigrid preconditioner
GMRES	Generalized Minimal Residual Method
ICC	Incomplete Cholesky factorization
ILU	Incomplete Lower Upper factorization
IRAM	Implicit Restarted Arnoldi Method
KSM	Krylov-Schur Method
LU	Lower Upper factorization
LWR	Light Water Reactor
MGBNM	Modified Generalized Block Newton Method
MLFE	Multilevel Finite Element Method preconditioner
MRE	Mean relative pin power per cent error
NCM	Nodal Colocation Method
NEM	Nodal Expansion Method
PETSc	Portable, Extensible Toolkit for Scientific computations
PIM	Power Inverse Method

PWR	Pressurized Water Reactor
SLEPc	Scalable Library for Eigenvalue Problem computations
VVER	Water-Water Energetic Reactor

Symbols

N_0	Avogadro's number, 6.022×10^{23}
N_d	Density number
σ	Microscopic cross section
λ_k^d	Neutron precursors decay constant
β	Total fraction of delayed neutrons
β_{al}	Albedo coefficient
β_k	Fraction of delayed neutrons of group kp
n	Neutron density probability distribution
E	Energy of the neutron
$\vec{\Omega}$	Direction of the neutrons
$\vec{r}$	Spatial position
R	Reaction rate
$\vec{J}$	Net current vector
J_n	Net current in the direction n
Q^{ex}	External source
C_k	Concentration of delayed neutron precursors of group k
$f(E)$	Energy dependent spectral weighting function
μ	Cosine of angle of the neutron velocity
μ_0	Cosine of angle scattering
$\Psi(\vec{r}, E, \vec{\Omega}, t)$	Angular neutron flux
$\Psi_g(\vec{r}, \vec{\Omega}, t)$	Angular neutron flux of energy group g

$\psi_g^{\delta}(\vec{r},\vec{\Omega})$	Angular δ-mode of energy group g
$\Phi_g(\vec{r},t)$	Scalar neutron flux of energy group g
$\phi_g^{\delta}(\vec{r})$	Scalar δ-mode of energy group g
$J^{\pm}$	Neutron partial current
Σ_t	Macroscopic total cross-section
Σ_r	Macroscopic removal cross-section
Σ_a	Macroscopic absorption cross-section
Σ_s	Macroscopic scattering cross-section
Σ_f	Macroscopic fission cross-section
D	Diffusion coefficient
χ	Spectrum of neutrons produced by fission
ν	Mean number of neutrons produced per fission
$\mathcal{L}$	Neutron leackage differential operator
$\mathcal{F}$	Neutron fission differential operator
$\mathcal{S}$	Neutron scattering differential operator
$\mathcal{V}$	Neutron velocity differential operator
V	Reactor domain
V_c	Mesh subdomain or cell
Γ	Problem boundary
Γ_c	Subdomain boundary at cell c
G	Number of energy groups
K	Number of precursor groups
N_{dofs}	Degrees of freedom
N_c	Number of homogenized cells in the mesh
q	Total number of modes

V_t	Total volume of the problem
$\mathcal{N}_i$	Shape function of the support point i
p	Polynomial degree of the shape functions of the finite element method
l_i^p	Lagrange basis polynomial of degree p
$\lvert\mathbf{J^k}\rvert$	Jacobian of the affine transformation
k_{eff}	Multiplicative factor in the system
P_n	Legendre polynomial of order n
u_g^n	Pseudo-flux for the moment n in energy group g
N	Total number of moments

CHAPTER 1

INTRODUCTION

1.1 Overview of modal analysis

The modal analysis, based on the definition given by Fu and He in 2001, is the process of determining the inherent dynamic characteristics of a system in the form of natural frequencies, damping factors and mode shapes; and using them to formulate a mathematical model for its dynamic behaviour.

The dynamical behaviour of a system can be physically decomposed by position and frequency. This is illustrated, for instance, in the analytical solution of PDEs for continuous systems such as beams and strings. The modal analysis assumes that a dynamic system can be represented as the linear combination of a set of harmonics known as the natural modes. Each mode depends on the dynamic system and it is defined by its physical properties and spatial distribution. Its shape can be real or complex and each one usually is associated with a natural frequency. The weight of each mode in the overall description of the movement is computed both by properties of the excitation sources and by the modes shapes of the system.

In the last decades, there are countless applications of modal analysis in fields of engineering, technology and science. It is not possible to introduce each one of these applications; nevertheless, a brief exposition of some practical works will help to understand the potential of modal analysis. Traditionally, most of practical work came from engineerings such as aeronautics, mechanics and

automobiles. Nevertheless, in the last years, it becomes more interdisciplinary discovering many applications for civil engineering, biomechanical problems, space structures, optics, acoustical instruments, transport and nuclear engineering, etc.

In automotive engineering, there is great interest to understand the dynamic properties of the vehicles to improve the design of automotive components and the enhancement of dynamic properties. For instance, troubleshooting tool, by using modal analysis, plays an important role in the study of vehicle noise and vibration harshness mainly for body-in-white or a sub-frame structure (Freeman and Weilnau, 2017). This technique derives experimentally natural frequencies, damping factors, and mode shapes to understand the structural characteristics. Another application is the study of the vehicle noise (Panza, 2015).

Other fields where the modal analysis is of great importance are the aeronautical and astronautical industries. In fact, the development of modal analysis has been associated with the rapid progress of this industry. Spacecraft structures require high requirements for structural integrity and dynamic behavior. Several modal tests have been conducted in areas ranging from an aircraft frame, a satellite to an unmanned aerial vehicle. (Kerschen et al., 2013; Li et al., 2016; Boudjemai et al., 2012).

On the other hand, modal analysis is widely applied to structural analysis for studying the dynamic behaviour of civil structures under seismic and wind charges. The response of a construction due to ambient vibration or external loading relies on accurate mathematical models that can be derived for instance by modal analysis. Examples of such applications range from tall buildings, soil-structure interactions, bridge testings to a dam-foundation systems (Pioldi et al., 2017; Brownjohn et al., 2010). Furthermore, acoustic modal analysis has provided essential information in the design of speaker cabinets with improved sound quality. Also, for the studies of instruments such as violins to provide scientific data behind of instrument makers (Chaigne and Kergomard, 2016).

Modal analysis has also developed in other fields less known as the waveguides analysis, very popular for instance, in the microwaves industry. Modal techniques allow analyzing waveguide junctions and the propagation characteristics of a given waveguide. An example is the mode matching method proposed by Wexley (Wexler, 1967) that permits the efficient computation of the scattering parameters of waveguide structures involving different planar discontinuities. Waveguides are also important in the optical field. Many works are devoted to analyze, through modal techniques, for example, the chromatic dispersion of optical fiber (Silvestre et al., 2005) or shaped optical dielectric waveguides (Ortega-Moñux et al., 2006).

In nuclear engineering, modal methods have been successfully used to study the dynamics of reactor cores and to classify BWR instabilities (Miró et al., 2000; Miró et al., 2002). Recently, modal analysis is developed to decompose the neutron noise produced by fuel assembly vibrations (Yamamoto and Sakamoto, 2019).

1.2 Motivation and objectives

One of the most important subjects in the nuclear safety analysis is the computation of the power evolution inside of the reactor core. An accurate approximation is essential for the design and safety of a nuclear reactor and other nuclear systems. This power depends basically on the transport of neutrons that can be modeled through the neutron Boltzmann equation or the neutron transport equation. The solution of this equation in a realistic nuclear reactor is not straightforward, and therefore, numerical approximations must be considered. The main aim of this thesis is the study and the implementation of different techniques to integrate efficiently the multigroup neutron diffusion equation and the simplified spherical harmonics equation.

First, it is focused on the computation of several spatial modes associated with the neutron diffusion equation. The previous section has emphasized the importance of the modal analysis in several applications for nuclear engineering. The majority of works are devoted to obtain the known λ-modes or in less cases, the α-modes and the γ-modes. In this thesis, we analyze the spectral properties these types of modes. For the spatial discretization of the equations, a high order finite element method is considered. The main reason to use this methodology is the capability of modelling any kind of geometry: using structured and unstructured meshes.

After that, some eigenvalue solvers and strategies are proposed to compute the algebraic eigenvalue problems obtained from the discretization. Most works related to solve the neutron diffusion equation are made for the approximation of two energy groups and without considering up-scattering. The main property of the proposed methodologies is that they are depend on neither the reactor geometry, the type of eigenvalue problem nor the number of energy groups.

Thereafter, the solution of the steady-state simplified spherical harmonics equation (SP_N) is obtained. The implementation of these equations has two main differences concerning for the neutron diffusion computation. The first one is that the spatial discretization is usually made at pin level. Thus, different strategies of meshes are studied. The second one is that the number of energy groups is commonly bigger than two. Therefore, block strategies are developed to optimize

the computation of the associated algebraic eigenvalue problems. Furthermore, the sizes of the problems for these equations can be sufficiently large to be unfeasible to be solved in personal computers. Thus, a matrix-free methodology, that avoids the allocation of the matrices in memory, has been studied, as well as, eigenvalue solvers and preconditioners do no need the full problem matrices allocated.

Finally, an updated modal method is efficiently implemented to integrate the time-dependent neutron diffusion equation. The modal methods are based on the expansion of the different types of eigenfunctions studied. Moreover, an adaptive time-step control is developed that avoids setting the time-step with a fixed value and it is adapted along the transient according to several error estimations.

The objectives of this thesis can be summarized as follows:

1. The implementation of a finite element code to solve the λ, the α and the γ-modes problem associated with the neutron diffusion equation to analyze the different spectral problems.
2. The optimization of the eigensolvers throught different block iterative methods, initialization techniques and preconditioners.
3. The resolution of the simplified spherical harmonics equations (SP_N) in an efficient way.
4. The integration of the time-dependent neutron diffusion equation with an updated modal methodology based on different spatial modes.
5. The development of an adaptive time-step control to improve the updated modal method.

1.3 Thesis outline

The thesis is organized in 6 chapters and 2 appendices. Chapter 2 exposes the neutron transport equations and presents the different approximations that are used in the next chapters: the time-dependent neutron diffusion equation, the spatial modes problems associated with the neutron diffusion equation and the λ-modes problem associated with the simplified harmonics equations (SP_N equations). Moreover, this Chapter includes the definition some basic neutron magnitudes and concepts.

Chapter 3 describes and tests the finite element method used for the spatial discretization of the steady-state equation. Numerical results compare the spatial

modes for the neutron diffusion equation and show the distribution of the λ-modes for the SP_N equations.

Chapter 4 presents the methods to solve the algebraic eigenvalue problems obtained from the spatial discretization. This Chapter describes some of the most used methods in nuclear engineering computations and presents the performance of several block eigensolvers as alternative to these ones. The computational efficiency of each eigenvalue solver can be improved through different techniques. In this thesis, some initialization strategies, preconditioners and different matrix allocations techniques are studied and tested on several reactors. The eigenvalue solvers presented are used in both neutron diffusion and SP_N computations.

Chapter 5 is devoted to integrate the neutron diffusion equation. The modal methodology with different spatial modes is studied to this aim. First, the modal methodology is compared for several spatial modes. Then, the updated modal methodology is used to avoid to use a high number of modes in the modal expansions. Finally, an adaptive updated modal method is developed such that the time-steps to update the modes are chosen depending on the state of the transient to minimize the error and maximize the performance of the method. This methodology is compared with a backward differential method for different transients.

Chapter 6 collects the main conclusions and results of this thesis. Appendix A develops the analytic solution of the spatial modes problems associated with the neutron diffusion equation for a three-dimensional homogeneous reactor. Finally, Appendix B describes the benchmarks used along the manuscript.

CHAPTER 2

THE NEUTRON TRANSPORT EQUATION

The prediction of the position of neutrons, the direction they are going and the velocity they are moving is essential for the design and the safety of nuclear reactors because neutros cause the nuclear fission of nuclei. This behaviour depends on the interactions between neutrons and nucleus and it can be described (under several assumptions) by the neutron transport equation, also known as Boltzmann equation. It states that the variation of the number of neutrons located in a control volume is due to the imbalance between the number of neutrons appearing and disappearing in this control volume.

This equation is an integro-differential equation with (usually) seven independent variables, whose solution is not smooth, and which can only be solved analytically for the simplest problems. For this reason, all neutron problems of practical interest must be solved either approximately or numerically. Two type of methods can be found to simulate and approximate neutron transport and interactions in the reactor. Deterministic methods that solve the Boltzmann transport equation numerically as a differential equation. And stochastic methods where discrete particle histories are tracked and averaged in a random walk directed by measured interaction probabilities. Even though, deterministic methods are faster, both types of methods are time demanding, to obtain approximations in realistic reactors.

The approximated equations for deterministic methods are normally classified by its angular dependence treatment. The most used approximation over the years is

the neutron diffusion equation. This equation, even if its use is limited by certain conditions, gives accurate solutions in a relatively short time. More accurate approximations of the neutron transport equations are used through different angular discretizations such as the discrete ordinates (S_N) or the spherical harmonics expansion (P_N) for the angular variable. Both discretizations, although different, can yield to equivalent solutions by taking an appropriate quadrature set in the definition of the discrete ordinates method, (Sanchez, 2012). However, in spite of the angular discretization, a large set of equations is needed to be solved to obtain precise results.

For the case of P_N equations, a simplified formulation was developed by Gelbard in (Gelbard, 1960) to reduce the computational cost known as the simplified P_N equations or SP_N equations. They are derived from the P_N equations where odd moments are solved and replaced again in the the equations, leading a simplified formulation where the number of variables is decreased. Gelbard applies heuristic arguments to justify the approximation. Over the years, several works (Larsen et al., 1996; Klose and Larsen, 2006) have verified its approach.

This chapter describes the underlying theory of the neutron transport equation and two types of approximations: the neutron diffusion equation and the simplified spherical harmonic equations. Section introduces 2.1 some basic magnitudes used in the context of the transport equation. Section 2.2 describes the neutron transport equation as well as the boundary conditions for this problem. Section 2.3 presents different modal problems associated with the neutron transport equation. Section 2.4 describes the time-dependent neutron diffusion equation. Section 2.5 presents the spatial modes associated with the neutron diffusion equation. Moreover, this Section includes the definition of the adjoint problems. Finally, Section 2.6 exposes the simplified spherical harmonic equations in steady-state.

2.1 Concepts of reactor physics

Nuclear reactors are based on the extraction of energy produced by nuclear fission reactions. In this process, a large fissile atomic nucleus such as U-235 or Pu-239 absorbs a neutron and it may undergo a nuclear fission. The heavy nucleus splits into two or more lighter nuclei, (the fission products), releasing kinetic energy, gamma radiation, and free neutrons (Lamarsh and Baratta, 2001). The neutrons produced are:

- **prompt neutrons**, emitted immediately after the fission ($\sim 10^{-14}$ s);

- **delayed neutrons**, emitted after beta decay of one of the fission fragments (neutron precursors) anytime from a few milliseconds to a few minutes later because a small fraction of precursors can undergo the neutron emission instead of the gamma emission.

Many precursors of delayed neutrons may appear, each having its *neutron decay constant*, λ_d. Frequently, neutron precursors are grouped into six groups according to their half-life. We denote the decay constant of the group k as λ_k^d. Moreover, we call the *fraction of delayed neutrons of group* k, as β_k. It represents the number of delayed neutrons in the group k divided by the total number of neutrons emitted. The number of precursor groups is K. The *total fraction of delayed neutrons*, β, is given by

$$\beta = \sum_{k=1}^{K} \beta_k \,. \tag{2.1}$$

The total fraction of delayed neutrons is small, being, for instance, 0.0127 for U-238, 0.0065 for U-235 and 0.0021 for Pu-239.

On the other hand, the fission products have different energies. The spectrum of the neutrons produced by fission (*Watt Fission Spectrum*) at energy E is denoted by $\chi(E)$. Therefore, the fraction of neutrons with energy between E and $E + \mathrm{d}E$ is given by $\chi(E)\mathrm{d}E$. As emitted neutrons can be prompt or delayed, we make the distinction by the superindex p for the prompt neutrons and d, k for delayed neutrons of delayed group k. These spectrums satisfy

$$\int_0^\infty \chi^p(E)\ \mathrm{d}E\ = 1, \qquad \int_0^\infty \chi^{d,k}(E)\ \mathrm{d}E\ = 1, \qquad \forall k = 1, \ldots, K. \tag{2.2}$$

Furthermore, we define ν that represents the average number of neutrons obtained per fission.

Fission is not the only type of reactions of importance (Lamarsh and Baratta, 2001; Demazière, 2019) and other nuclear reactions need to be taken into account. In reactor physics, these reactions are divided into two types:

- **Scattering reactions**, where a neutron collides on a target nucleus and can change its energy and direction after the collision. An scattering reaction can also lead to the formation of a compound nucleus before neutron emission. They can be elastic collisions or inelastic collisions. The laws of conservation of momentum and kinetic energy govern the elastic scattering interactions. Inelastic scattering causes a loss of kinetic energy

of the neutron due to an increase of the energy state of the nucleus. This kind of reaction is denoted by the subscript s.

- **Absorption reactions**, where a neutron is absorbed by a target nucleus. In this case, there are two possibilities: either the compound nucleus does not emit neutrons (i. e. it emits γ rays, α particles, etc), or the compound nucleus does emit neutrons. The last one is referred to as fission reaction. Absorptions reactions are denoted by the subscript a and fission reactions are denoted by the subscript f.

To quantify the probability of a nuclear reaction taking place, the *microscopic cross-section* is defined as

$$\sigma_\varrho := \frac{R_\varrho}{n\mathrm{v}N_n}, \tag{2.3}$$

where R_ϱ denotes the total number of collisions of type ϱ per second and n the number of neutrons travelling with velocity v (cm/s) in a material with N_n nucleus in the target. The subscript ϱ denotes the reaction type $\rho = s, a, t$. The sum of the cross-sections for all possible interactions is known as the *total cross-section* and is indicated by the symbol σ_t; that is

$$\sigma_t = \sigma_s + \sigma_a. \tag{2.4}$$

The value of σ_ϱ remits to the concept of cross-sectional area of interaction presented to the neutron for a reaction of type ϱ. Microscopic cross section are usually measured in barns where 1 barn = 10^{-24} cm^2. This cross-sectional area can be much larger than the geometric cross section of the nucleus (Stacey, 2018).

Usually, the product of the atom density N_d and microscopic cross section σ_ϱ appears in the equations of nuclear engineering. Thus, this value is refereed to as *macroscopic cross-section* and it is denoted by Σ_ϱ. In particular, $\Sigma_t := N_d\sigma_t$ denotes the total macroscopic cross-section; Σ_f denotes the fission macroscopic cross-section; Σ_s, the scattering macroscopic cross-section; and Σ_a, the absorption macroscopic cross-section. This variable indicates the probability of iteration of a neutron per unit path length. Macroscopic cross-sections are measured in 1/cm (Stacey, 2018). From this point onwards, all cross section will be macroscopic by removing the details of the interaction processes.

The complex behaviour of neutron cross sections means that these cannot be obtained from first principles using properties of the nucleus (Weinberg and Wigner, 1958). Therefore, data must be calculated empirically as a function of energy for each nuclide and reaction. The estimation of the neutron cross

sections has required years of effort in measuring, calculating and evaluating cross sections for hundreds of isotopes. These efforts are gathered in *evaluated nuclear data files* (ENDF) that contain the cross sections of the main reactions, the energy and the angular distributions of the resulting particles. Nowadays, the most complete collection of experimental results is the EXFOR computer library (Otuka et al., 2014). It includes the major evaluated nuclear data files: United States Evaluated Nuclear Data File (END/B-VII.1), Joint Evaluated File of NEA Countries (JEFF-3.2), Japanese Evaluated Nuclear Data Library (JENDL-4.0) and Russian Evaluated Nuclear Data File (BROND-3.1), among others (Vidal Ferràndiz, 2018).

2.1.1 Functions in reactor physics

The distribution of neutrons depends on independent variables with space support of dimension seven: the position, $\vec{r}$, described by three spatial coordinates; the direction of travel, $\vec{\Omega}$, given by two angles; the particle energy, E, and finally, the time denoted by t.

The **neutron density probability distribution**, $n(\vec{r},\, E,\, \vec{\Omega},\, t)$, is defined such that $n(\vec{r},\, E,\, \vec{\Omega},\, t)\ \mathrm{d}V\, \mathrm{d}\vec{\Omega}\, \mathrm{d}E$ is the number of neutrons in a differential volume element $\mathrm{d}V$ about $\vec{r}$ travelling in the direction $\mathrm{d}\vec{\Omega}$ around $\vec{\Omega}$ with energies between E and $E + \mathrm{d}E$ at time t (Henry, 1975; Hébert, 2009).

Normally, the transport problems are expressed in terms of the **angular neutron flux** given by

$$\Psi(\vec{r},\, E,\, \vec{\Omega},\, t) \coloneqq \upsilon(E) n(\vec{r},\, E,\, \vec{\Omega},\, t)\,, \tag{2.5}$$

where $\upsilon(E)$ is the neutron speed. The angular flux can be defined as the total path travelled during $\mathrm{d}t$ by all particles in the differential phase space volume $\mathrm{d}V\, \mathrm{d}\vec{\Omega}\, \mathrm{d}E$.

Sometimes, the direction is not relevant in the computation of reaction rates. In this cases, the **scalar neutron flux** is used integrating ψ over all directions. It can expressed as

$$\Phi(\vec{r},\, E,\, t) \coloneqq \int_{(4\pi)} \Psi(\vec{r},\, \vec{\Omega},\, E, t)\, \mathrm{d}\Omega. \tag{2.6}$$

This term does not mean a flow of neutrons through a surface, but it corresponds to the total length travelled by all neutrons per unit time and volume. It can be interpreted as the number of neutrons per unit area, energy and time.

The **net current density vector** is a vectorial quantity defined as

$$\vec{J}(\vec{r},\, E,\, t) \coloneqq \int\limits_{(4\pi)} \vec{\Omega}\, \Psi(\vec{r},\, \vec{\Omega},\, E,\, t)\, \mathrm{d}\Omega. \tag{2.7}$$

Given a position, energy and time, the definition of

$$J_n(\vec{r},\, E,\, t) \coloneqq \vec{n} \cdot \vec{J}(\vec{r},\, \vec{\Omega},\, E,\, t) = \int\limits_{(4\pi)} (\vec{n} \cdot \vec{\Omega}) \Psi(\vec{r},\, \vec{\Omega},\, E,\, t)\, \mathrm{d}\Omega\,. \tag{2.8}$$

gives the net number of particles crossing per unit area of surface per unit time and per unit energy in the positive direction of the normal vector, $\vec{n}$. The angular current is positive if the particle crosses in the direction of $\vec{n}$, and negative otherwise. Thus, we can define the *outgoing current*, J^+, and *incoming current*, J^- by integrating the angular current over outgoing and incoming directions, respectively, as

$$J_n^+ = \int_{\vec{\Omega}\cdot\vec{n}>0} \vec{n} \cdot \vec{\Omega}\, \Psi(\vec{r},\, \vec{\Omega},\, E,\, t)\, \mathrm{d}\Omega, \qquad J_n^- = -\int_{\vec{\Omega}\cdot\vec{n}<0} \vec{n} \cdot \vec{\Omega}\, \Psi(\vec{r},\, \vec{\Omega},\, E,\, t)\, \mathrm{d}\Omega,$$

so that

$$J_n(\vec{r},\, E,\, t) = J_n^+(\vec{r},\, E,\, t) - J_n^-(\vec{r},\, E,\, t)\,. \tag{2.9}$$

2.2 The neutron transport equation

The behaviour of a nuclear reactor core is governed by the distribution of the neutrons (in space, angle, energy and time) in the system. This can be modelled by the neutron transport equation, often called the Boltzmann transport equation. This equation is a balance between the proportion neutrons which enter and the proportion of neutrons which exit in a control volume (Lewis and Miller, 1984). This equation, assuming that delayed neutrons and fission neutrons are emitted

isotropically in the reference system, can be expressed as

$$\begin{aligned}\frac{1}{v}\frac{\partial\Psi}{\partial t}(\vec{r},\,E,\,\vec{\Omega},\,t) = &-\vec{\Omega}\cdot\vec{\nabla}\,\Psi(\vec{r},\,E,\,\vec{\Omega},\,t) - \Sigma_t(\vec{r},\,E,\,t)\Psi(\vec{r},\,E,\vec{\Omega},\,t)\\ &+\int_0^\infty \mathrm{d}E' \int_{(4\pi)} \mathrm{d}\Omega'\,\Sigma_s(\vec{r},\,E'\to E,\,\vec{\Omega}'\cdot\vec{\Omega},\,t)\,\Psi(\vec{r},E',\vec{\Omega}',t)\\ &+(1-\beta)\frac{\chi^p(E)}{4\pi}\int_0^\infty \nu\Sigma_f(\vec{r},\,E',\,t)\,\Phi(\vec{r},\,E',\,t)\mathrm{d}E'\\ &+\sum_{k=1}^{K}\lambda_k^d\frac{\chi^{d,k}(E)}{4\pi}\mathcal{C}_k(\vec{r},\,t) + Q^{\mathrm{ex}}(\vec{r},\,E,\,\vec{\Omega},\,t).\end{aligned} \tag{2.10}$$

The first member, $\frac{1}{v}\frac{\partial\Psi}{\partial t}$, represents the time-variation of the number of neutrons. The first term on the right hand side, $\vec{\Omega}\cdot\vec{\nabla}\,\Psi$, represents the neutron transfer going out the phase control volume during $\mathrm{d}t$, where $\vec{\Omega}$ is the unit vector denoting the direction of neutrons. The second term, $\Sigma_t\,\Psi$, describes the disappearance of neutrons during $\mathrm{d}t$, i.e. the rate at which neutrons are absorbed or scattered to other energies or directions. The third term represents the neutrons introduced into the volume element by scattering from other energies and directions. In writing $\Sigma_s(\vec{r},\,E'\to E,\,\vec{\Omega}'\cdot\vec{\Omega},\,t)$, the dot indicates that scattering in media with randomly distributed scattering centers, is rotationally invariant. That means that the probability that a neutron scatter from direction $\vec{\Omega}$ to direction $\vec{\Omega}'$ depends only on the scattering angle θ_0 between $\vec{\Omega}$ and $\vec{\Omega}'$. The fourth term represents the number of prompt neutrons introduced inside the volume element by fission processes. The fifth term takes into account the quantity of delayed neutrons that appear from the precursor decay. Finally, the emission density of neutrons of a possible external source is given by $Q^{\mathrm{ex}}(\vec{r},E,\vec{\Omega},t)$.

The concentration of delayed neutron precursors, $\mathcal{C}_k$, for the precursor group $k=1,\ldots,K$, satisfies the equation

$$\begin{aligned}\frac{\partial\mathcal{C}_k}{\partial t}(\vec{r},t) &= \beta_k\int_0^\infty dE\int_{(4\pi)}\mathrm{d}\Omega\,\nu\Sigma_f(\vec{r},E,t)\,\Psi(\vec{r},E,\vec{\Omega},t) - \lambda^{d,k}\mathcal{C}_k(\vec{r},t),\\ &= \beta_k\int_0^\infty dE\,\nu\Sigma_f(\vec{r},E,t)\,\Phi(\vec{r},E,t) - \lambda^{d,k}\mathcal{C}_k(\vec{r},t),\end{aligned} \tag{2.11}$$

where ν is the average number of neutrons released per fission.

The neutron transport equation is an integro-differential equation whose solution is unique and non-negative provided all cross sections, sources and non negative initial and boundary conditions (Lewis and Miller, 1984).

In the following, we develop *the multigroup neutron transport* equation approximation by discretizing the energy variable E into G intervals. We suppose the scalar energy starts in $E_G = 0$ and ends in a sufficiently large value for energy E_0. The neutrons with energies between E_g and E_{g-1} with $E_g < E_{g-1}$ belong to group g. Under this assumption, one can define the group angular flux Ψ as

$$\Psi_g(\vec{r}, \vec{\Omega}, t) = \int_{E_g}^{E_{g-1}} \Psi(\vec{r},\, \vec{\Omega},\, E, t)\mathrm{d}E\,. \tag{2.12}$$

In this way, one can suppose that a function $f(E)$ can be defined such that

$$\Psi(\vec{r},\, \vec{\Omega},\, E, t) \approx f(E)\Psi_g(\vec{r},\, \vec{\Omega}, t)\,, \tag{2.13}$$

where the energy-dependent weighting function $f(E)$ is normalized as

$$\int_{E_g}^{E_{g-1}} f(E)\mathrm{d}E = 1\,. \tag{2.14}$$

This formalism is needed to preserve the quantities involved in the neutron transport equations.

Analogously, to the angular flux, the neutron transport equation, (Equation (2.10)), can be integrated for all energy interval to obtain the multigroup neutron transport equation

$$\begin{aligned}
\frac{1}{v_g}\frac{\partial \Psi_g}{\partial t}(\vec{r},\, \vec{\Omega},\, t) = &- \vec{\Omega}\cdot\vec{\nabla}\,\Psi_g(\vec{r},\, \vec{\Omega},\, t) - \Sigma_{tg}(\vec{r},\, t)\Psi_g(\vec{r}, \vec{\Omega},\, t) \\
&+ \sum_{g'=1}^{G} \int_{(4\pi)} \Sigma_{s,\,g'g}(\vec{r},\, \vec{\Omega}'\cdot\vec{\Omega},\, t)\,\Psi_g(\vec{r}, \vec{\Omega}', t)\; d\Omega' \\
&+ \sum_{g'=1}^{G} (1-\beta)\frac{\chi_g^p}{4\pi}\nu_g\Sigma_{fg'}(\vec{r}, t)\Phi_{g'}(\vec{r},\, t) \\
&+ \sum_{k=1}^{K} \lambda_k^d \frac{\chi_g^{d,k}}{4\pi}\mathcal{C}_k(\vec{r},\, t) + Q_g^{\mathrm{ex}}(\vec{r},\, \vec{\Omega},\, t)
\end{aligned} \tag{2.15}$$

and the concentration on neutron precursors

$$\frac{\partial \mathcal{C}_k}{\partial t}(\vec{r}, t) = \beta_k \sum_{g'=1}^{G} \nu\Sigma_{fg'}(\vec{r}, t)\, \Phi_{g'}(\vec{r}, t) - \lambda_k^d \mathcal{C}_k(\vec{r}, t)\,. \tag{2.16}$$

The magnitudes associated with each group g are defined as

$$\Phi_g(\vec{r}, t) \coloneqq \int_{E_g}^{E_{g-1}} \Phi(\vec{r}, E, t) f(E)\,\mathrm{d}E\,, \tag{2.17}$$

$$\frac{1}{\mathrm{v}_g} \coloneqq \int_{E_g}^{E_{g-1}} \frac{1}{\mathrm{v}(E)} f(E)\,\mathrm{d}E\,, \tag{2.18}$$

$$\Sigma_{tg}(\vec{r}, \vec{\Omega}, t) \coloneqq \int_{E_g}^{E_{g-1}} \Sigma_t(\vec{r}, E, t) f(E)\,\mathrm{d}E\,, \tag{2.19}$$

$$\Sigma_{s\,g'g}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}, t) \coloneqq \int_{E'_g}^{E_{g'-1}} \int_{E_g}^{E_{g-1}} \Sigma_s(\vec{r}, E' \to E, \vec{\Omega}' \cdot \vec{\Omega}, t) f(E)\mathrm{d}E\,\mathrm{d}E'\,, \tag{2.20}$$

$$\nu_g\Sigma_{f,\,g}(\vec{r}, t) \coloneqq \int_{E_g}^{E_{g-1}} \nu\Sigma_f(\vec{r}, E, t) f(E)\,\mathrm{d}E\,, \tag{2.21}$$

$$\chi_g^p \coloneqq \int_{E_g}^{E_{g-1}} \chi^p(E) f(E)\,\mathrm{d}E\,, \tag{2.22}$$

$$\chi_g^{d,k} \coloneqq \int_{E_g}^{E_{g-1}} \chi^{d,k}(E) f(E)\,\mathrm{d}E, \quad k = 1, \ldots, K\,, \tag{2.23}$$

$$Q_g^{\mathrm{ex}} \coloneqq \int_{E_g}^{E_{g-1}} Q^{\mathrm{ex}} f(E)\,\mathrm{d}E\,. \tag{2.24}$$

In the following, we assume that there are not external sources in the neutron transport equation, i.e. $Q^{\mathrm{ex}} = 0$.

2.2.1 Boundary conditions

The reactor domain V is surrounded by a boundary $\Gamma \equiv \partial V$ where boundary conditions are imposed. The solution of the neutron transport equation (Equation (2.10)) requires the knowledge of the incoming angular flux $\Psi(\vec{r}_0, \vec{\Omega}_{\text{in}}, E)$ for $\vec{\Omega}_{\text{in}} \cdot \vec{n} < 0$ where $\vec{n}$ is the outward normal vector at $\vec{r}_0 \in \Gamma$. The most usual boundary conditions are:

- **Albedo boundary condition**. The incoming flux is related with the known outgoing flux by means of an albedo coefficient β_{al}. This condition can be written as

$$\Psi(\vec{r}_0, \vec{\Omega}_{\text{in}}, E) = \beta_{\text{al}}\, \Psi(\vec{r}_0, \vec{\Omega}_{\text{out}}, E), \text{ for } \vec{\Omega}_{\text{in}} \cdot \vec{n} < 0,\ \vec{r}_0 \in \Gamma\,, \tag{2.25}$$

  where $\vec{\Omega}_{\text{in}}$ is the refection angle corresponding to an outgoing direction $\vec{\Omega}_{\text{out}}$.

- **Vacuum boundary condition**. Also it is known as free surface boundary condition and it can be expressed as

$$\Psi(\vec{r}_0, \vec{\Omega}_{\text{in}}) = 0, \text{ for } \vec{\Omega}_{\text{in}} \cdot \vec{n} < 0,\ \vec{r}_0 \in \Gamma\,. \tag{2.26}$$

  Note that, they are a particular case of albedo condition taking $\beta_{\text{al}} = 0$.

- **Reflective boundary condition** can be defined as

$$\Psi(\vec{r}_0, \vec{\Omega}_{\text{in}}) = \Psi(\vec{r}_0, \vec{\Omega}_{\text{out}}) \text{ for } \vec{\Omega}_{\text{in}} \cdot \vec{n} < 0,\ \vec{r}_0 \in \Gamma\,. \tag{2.27}$$

  This type of boundary condition is a particular case of albedo boundary conditions taking $\beta_{\text{al}} = 1$, since all outgoing particles are reflected back.

- **White boundary condition**. It is a reflective condition where all neutrons leaving the system through the boundary are isotropically emitted back into the domain. This is expressed as

$$\Psi(\vec{r}_0, \vec{\Omega}_{\text{in}}) = \frac{J^+(\vec{r}_0)}{\pi}, \qquad \text{for } \vec{\Omega}_{\text{in}} \cdot \vec{n} < 0,\ \vec{r}_0 \in \Gamma\,. \tag{2.28}$$

2.3 Spatial modes definition

Some steady-state transport calculations are carried out by using the equation without taking into account the time dependence. However, in this multiplying system (i.e. $\Sigma_f \neq 0$), the critical state of the reactor must be considered. It is

said that a reactor is in critical state if there is a self-sustaining time-independent chain reaction in the absence of external sources of neutrons, $Q^{\text{ex}} = 0$. This means that if neutrons are inserted in a critical system, a time-independent distribution of neutrons will exist in which the rate of fission neutron production is just equal to the losses due to absorption and leakage. When an equilibrium cannot be established, the asymptotic distribution of neutrons will not be in steady state and will either increase or decrease exponentially. In these cases, the system would be supercritical or subcritical, respectively.

Different kind of spatial modes can be defined for the neutron transport equation to force the criticality of the system by modifying the cross-sections in different ways (Bell and Glasstone, 1970; Henry, 1975; Ronen et al., 1976), obtaining different eigenvalue problems such as the λ-modes (denoted in these works as k-modes), the α-modes, γ-modes or the $\hat{\lambda}$-modes. In (Ronen et al., 1976; Velarde et al., 1978) these spatial modes are discussed and compared for fast neutron plutonium systems. Other modal problems are the δ-modes. These are proposed by Avvakumov et al. by connecting to self-adjoint part of the operator of neutron absorption-generation to make an a priori estimate of neutron flux dynamics. Modal methods have been developed for these modes (Avvakumov et al., 2018b). In this thesis, the modal study has been restricted to the λ-modes, the α-modes and the γ-modes.

The spatial modes have been defined for different approximations of the neutron transport equation and used for different purposes. The λ-modes are the most common problem to study the criticality of the system (Verdú et al., 1994). Moreover, these modes have been successfully used to study and classify the neutronic oscillations in BWR reactors (March-Leuba and Blakeman, 1991; Verdú et al., 1998; Miró et al., 2000) and to develop modal methods to solve the time dependent neutron diffusion equation (Miró et al., 2002). The λ-modes problem is obtained by diving the fission term of the neutron transport equation (Equation (2.15)) by a positive number λ as

$$\begin{aligned}\vec{\Omega}\cdot\vec{\nabla}\psi_g^{\lambda}(\vec{r},\vec{\Omega}) + \Sigma_{tg}(\vec{r})\psi_g^{\lambda}(\vec{r},\vec{\Omega}) - \sum_{g'=1}^{G}\int_{(4\pi)}\Sigma_{s\,g'g}(\vec{r},\vec{\Omega}'\cdot\vec{\Omega})\psi_g^{\lambda}(\vec{r},\vec{\Omega}')d\Omega' \\ = \frac{1}{\lambda}\sum_{g'=1}^{G}\frac{\chi_g}{4\pi}\nu_g\Sigma_{fg'}(\vec{r})\phi_{g'}^{\lambda}(\vec{r})\,,\end{aligned} \tag{2.29}$$

where $\chi_g = (1-\beta)\chi_g^p + \sum_{k=1}^{K}\beta_k\chi_g^{d,k}$, $g = 1,\ldots,G$.

In the λ-modes problem, the dominant eigenvalue (the largest in magnitude) is referred to as k_{eff}. The k_{eff} is interpreted as the asymptotic ratio of the number

of neutrons in one generation and this number in the next generation (Lewis and Miller, 1984). The eigenfunction associated with this mode shows the distribution of the neutrons in steady-state. Clearly, the system is critical when the dominant eigenvalue is $k_{\text{eff}} = 1$. For values of $k_{\text{eff}} < 1$, the system would be subcritical because not enough neutrons are produced by fission. On the other hand, values of $k_{\text{eff}} > 1$ would imply that the system is supercritical.

The γ-modes were presented in (Ronen et al., 1976). There is not a lot of work devoted to this kind of modes but they have interesting spectral properties to study the criticality (Carreño et al., 2017b) and to develop modal methods (Carreño et al., 2019c). The γ-modes problem is obtained diving the fission and scattering terms of Equation (2.15) by a positive number γ as

$$\begin{aligned}\vec{\Omega}\cdot\vec{\nabla}\psi_g^\gamma(\vec{r},\vec{\Omega}) + \Sigma_{tg}(\vec{r})\psi_g^\gamma(\vec{r},\vec{\Omega}) = \frac{1}{\gamma}\sum_{g'=1}^{G}\int_{(4\pi)}\Sigma_{s,\,g'g}(\vec{r},\vec{\Omega}'\cdot\vec{\Omega})\psi_g^\gamma(\vec{r},\vec{\Omega}')d\Omega' \\ + \frac{1}{\gamma}\sum_{g'=1}^{G}\frac{\chi_g}{4\pi}\nu_g\Sigma_{fg'}(\vec{r})\phi_{g'}^\gamma(\vec{r})\,,\end{aligned} \tag{2.30}$$

where $\chi_g = (1-\beta)\chi_g^p + \sum_{k=1}^{K}\beta_k\chi_g^{d,k}$, $g = 1,\ldots,G$.

The classification of the criticality of the system according to these modes is the same as for the λ-modes. A value of $\gamma = 1$ is obtained when system is in critical state. Otherwise, the system will be in subcritical or supercritical state depending on whether $\gamma < 0$ or $\gamma > 0$, respectively.

The α-modes problem is basic in the field of nuclear reactor physics (Bell and Glasstone, 1970). These modes are important to develop monitoring techniques for subcritical systems (Lewins, 2013; Kópházi and Lathouwers, 2012; Uyttenhove et al., 2014). Recently, they are also used to decompose the neutron noise (Yamamoto and Sakamoto, 2019). These modes are obtained by assuming an exponential behaviour of the neutron flux in (2.15) as

$$\Psi_g(\vec{r},\vec{\Omega},t) = \psi_g^\alpha(\vec{r},\vec{\Omega})e^{\alpha t}. \tag{2.31}$$

Furthermore, the delayed neutron precursors are supposed to be in steady state and the intermediate α-modes are obtained. Other treatment of the neutron precursors leads to the prompt or total α-modes (Verdú et al., 2010). This

problem has the following expression

$$\begin{aligned}
&\vec{\Omega}\cdot\vec{\nabla}\psi_g^{\alpha}(\vec{r},\vec{\Omega})+\Sigma_{tg}(\vec{r})\psi_g^{\alpha}(\vec{r},\vec{\Omega})\\
&-\sum_{g'=1}^{G}\int_{(4\pi)}\Sigma_{s,\,g'g}(\vec{r},\vec{\Omega}'\cdot\vec{\Omega})\psi_g^{\alpha}(\vec{r},\vec{\Omega}')d\Omega'\\
&-\sum_{g'=1}^{G}\frac{\chi_g}{4\pi}\nu_g\Sigma_{fg'}(\vec{r})\phi_{g'}^{\alpha}(\vec{r})=\alpha\frac{1}{v_g}\psi_g^{\alpha}(\vec{r},\vec{\Omega})\,,
\end{aligned}\tag{2.32}$$

where $\chi_g=(1-\beta)\chi_g^p+\sum_{k=1}^{K}\beta_k\chi_g^{d,k}$, $g=1,\ldots,G$.

From the physical definition of criticality, we have that the system will be critical if $\alpha=0$. For values where $\alpha>0$ or $\alpha<0$ a supercritical or subcritical system is obtained, respectively.

2.4 The neutron diffusion equation

The neutron transport equation, even if the multigroup approximation is considered, is a challenging problem that cannot be solved explicitly with deterministic methods. Only approximated forms of the transport equation are solved that are classified by the angular dependence treatment. The approximation most commonly used over the years is the neutron diffusion equation. It removes the angular dependence through the development of the flux and scattering cross-section in spherical harmonics in the first approximation, P_1. It is assumed that the neutron current is proportional to the gradient of the scalar neutron flux with a diffusion coefficient as

$$\vec{J}_g(\vec{r},t)=-D_g\vec{\nabla}\Phi_g(\vec{r},t),\tag{2.33}$$

where $\vec{J}_g$ is the current vector for the group g. This approximation is known as the Fick's first law. Moreover, it can assume that the neutron velocities are very large and the transport cross sections are large.

Under these considerations the time-dependent multigroup neutron transport equation (Equation (2.15)) can be integrated over all directions to obtain the

time-dependent multigroup neutron diffusion equation (Lewis and Miller, 1984)

$$\frac{1}{v_g}\frac{\partial \Phi_g}{\partial t}(\vec{r},\, t) = \vec{\nabla}\cdot\left(D_g\vec{\nabla}\,\Phi_g(\vec{r},\, t)\right) - \Sigma_{rg}(\vec{r},\, t)\,\Phi_g(\vec{r},\, t) + \sum_{\substack{g'=1\\ g'\neq g}}^{G}\Sigma_{s\,g'g}(\vec{r},\, t)\,\Phi_{g'}(\vec{r},\, t)$$
$$+(1-\beta)\chi_g^p\sum_{g'=1}^{G}\nu_g\Sigma_{fg'}(\vec{r},\, t)\,\Phi_{g'}(\vec{r},\, t) + \sum_{k=1}^{K}\lambda_k^d\chi_g^{d,k}\,\mathcal{C}_k(\vec{r},\, t), \qquad g=1,\,\ldots,\,G, \tag{2.34}$$

where the concentration of neutron precursors is written in the form

$$\frac{\partial \mathcal{C}_k}{\partial t}(\vec{r},\, t) = \beta_k\sum_{g=1}^{G}\nu_g\Sigma_{fg}(\vec{r},\, t)\Phi_g(\vec{r},\, t) - \lambda_k^d\mathcal{C}_k(\vec{r},\, t), \qquad k=1,\ldots,K. \tag{2.35}$$

In this notation, the removal cross-section (Σ_{rg}) is introduced. It is defined as

$$\Sigma_{rg} = \Sigma_{ag} + \sum_{\substack{g'=1\\ g'\neq g}}^{G}\Sigma_{sgg'}, \qquad g=1,\ldots,G.$$

In a matrix form, this equation can be also expressed as

$$\begin{aligned}\mathcal{V}^{-1}\frac{\partial\Phi}{\partial t} + \mathcal{L}\Phi + \mathcal{S}\Phi &= (1-\beta)\mathcal{F}\Phi + \sum_{p=1}^{N_p}\lambda_k^d\chi_g^{d,k}\mathcal{C}_k,\\ \frac{\mathrm{d}\mathcal{C}_k}{\mathrm{d}t} &= \beta_k\mathcal{F}_1\Phi - \lambda_k^d\mathcal{C}_k, \qquad k=1,\ldots,K,\end{aligned} \tag{2.36}$$

where the neutron flux is

$$\Phi := \Phi(\vec{r},t) = \begin{pmatrix}\Phi_1 & \Phi_2 & \cdots & \Phi_G\end{pmatrix}^{\mathsf{T}},$$

and the matrix operators are defined as

$$\mathcal{L} = \begin{pmatrix} -\vec{\nabla}(D_1\vec{\nabla}) + \Sigma_{r1} & 0 & \cdots & 0 \\ 0 & -\vec{\nabla}(D_2\vec{\nabla}) + \Sigma_{r2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\vec{\nabla}(D_G\vec{\nabla}) + \Sigma_{rG} \end{pmatrix},$$

$$\mathcal{S} = \begin{pmatrix} 0 & -\Sigma_{s21} & \cdots & -\Sigma_{sG1} \\ -\Sigma_{s12} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\Sigma_{sG,G-1} \\ -\Sigma_{s1G} & \cdots & -\Sigma_{sG-1,G} & 0 \end{pmatrix},$$

$$\mathcal{F} = \begin{pmatrix} \chi_1^p\nu_1\Sigma_{f1} & \chi_1^p\nu_2\Sigma_{f2} & \cdots & \chi_1^p\nu_G\Sigma_{fG} \\ \chi_2^p\nu_1\Sigma_{f1} & \chi_2^p\nu_2\Sigma_{f2} & \ddots & \chi_2^p\nu_G\Sigma_{fG} \\ \vdots & \ddots & \ddots & \vdots \\ \chi_G^p\nu_1\Sigma_{f1} & \chi_G^p\nu_2\Sigma_{f2} & \cdots & \chi_G^p\nu_G\Sigma_{fG} \end{pmatrix}, \quad \mathcal{V}^{-1} = \begin{pmatrix} 1/v_1 & 0 & \cdots & 0 \\ 0 & 1/v_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1/v_G \end{pmatrix},$$

$$\chi_g^d = \begin{pmatrix} \chi_g^{d,1} & \chi_g^{d,2} & \cdots & \chi_g^{d,K} \end{pmatrix}^{\mathsf{T}}, \qquad \mathcal{F}_1 = \begin{pmatrix} \nu_1\Sigma_{f1} & \nu_2\Sigma_{f2} & \cdots & \nu_G\Sigma_{fG} \end{pmatrix}. \tag{2.37}$$

In neutron diffusion computations, it is usual to utilize the two energy groups approximation, where the energy is divided into a fast group ($g = 1$), corresponding to the neutrons whose energy is above 0.625 MeV, and a thermal group ($g = 2$), corresponding to the neutrons whose energy is smaller than the previous ones (Stacey, 1969). Moreover, it is supposed that there is not up-scattering, i.e. $\Sigma_{s21} = 0$ and there is not neutron production in the thermal group, i.e. $\chi_2^p = 0 = \chi_{k,2}^d, \forall k$. As a consequence of these assumptions and using $\chi = \chi^p = \chi^d$ for this case yields to

$$\mathcal{V}^{-1}\frac{\partial\Phi}{\partial t} + \mathcal{L}\Phi + \mathcal{S}\Phi = (1-\beta)\mathcal{F}\Phi + \sum_{k=1}^{K}\lambda_k^d\chi\mathcal{C}_k \quad , \tag{2.38}$$

$$\frac{\partial\mathcal{C}_p}{\partial t} = \beta_k\left(\nu\Sigma_{f1}\ \nu\Sigma_{f2}\right)\Phi - \lambda_k^d\mathcal{C}_k\,, \quad k = 1,\ldots,K, \tag{2.39}$$

where

$$\mathcal{L} = \begin{pmatrix} -\vec{\nabla}\cdot(D_1\vec{\nabla}) + \Sigma_{a1} + \Sigma_{s12} & 0 \\ 0 & -\vec{\nabla}\cdot(D_2\vec{\nabla}) + \Sigma_{a2} \end{pmatrix}, \quad \mathcal{V}^{-1} = \begin{pmatrix} \frac{1}{\mathrm{v}_1} & 0 \\ 0 & \frac{1}{\mathrm{v}_2} \end{pmatrix},$$

$$\mathcal{S} = \begin{pmatrix} 0 & 0 \\ -\Sigma_{s12} & 0 \end{pmatrix}, \quad \mathcal{F} = \begin{pmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{pmatrix}, \quad \Phi = \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix}, \quad \chi = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Diffusion theory provides a valid description of the neutron flux when three main assumptions are satisfied: the absorption is much less likely than scattering, the neutron distribution is spatially linear and the scattering is isotropic. The first condition is satisfied for most of the moderating and structural materials found in a nuclear reactor but not for the fuel and control elements. The second one is satisfied a few mean free paths away from the boundary of large homogeneous media with relatively uniform source distributions. The third condition is satisfied for scattering from heavy atomic mass nuclei (Stacey, 2018).

However, a realistic nuclear reactor is composed of thousands of elements, many of them highly absorbent. Thus, diffusion theory is not strictly valid. Nevertheless, diffusion theory is widely used in nuclear reactor analysis and makes accurate predictions. To do that, the many small elements in one large region are substituted by a homogenized mixture with effective averaged cross sections and diffusion coefficients, taking a computational model for which diffusion theory is valid Stacey, 2018. At pin level, neutron diffusion theory is not valid. The strategy to study these problems is using a more accurate transport approximations where the diffusion theory would be expected to fail.

2.4.1 Boundary conditions for the neutron diffusion equation

The conditions considered at boundary of the reactor domain $\Gamma =: \partial V$ for the neutron diffusion equation are:

- **Albedo boundary conditions**, that are of the form

$$\vec{n}\vec{\nabla}\Phi_g(\vec{r}_0, t) + \frac{1}{D_g}\frac{1}{2}\frac{1-\beta_{\mathrm{al}}}{1+\beta_{\mathrm{al}}}\Phi_g(\vec{r}_0, t) = 0\,, \qquad \vec{r}_0 \in \Gamma\,. \tag{2.40}$$

  where $\vec{n}$ is an outgoing normal vector to the boundary, β_{al} is the albedo factor. This value going from $\beta_{\mathrm{al}} = 0$, leading to vacuum boundary conditions, to $\beta_{\mathrm{al}} = 1$, giving zero-current boundary conditions.

- **Vacuum boundary conditions** can be written as

$$\vec{n} \cdot D_g \vec{\nabla} \Phi_g(\vec{r}_0, t) = \frac{1}{2} \Phi_g(\vec{r}_0, t)\,, \qquad \vec{r}_0 \in \Gamma\,. \tag{2.41}$$

- **Zero-current boundary conditions** are expressed as

$$\vec{n} \cdot D_g \vec{\nabla} \Phi_g(\vec{r}_0, t) = 0\,, \qquad \vec{r}_0 \in \Gamma\,. \tag{2.42}$$

- **Zero-flux boundary conditions** are assumed if the flux on the boundary is exactly fixed to zero. This simply is written as

$$\Phi_g(\vec{r}_0, t) = 0\,, \qquad \vec{r}_0 \in \Gamma\,.$$

2.5 Spatial modes associated with the neutron diffusion equation

Analogously to the multigroup neutron transport equation in Section 2.3, the multigroup neutron diffusion equation (Equation (2.36)) can be transformed into several eigenvalue problems by forcing the criticality of the system.

The λ-modes problem is obtained by dividing the fission nuclear cross sections by a positive number, λ, as

$$(\mathcal{L} + \mathcal{S})\phi_m^\lambda = \frac{1}{\lambda_m} \mathcal{F} \phi_m^\lambda. \tag{2.43}$$

If, now, the fission and scattering terms of Equation (2.36) are divided by $\gamma > 0$ to obtain the steady-state equations, the γ-modes problem is

$$\mathcal{L}\phi_m^\gamma = \frac{1}{\gamma_m} (\mathcal{F} - \mathcal{S})\phi_m^\gamma. \tag{2.44}$$

Finally, it is considered again the neutron diffusion equation (2.36) with the delayed neutron precursors in steady state, i.e.

$$0 = \beta_k \mathcal{F}_1 \phi - \lambda_k^d \mathcal{C}_k, \qquad k = 1, \ldots, K. \tag{2.45}$$

And it is supposed that the neutron flux admits a factorization

$$\Phi(\vec{r}, t) = e^{\alpha t} \phi^\alpha(\vec{r}), \tag{2.46}$$

to obtain the intermediate α-modes problem associated with the neutron diffusion equation

$$(\mathcal{F} - \mathcal{L} - \mathcal{S})\phi_m^\alpha = \alpha_m \mathcal{V}^{-1}\phi_m^\alpha. \tag{2.47}$$

Other type of α-modes such as the prompt or the total α-modes can be studied (Verdú et al., 2010). In this thesis, the simple term α-modes is used to call to the intermediate α-modes.

2.5.1 The adjoint spatial problems

Associated with each spatial problem, one can formulate an adjoint problem (Henry, 1975). In the case of the λ-modes problem, we define the adjoint problem as

$$(\mathcal{L}^\dagger + \mathcal{S}^\dagger)\phi_l^{\lambda,\dagger} = \frac{1}{\lambda_l}\mathcal{F}^\dagger\phi_l^{\lambda,\dagger}, \tag{2.48}$$

where $\mathcal{L}^\dagger$, $\mathcal{S}^\dagger$ and $\mathcal{F}^\dagger$ are the transpose operators of $\mathcal{L}$, $\mathcal{S}$ and $\mathcal{F}$, respectively.

The λ-modes, ϕ_m^λ, and the adjoint λ-modes, $\phi_l^{\lambda,\dagger}$, satisfy the biorthogonality relation

$$\langle \phi_l^{\lambda,\dagger}, \mathcal{F}\phi_m^\lambda \rangle = \int_V (\phi_l^{\lambda,\dagger})^\intercal \mathcal{F}\phi_m^\lambda \, dV = \delta_{l,m}\langle \phi_m^{\lambda,\dagger}, \mathcal{F}\phi_m^\lambda \rangle, \quad \forall\, l, m = 1, \ldots, q \tag{2.49}$$

where V is the volume defined by the reactor core, $\delta_{l,m}$ is the Kronecker's delta and q the total number of modes.

Likewise, we define the adjoint problem for the γ-modes as

$$\mathcal{L}^\dagger\phi_l^{\gamma,\dagger} = \frac{1}{\gamma_l}(\mathcal{F}^\dagger - \mathcal{S}^\dagger)\phi_l^{\gamma,\dagger}. \tag{2.50}$$

In this case, the adjoint γ-modes, $\phi_l^{\gamma,\dagger}$, satisfy the biorthogonality condition

$$\langle \phi_l^{\gamma,\dagger}, (\mathcal{F} - \mathcal{S})\phi_m^\gamma \rangle = \langle \phi_m^{\gamma,\dagger}, (\mathcal{F} - \mathcal{S})\phi_m^\gamma \rangle \delta_{l,m}, \quad \forall\, l, m = 1, \ldots, q. \tag{2.51}$$

Lastly, for the α-modes, we introduce the adjoint problem

$$(\mathcal{F}^\dagger - \mathcal{L}^\dagger - \mathcal{S}^\dagger)\phi_l^{\alpha,\dagger} = \alpha_l \mathcal{V}^{-1}\phi_l^{\alpha,\dagger}. \tag{2.52}$$

Note that, the symmetry of the $\mathcal{V}^{-1}$ operator implies that $\mathcal{V}^{-1,\dagger} = \mathcal{V}^{-1}$. In that case, the adjoint α-modes $\phi_l^{\alpha,\dagger}$, satisfy the biorthogonality condition

$$\langle \phi_l^{\alpha,\dagger}, \mathcal{V}^{-1}\phi_m^\alpha \rangle = \langle \phi_m^{\alpha,\dagger}, \mathcal{V}^{-1}\phi_m^\alpha \rangle \delta_{l,m}, \quad \forall\, l, m = 1, \ldots, q. \tag{2.53}$$

2.5.2 Relation between the spatial modes

The definition of the adjoint λ-modes problem can establish a relation between the λ-modes and the γ-modes. For that purpose, the γ-modes problem (Equation (2.44)) are multiplied by the adjoint λ-mode, $\phi_m^{\lambda,\dagger}$ and it is integrated over the domain. It yields

$$\langle \phi_m^{\lambda,\dagger}, \mathcal{L}\phi_m^{\gamma} \rangle = \langle \phi_m^{\lambda,\dagger}, \frac{1}{\gamma_m}(\mathcal{F} - \mathcal{S})\phi_m^{\gamma} \rangle, \tag{2.54}$$

or equivalently, by the symmetry of $\mathcal{L}$,

$$\langle \mathcal{L}\phi_m^{\lambda,\dagger}, \phi_m^{\gamma} \rangle = \langle \phi_m^{\lambda,\dagger}, \frac{1}{\gamma_m}(\mathcal{F} - \mathcal{S})\phi_m^{\gamma} \rangle. \tag{2.55}$$

Taking the reordering of Equation (2.48),

$$\mathcal{L}\phi_m^{\lambda,\dagger} = \frac{1}{\lambda_m}\mathcal{F}^{\dagger}\phi_m^{\lambda,\dagger} - \mathcal{S}^{\dagger}\phi_m^{\lambda,\dagger}, \tag{2.56}$$

the Equation (2.55) is equivalent to

$$\frac{1}{\lambda_m}\langle \mathcal{F}^{\dagger}\phi_m^{\lambda,\dagger}, \phi_m^{\gamma} \rangle = \frac{1}{\gamma_m}\langle \phi_m^{\lambda,\dagger}, (\mathcal{F} - \mathcal{S})\phi_m^{\gamma} \rangle + \langle \phi_m^{\lambda,\dagger}, \mathcal{S}\phi_m^{\gamma} \rangle. \tag{2.57}$$

Simplifying and isolating λ_m from Equation (2.57) gives

$$\frac{1}{\lambda_m} = \frac{1}{\gamma_m} + \left(1 - \frac{1}{\gamma_m}\right)\frac{\langle \phi_m^{\lambda,\dagger}, \mathcal{S}\phi_m^{\gamma} \rangle}{\langle \phi_m^{\lambda,\dagger}, \mathcal{F}\phi_m^{\gamma} \rangle}. \tag{2.58}$$

Likewise, the relation between the α-modes and the λ-modes is given in (Verdú et al., 2010)

$$\alpha_m = \left(1 - \frac{1}{\lambda_m}\right)\frac{\langle \phi_m^{\lambda,\dagger}, \mathcal{F}\phi_m^{\alpha} \rangle}{\langle \phi_m^{\lambda,\dagger}, \mathcal{V}^{-1}\phi_m^{\alpha} \rangle}. \tag{2.59}$$

2.6 The Simplified Spherical harmonic equations (SP_N)

The accuracy of the diffusion theory to describe the neutron distribution inside of reactor core is restricted to some situations. If the system presents a strong material and/or flux gradients, if the neutron streaming is significant or if neutron scattering has a strongly anisotropic component, the diffusion approximation is not accurate. These cases generally occur if there are complex fuel assemblies or

if an pin level calculation is carried out. For these reasons, more accurate approximations of the neutron transport equation to improve the results of diffusion theory are used by means of the simplified spherical harmonic equations. In this thesis, we simply study the solution of the λ-modes problem associated with the SP_N equations. The derivation presented follows the Gelbard method (Gelbard, 1960) where the one-dimensional P_N equations are extended to multidimensional geometries substituting the one-dimensional derivatives by a multidimensional gradient. Because of that, only P_N equations for a slab geometry are presented. More details about the multidimensional P_N equations and their convergence can be found in (Hébert, 2006).

2.6.1 The P_N equations

The λ-modes problem associated with the one-dimensional multigroup neutron transport equation can be written as

$$(\mu\frac{\mathrm{d}}{\mathrm{d}x}+\Sigma_{tg}(x))\psi_g^\lambda(x,\mu)-\sum_{g'=1}^{G}\int_{-1}^{1}\Sigma_{sgg'}(x,\mu_0)\psi_{g'}^\lambda(x,\mu')\mathrm{d}\mu' = \frac{1}{\lambda}\sum_{g'=1}^{G}\frac{\chi_g(x)}{2}\nu_g\Sigma_{fg'}(x)\int_{-1}^{1}\psi_{g'}^\lambda(x,\mu')\mathrm{d}\mu',$$

with $g=1,\ldots,G$, $x\in[0,\ L_t]$ and vacuum boundary conditions

$$\psi_g^\lambda(0,\mu_{in})=0=\psi_g^\lambda(L_t,\mu_{in})\,.$$

The coefficient μ is equal to $\mu\equiv cos(\theta)$ where θ is the angle between the direction of the incident neutron velocity and the x axis. Likewise, it is defined the coefficient $\mu_0\equiv cos(\theta_0)$ where θ_0 denotes the angle between the incident neutrons and the scattered neutrons. Finally, μ_{in} denotes the set of directions cosines that are incident at a given boundary, i.e., at $x=0$, $0<\mu_{in}\leq 1$ and at $x=L_t$, $-1\leq\mu_{in}<0$.

The P_N approximation assumes that the angular dependence of the angular neutron flux and the scattering cross-section can be expanded in terms of $N+1$

Legendre polynomials (N odd), $P_n(\mu)$, as

$$\psi_g^{\lambda}(x,\mu) = \sum_{n=0}^{N} \frac{2n+1}{2} \phi_g^n(x)\, P_n(\mu)\ , \tag{2.60}$$

$$\Sigma_{sgg'}(x,\mu_0) = \sum_{n=0}^{L} \frac{2n+1}{2} \Sigma_{sgg'}^n(x)\, P_n(\mu_0)\ . \tag{2.61}$$

where ϕ_g^n is the n-th angular moment of the neutron flux of energy group g and $\Sigma_{sgg'}^n$ is the n-th scattering cross sections moment. The expansions (2.60) and (2.61) are substituted into equation (2.60) and the orthogonality relations for these polynomials are taken into account, to obtain the $\mathrm{P_N}$ equations (Capilla et al., 2005),

$$\begin{aligned} \frac{\mathrm{d}\phi_g^1}{\mathrm{d}x} + \sum_{g'=1}^{G} \left(\Sigma_{tg} - \Sigma_{sgg'}^0\right) \phi_{g'}^0 &= \frac{1}{\lambda} \sum_{g'=1}^{G} \chi_g \nu_{g'} \Sigma_{fg'} \phi_{g'}^0, \\ \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{n}{2n+1}\phi_g^{n-1} + \frac{n+1}{2n+1}\phi_g^{n+1}\right) + \sum_{g'=1}^{G} (\delta_{gg'}\Sigma_{tg} - \Sigma_{sgg'}^n)\phi_{g'}^n &= 0\,, \\ \text{for } \ n = 1,\ldots,N\,, \quad g = 1,\ldots,G. & \end{aligned} \tag{2.62}$$

In this formulation, the expansion order for the angular flux, N, is considered to be larger than the order of anisotropic scattering, L. Moreover, the components of the scattering are assumed to be equal to zero for moments higher than L, but they have been maintained for simplicity in the formulation. The $\mathrm{P_N}$ equations (2.62) are composed of $N+1$ equations with $N+2$ unknowns. This fact can be solved by imposing that the derivative of the highest order moment to zero $\frac{\mathrm{d}}{\mathrm{d}x}\phi^{N+1} = 0$. This assignment is the most common and straightforward one, but in some time dependent computations, can be problematic. Thus other assignments have also been studied in the literature (Hauck and McClarren, 2010).

In matrix form, the system of equations (2.62) can be expressed as (Hamilton and Evans, 2015)

$$\begin{aligned} \frac{\mathrm{d}\phi^1}{\mathrm{d}x} + \mathbf{\Sigma^0}\phi^0 &= \frac{1}{\lambda}\mathbf{F}\phi^0\,, \\ \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{n}{2n+1}\phi^{n-1} + \frac{n+1}{2n+1}\phi^{n+1}\right) + \mathbf{\Sigma^n}\phi^n &= 0\,, \qquad n = 1,\ldots,N. \end{aligned} \tag{2.63}$$

where,

$$\boldsymbol{\Sigma}^{\mathbf{n}} = \begin{pmatrix} \Sigma_t^1 - \Sigma_{s11}^n & -\Sigma_{s12}^n & \dots & -\Sigma_{s1G}^n \\ -\Sigma_{s21}^n & \Sigma_{t2} - \Sigma_{s22}^n & \dots & -\Sigma_{s2G}^n \\ \vdots & \vdots & \ddots & \vdots \\ -\Sigma_{sG1}^n & -\Sigma_{sG2}^n & \dots & \Sigma_{tG} - \Sigma_{sGG}^n \end{pmatrix},$$

$$\mathbf{F} = \begin{pmatrix} \chi_1\nu_1\Sigma_{f1} & \chi_1\nu_2\Sigma_{f2} & \dots & \chi_1\nu_G\Sigma_{fG} \\ \chi_2\nu_1\Sigma_{f1} & \chi_2\nu_2\Sigma_{f2} & \dots & \chi_2\nu_G\Sigma_{fG} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_G\nu_1\Sigma_{f1} & \chi_G\nu_2\Sigma_{f2} & \dots & \chi^G\nu_G\Sigma_{fG} \end{pmatrix},$$

$$\phi^n = \left(\phi_1^n, \phi_2^n, \dots, \phi_G^n\right)^T .$$

Therefore, if the equations related to the odd moments of the flux are substituted into equation (2.63) one have

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{n\left(\boldsymbol{\Sigma}^{\mathbf{n-1}}\right)^{-1}}{(2n+1)(2n-1)}\frac{\mathrm{d}}{\mathrm{d}x}\left((n-1)\phi^{n-2} + n\phi^n\right)\right.$$
$$\left. + \frac{(n+1)\left(\boldsymbol{\Sigma}^{\mathbf{n+1}}\right)^{-1}}{(2n+1)(2n+3)}\frac{\mathrm{d}}{\mathrm{d}x}\left((n+1)\phi^n + (n+2)\phi^{n+2}\right)\right) + \boldsymbol{\Sigma}^{\mathbf{n}}\phi^n = \frac{1}{\lambda}\mathbf{F}\phi^n\delta_{n0}\,,$$
$$n = 0, 2, \dots, N-1\,. \tag{2.64}$$

For instance, the set of P_3 equations are

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{3}\left(\boldsymbol{\Sigma}^{\mathbf{1}}\right)^{-1}\frac{\mathrm{d}}{\mathrm{d}x}\left(\phi^0 + 2\phi^2\right)\right) + \boldsymbol{\Sigma}^{\mathbf{0}}\phi^0 = \frac{1}{\lambda}\mathbf{F}\phi^0\,,$$
$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{2}{15}\left(\boldsymbol{\Sigma}^{\mathbf{1}}\right)^{-1}\frac{\mathrm{d}}{\mathrm{d}x}\left(\phi^0 + 2\phi^2\right) + \frac{3}{35}\left(\boldsymbol{\Sigma}^{\mathbf{3}}\right)^{-1}\frac{\mathrm{d}}{\mathrm{d}x}3\phi^2\right) + \boldsymbol{\Sigma}^{\mathbf{2}}\phi^2 = 0\,. \tag{2.65}$$

On the other hand, the set of P_5 equations are

$$\begin{aligned}
-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1}{3}(\boldsymbol{\Sigma^1})^{-1}\frac{\mathrm{d}}{\mathrm{d}x}\left(\phi^0+2\phi^2\right)\right)+\boldsymbol{\Sigma^0}\phi^0&=\frac{1}{\lambda}\mathbf{F}\phi^0\,,\\
-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{2}{15}(\boldsymbol{\Sigma^1})^{-1}\frac{\mathrm{d}}{\mathrm{d}x}\left(\phi^0+2\phi^2\right)+\frac{3}{35}(\boldsymbol{\Sigma^3})^{-1}\frac{\mathrm{d}}{\mathrm{d}x}\left(3\phi^2+4\phi^4\right)\right)+\boldsymbol{\Sigma^2}\phi^2&=0\,,\\
-\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{4}{63}(\boldsymbol{\Sigma^3})^{-1}\frac{\mathrm{d}}{\mathrm{d}x}\left(3\phi^2+4\phi^4\right)+\frac{5}{99}(\boldsymbol{\Sigma^5})^{-1}\frac{\mathrm{d}}{\mathrm{d}x}5\phi^4\right)+\boldsymbol{\Sigma^4}\phi^4&=0\,.
\end{aligned} \tag{2.66}$$

The Equation (2.64) defines an eigenvalue problem associated with a linear system of $(N+1)/2$ elliptic, second-order equations. This problem can be transformed into a problem composed of a set of $M=(N+1)/2$ diffusion-like equations if the following linear change of variables is considered

$$U^m=(2m-1)\,\phi^{2m-2}+2m\,\phi^{2m},\qquad m=1,2,\ldots,M-1\,, \tag{2.67}$$

$$U^M=(2M-1)\,\phi^{2M-2}, \tag{2.68}$$

where U^m contains the group dependent diffusive pseudo-moments

$$U^m=(u_1^m,\,u_2^m,\,\ldots,\,u_G^m)^{\mathsf{T}}\,. \tag{2.69}$$

The system obtained would have the following form

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(\mathbb{D}\frac{\mathrm{d}}{\mathrm{d}x}U\right)+\mathbb{A}U=\frac{1}{\lambda}\mathbb{F}U\,. \tag{2.70}$$

As an example, in the case of the P_3 equations, the change of variables is

$$U^1=\phi^0+2\phi^2\,,\quad U^2=3\phi^2\,, \tag{2.71}$$

such that

$$U=\left(U^1,\,U^2\right)^{\mathsf{T}}\,.$$

In the system (2.70), the effective diffusion matrix, $\mathbb{D}$, the absorption matrix, $\mathbb{A}$, and the fission matrix, $\mathbb{F}$, are given by

$$\mathbb{D}=\begin{pmatrix}\frac{1}{3}(\boldsymbol{\Sigma^1})^{-1} & 0\\ 0 & \frac{1}{7}(\boldsymbol{\Sigma^3})^{-1}\end{pmatrix},\quad \mathbb{A}_{ij}=\sum_{m=1}^{2}\mathbf{c}_{ij}^{(\mathbf{m})}\boldsymbol{\Sigma^m},\quad \mathbb{F}_{ij}=\mathbf{c}_{ij}^{(\mathbf{1})}\mathbf{F}, \tag{2.72}$$

and the coefficients matrix, $\mathbf{c^{(m)}}$,

$$\mathbf{c^{(1)}} = \begin{pmatrix} 1 & -\frac{2}{3} \\ -\frac{2}{3} & \frac{4}{9} \end{pmatrix}, \quad \mathbf{c^{(2)}} = \begin{pmatrix} 0 & 0 \\ 0 & \frac{5}{9} \end{pmatrix}, \tag{2.73}$$

For the P_5 equations, the change of variables is

$$U^1 = \phi^0 + 2\phi^2\,, \quad U^2 = 3\phi^2 + 4\phi^4\,, \quad U^3 = 5\phi^4\,, \tag{2.74}$$

and the elements of the system (2.70) are given by

$$\begin{gathered} U = \left(U^1, U^2, U^3\right)^{\mathsf{T}}, \quad \mathbb{A}_{ij} = \sum_{m=1}^{3} \mathbf{c}_{ij}^{(\mathbf{m})} \mathbf{\Sigma^m}, \quad \mathbb{F}_{ij} = \mathbf{c}_{ij}^{(\mathbf{1})} \mathbf{F}\,, \\ \mathbb{D} = \begin{pmatrix} \frac{1}{3}\left(\mathbf{\Sigma^1}\right)^{-1} & 0 & 0 \\ 0 & \frac{1}{7}\left(\mathbf{\Sigma^3}\right)^{-1} & 0 \\ 0 & 0 & \frac{1}{11}\left(\mathbf{\Sigma^5}\right)^{-1} \end{pmatrix}, \end{gathered} \tag{2.75}$$

where the coefficients matrix, $\mathbf{c^{(m)}}$ are, in this case,

$$\mathbf{c^{(1)}} = \begin{pmatrix} 1 & -\frac{2}{3} & \frac{8}{15} \\ -\frac{2}{3} & \frac{4}{9} & -\frac{16}{45} \\ -\frac{8}{15} & -\frac{16}{45} & \frac{64}{225} \end{pmatrix}, \quad \mathbf{c^{(2)}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{5}{9} & -\frac{4}{9} \\ 0 & -\frac{4}{9} & \frac{16}{45} \end{pmatrix}, \quad \mathbf{c^{(3)}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{9}{25} \end{pmatrix}. \tag{2.76}$$

2.6.2 Boundary Conditions in one-dimensional P_N

In this thesis, two types of boundary conditions are considered: vacuum boundary conditions and reflective boundary conditions.

- For **vacuum boundary conditions**, we will employ the Marshak conditions (Stacey, 2018). The generalized Marshak boundary condition, for a boundary position x_0, is

$$\int_{\mu_{in}} P_n\left(\mu\right) \psi_g^{\lambda}(x_0, \mu) \mathrm{d}\mu = 0, \qquad g = 1, 2, \ldots, G, \quad n = 1, 3, \ldots, N. \tag{2.77}$$

Expanding $\psi_g^\lambda(x_0,\mu)$ in terms of Legendre polynomials and using Equation (2.60) yields to

$$\int_{\mu_{in}} P_n(\mu) \sum_{n'=0}^{N} \frac{2n'+1}{2} \phi_g^{n'}(x_0) P_{n'}(\mu) \mathrm{d}\mu = 0\,,$$
$$g = 1, 2, \ldots, G, \quad n = 1, 3, \ldots, N. \tag{2.78}$$

If we reconsider the P_3 equations, the Marshak boundary conditions are

$$\frac{1}{2}\phi^0 + \frac{5}{8}\phi^2 = -\phi^1\,,$$
$$-\frac{1}{8}\phi^0 + \frac{5}{8}\phi^2 = -\phi^3\,, \tag{2.79}$$

If now, we use the Equation (2.63), to remove the odd moments from the previous Equation, and the change of variables proposed in Equation (2.71); the vacuum boundary conditions can be applied by imposing

$$-\hat{n}\,\mathbb{D}\frac{\mathrm{d}}{\mathrm{d}x}U(x_0) = \mathbb{B}U(x_0)\,, \tag{2.80}$$

where $\hat{n}$ is the normal direction of the boundary (equal to 1 or -1 in 1D) and the matrix $\mathbb{B}$ is the result of the Kronecker product of matrix $\mathbf{b}$ by an $G \times G$ identity matrix as

$$\mathbb{B} = \mathbf{b} \otimes \mathbf{I}_{(G\times G)}\,, \qquad \mathbf{b} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{8} \\ -\frac{1}{8} & \frac{7}{24} \end{pmatrix}. \tag{2.81}$$

For the P_5 equations, the Marshak conditions are given by

$$\frac{1}{2}\phi^0 + \frac{5}{8}\phi^2 - \frac{3}{16}\phi^4 = -\phi^1\,,$$
$$-\frac{1}{8}\phi^0 + \frac{5}{8}\phi^2 - \frac{81}{128}\phi^4 = -\phi^3\,,$$
$$\frac{1}{16}\phi^0 - \frac{25}{128}\phi^2 - \frac{81}{128}\phi^4 = -\phi^5\,.$$

In the same way that for the P_3 equations, one can impose vacuum conditions (2.80), but in this case the matrix $\mathbb{B}$ is given by

$$\mathbb{B} = \mathbf{b} \otimes \mathbf{I}_{(G\times G)}\,, \qquad \mathbf{b} = \begin{pmatrix} \frac{1}{2} & -\frac{1}{8} & \frac{1}{16} \\ -\frac{1}{8} & \frac{7}{24} & -\frac{41}{384} \\ \frac{1}{16} & -\frac{41}{384} & \frac{407}{1920} \end{pmatrix}. \tag{2.82}$$

- On the other hand, **reflective boundary conditions** can be used. The only conditions that make physical sense in this case is setting all the odd moments to zero

$$\phi_g^n(x_0) = 0, \qquad g = 1, 2, \ldots, G, \quad n = 1, 3, \ldots, N\,. \tag{2.83}$$

Using the Equations (2.63), to isolate the odd-moments, and then, the change of variables (2.68), yields to reflective boundary conditions

$$\frac{\mathrm{d}}{\mathrm{d}x} u_g^m(x_0) = 0, \qquad g = 1,\ 2,\ \ldots,\ G, \quad m = 1,\ 2,\ \ldots,\ (N+1)/2\,. \tag{2.84}$$

For the P_3 equations, the conditions

$$\phi_g^1(x_0) = 0, \quad \phi_g^3(x_0) = 0, \qquad g = 1, 2, \ldots, G\,. \tag{2.85}$$

are imposed by setting

$$\frac{\mathrm{d}}{\mathrm{d}x} u_g^1(x_0) = 0, \quad \frac{\mathrm{d}}{\mathrm{d}x} u_g^2(x_0) = 0, \qquad g = 1, 2, \ldots, G. \tag{2.86}$$

In the P_5 equations, the reflective conditions

$$\phi_g^1(x_0) = 0, \quad \phi_g^3(x_0) = 0, \quad \phi_g^5(x_0) = 0 \qquad g = 1, 2, \ldots, G\,. \tag{2.87}$$

are introduced in the system by

$$\frac{\mathrm{d}}{\mathrm{d}x} u_g^1(x_0) = 0, \quad \frac{\mathrm{d}}{\mathrm{d}x} u_g^2(x_0) = 0, \quad \frac{\mathrm{d}}{\mathrm{d}x} u_g^3(x_0) = 0, \qquad g = 1, 2, \ldots, G. \tag{2.88}$$

Note that both vacuum and reflective boundary condition treatments contain asymmetric components when N is even. Thus, only odd sets of $\mathrm{P_N}$ equations are considered. Moreover, it must be noted that for each group the $\mathrm{P_N}$ system of equations (2.70) is symmetric because the coefficients $\mathbf{c^{(m)}}$ and $\mathbb{B}$ are symmetric.

2.6.3 Simplified spherical harmonics equations (SP_N)

As mentioned before, for multidimensional problems, the SP_N approximation is obtained substituting the x derivative operator of the one-dimensional P_N equations (2.64) by the corresponding two-dimensional or three-dimensional gradient operator as

$$-\vec{\nabla}\left(\frac{n\left(\boldsymbol{\Sigma}^{\mathbf{n-1}}\right)^{-1}}{(2n+1)(2n-1)}\vec{\nabla}\left((n-1)\phi^{n-2}+n\phi^{n}\right)\right.$$
$$\left.+\frac{(n+1)\left(\boldsymbol{\Sigma}^{\mathbf{n+1}}\right)^{-1}}{(2n+1)(2n+3)}\vec{\nabla}\left((n+1)\phi^{n}+(n+2)\phi^{n+2}\right)\right)+\boldsymbol{\Sigma}^{\mathbf{n}}\phi^{n}=\frac{1}{\lambda}\mathbf{F}\phi^{n}\delta_{n0}\,,$$
$$n=0,2,\ldots,N-1\,. \tag{2.89}$$

This approximation may seen ad-hoc, but in (Brantley and Larsen, 2000) a variational analysis of the SP_N equations is provided. The authors showed that these equations are high-order asymptotic solutions of the neutron transport equation when diffusion theory is the leading-order approximation as it is usual in full reactor simulations. However, the SP_N approximation does not converge to the transport solution when $N \to \infty$.

The resulting system of the SP_N equations is a set of elliptic, diffusion-like second order differential equations. In this way, these equations can be easily implemented using numerical methods suited for the diffusion equation without major changes.

Note that the SP_1 approximation is equivalent to the neutron diffusion equation. As other examples of SP_N equations, the set of SP_3 equations has the form

$$-\vec{\nabla}\left(\frac{1}{3}(\boldsymbol{\Sigma}^{\mathbf{1}})^{-1}\vec{\nabla}\left(\phi^{0}+2\phi^{2}\right)\right)+\boldsymbol{\Sigma}^{\mathbf{0}}\phi^{0}=\frac{1}{\lambda}\mathbf{F}\phi^{0}\,,$$
$$-\vec{\nabla}\left(\frac{2}{15}(\boldsymbol{\Sigma}^{\mathbf{1}})^{-1}\vec{\nabla}\left(\phi^{0}+2\phi^{2}\right)+\frac{3}{35}(\boldsymbol{\Sigma}^{\mathbf{3}})^{-1}\vec{\nabla}3\phi^{2}\right)+\boldsymbol{\Sigma}^{\mathbf{2}}\phi^{2}=0\,, \tag{2.90}$$

and the set of the SP_5 equations

$$\begin{aligned}
-\vec{\nabla}\left(\frac{1}{3}(\mathbf{\Sigma^1})^{-1}\vec{\nabla}\left(\phi^0+2\phi^2\right)\right)+\mathbf{\Sigma_0}\phi^0&=\frac{1}{\lambda}\mathbf{F}\phi^0\,,\\
-\vec{\nabla}\left(\frac{2}{15}(\mathbf{\Sigma^1})^{-1}\vec{\nabla}\left(\phi^0+2\phi^2\right)+\frac{3}{35}(\mathbf{\Sigma^3})^{-1}\vec{\nabla}\left(3\phi^2+4\phi^4\right)\right)+\mathbf{\Sigma_2}\phi^2&=0\,,\\
-\vec{\nabla}\left(\frac{4}{63}(\mathbf{\Sigma^3})^{-1}\vec{\nabla}\left(3\phi^2+4\phi^4\right)+\frac{5}{99}(\mathbf{\Sigma^5})^{-1}\vec{\nabla}\left(5\phi^4+6\Phi_6\right)\right)+\mathbf{\Sigma_4}\phi^4&=0\,.
\end{aligned} \tag{2.91}$$

Applying the same change of variables than in Equation (2.68) leads in both cases to a system of the form

$$-\vec{\nabla}\left(\mathbb{D}\,\vec{\nabla}U\right)+\mathbb{A}U=\frac{1}{\lambda}\mathbb{F}U\,, \tag{2.92}$$

where the matrix operators $\mathbb{D}$, $\mathbb{A}$ and $\mathbb{F}$ are defined, for the SP_3 case in Equations (2.72) and for the SP_5 equations in the expression (2.75).

2.6.4 Boundary Conditions in SP_N equations

Likewise as in the previous section, boundary conditions for SP_N equations are given by substituting the x derivative operator by the gradient operator in equations (2.80) and (2.84).

- **Marshak conditions** (2.77) are applied to the SP_N equations by using the same procedure that was used from the P_N equations to the SP_N boundary approximation

$$\pm\frac{\partial}{\partial x}\rightarrow\vec{\mathbf{n}}\cdot\vec{\nabla}.$$

  Under these considerations and taking into account the change of variables, the SP_N Marshak boundary conditions can imposed as

$$-\vec{n}\,\vec{\mathbb{J}}=\mathbb{B}\,U(r_0)\,, \tag{2.93}$$

  where matrix $\mathbb{B}$ is given in Equation (2.81) and Equation (2.82) for the SP_3 or SP_3 case, respectively. The vector $\vec{n}$ is the normal direction to the boundary. The current $\vec{\mathbb{J}}=(\vec{\mathbb{J}}^1,\ldots,\vec{\mathbb{J}}^{(N+1)/2})$, it is related to the flux by Fick's Law

$$\vec{\mathbb{J}}^m=-\mathbb{D}\vec{\nabla}\mathbb{U}^m\,. \tag{2.94}$$

- From the P_N boundary conditions for reflecting surfaces (Equation (2.84)), **reflective boundary conditions** for SP_N are imposed as

$$\vec{\nabla} u_g^m(r_0) = 0, \qquad g = 1, 2, \ldots, G, \qquad m = 1, 2, \ldots, (N+1)/2\,. \quad (2.95)$$

  This implies that $\vec{n} \cdot \vec{\mathbb{J}} = 0$ on the boundaries.

CHAPTER 3

FINITE ELEMENT DISCRETIZATION

The time dependent and the steady-state neutron transport approximations are differential equations that need to be spatially discretized. Different methods have been proposed to discretize the spatial modes problems associated with the neutron diffusion equation or other approximations for the neutron transport equation.

Classically, finite difference methods (FDM) (Hébert, 2009) are used to make homogenized assembly level computations with structured meshes where the derivatives are substituted by finite differences approximations. This is a simple tool, but it requires a discretization with a large amount of nodes in the meshes to obtain accurate results.

More sophisticated integration methods are the nodal methods. This methodology integrates over large homogenized regions known as nodes to obtain a balance with average surface currents and fluxes as unknowns. Some of them are: the Analytical Nodal Method (ANM) developed in (Smith, 1979), the Nodal Expansion Method (NEM) studied in (Finnemann, 1975; Singh et al., 2014); and the Nodal Collocation Method (NCM) developed in (Hébert, 1987). The NCM has been studied for some modal problems in (Verdú et al., 1994; Verdú et al., 2010) and for the spherical harmonics equations in (Capilla et al., 2008). The main drawback of this type of discretization is that it can be only applied in structured meshes.

On the other hand, one can consider finite volume methods (FVM) to integrate this type of approximations for any kind of spatial discretization (Theler, 2013; Bernal García, 2018). This numerical method transforms the partial differential equations representing conservation laws over differential volumes into discrete algebraic equations over finite cells or elements.

Finite element methods (FEM) have also been designed for different types of geometry to study the neutron diffusion equation. Structured meshes are used for reactors with rectangular geometry, such as PWR and BWR reactors (Vidal-Ferrandiz et al., 2014) and with hexagonal geometries for VVER reactors (Hebert, 2008; González-Pintor et al., 2009). Unstructured grid schemes have been developed to solve the diffusion problem in non standard geometries (Theler, 2013). They also have been efficiently used to compute the solution of the SP_N equations (Hamilton and Evans, 2015).

Other of the main advantages of the finite element method is the adaptivity. A code with *h*-adaptable meshes has been proposed to obtain the static configuration of a nuclear reactor core (Turcksin et al., 2010). A FEM code (`FEMFFUSION`) with a *hp*-adaptable finite element method has been implemented for the solution of the λ-modes problem for the neutron diffusion equation (Vidal-Ferrandiz et al., 2014). In this last work, the refinement of meshes and the increase of the polynomial degree of the FEM is analyzed showing that increasing the polynomial degree is a better strategy than reducing the size of the mesh when homogenized assemblies are considered.

In this thesis, a continuous Galerkin finite element method is implemented. It has been analyzed for the solution of the λ, γ and α-modes problem associated with the neutron diffusion equation and the λ-modes problem associated with the SP_N equations. The finite element method has been implemented by using the open source finite elements library `deal.II` (Bangerth et al., 2007) and the open nuclear code `FEMFFUSION` (Vidal-Ferràndiz, A. and Ginestar, D. and Verdú, G.) presented in (Vidal Ferràndiz, 2018). The library permits an implementation independent of the dimension of the and to manage different cell sizes, level of meshes and types of finite elements.

This chapter presents and studies the finite element discretization method for steady-state problems. Moreover, the spatial modes associated with the neutron diffusion equation are compared. Section 3.1 describes the type of elements used in the discretizations. Section 3.2 applies the finite element method to the two energy groups λ-modes problem. This Section includes the boundary conditions and the normalization used. Section 3.3 extends the application of the FEM to the simplified harmonics equations. Finally, Section 3.4 presents

some numerical results in several benchmark problems. This chapter rewrites some results presented in (Carreño et al., 2017b; Vidal-Ferràndiz et al., 2019).

3.1 Fundamental concepts of finite element discretization

Before we start with the spatial discretization of the equations, a briefly description of the elements that we have used in FEM is presented. First of all, the domain of the whole reactor is divided into a set of elements (or subdomains) where each element has a simple geometry and it is composed of only one material. These discrete elements are called cells and the set of all them is called mesh. The mesh defined for FEM can be structured (identified by regular connectivity) or unstructured (identified by irregular connectivity).

Nuclear reactor cores are constituted typically of 150-700 fuel assemblies. The geometry of the reactor assemblies and their configuration depends on the type of reactors. Figure 3.1 shows the reactor core geometries and assemblies for PWR and VVER reactors. Each fuel assembly is itself constituted by typically less than 300 fuel rods containing fissile nuclei (pin). Usually, diffusion computations are done such that each fuel assembly is homogenized as one cell with one material type (homogenized assembly level). Due to their geometry, PWR and BWR are homogenized by using parallelepipeds and VVER by using hexagonal prisms. In systems homogenized at pin level, where the fuel rods and the moderator are homogenized with different materials, $\mathrm{SP_N}$ approximations are used, since diffusion calculations do not provide accurate results. Circular pin cells are discretized with non-regular polygons as we will present in numerical results.

The cells in the finite element method must be mapped to the same reference cell. For quadrangular cells, we use the reference cell $[0,1]^d$, where d is the dimension of the problem. Thus, an affine transformation between the both coordinates systems is required to transform each physical element (or cell) in the system $(x,\, y)$ to the reference element $(\xi,\, \eta)$. Figure 3.2 displays an example of this mapping for a two-dimensional cell.

In two-dimensional case, the change of variables that relates physical coordinates (x, y), with the coordinates of the reference domain (ξ, η) is given by

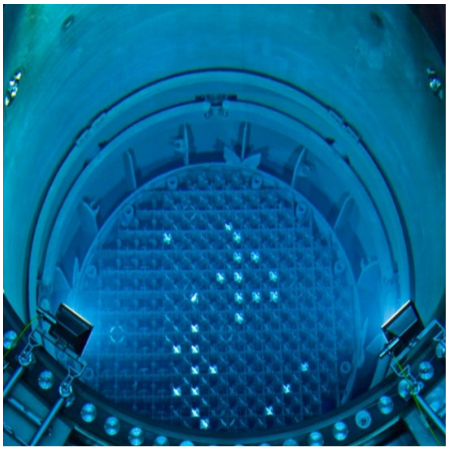
$$x(\xi,\eta) = (1-\xi)(1-\eta)x_1 + \xi(1-\eta)x_2 + \eta(1-\xi)x_3 + \xi\eta x_4\,, \tag{3.1}$$

$$y(\xi,\eta) = (1-\xi)(1-\eta)y_1 + \xi(1-\eta)y_2 + \eta(1-\xi)y_3 + \xi\eta y_4\,. \tag{3.2}$$

The affine map allows to establish the change of variables by means of the Jacobian of the transformation $|\mathbf{J^k}|$. It is necessary to compute the integrals

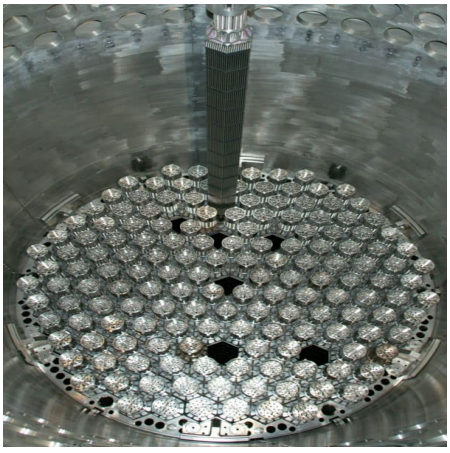
that define the matrix elements in the FEM. This relation is given by

$$\mathrm{d}x\mathrm{d}y = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} \mathrm{d}\xi\mathrm{d}\eta = |\mathbf{J^k}|\mathrm{d}\xi\mathrm{d}\eta\,. \tag{3.3}$$

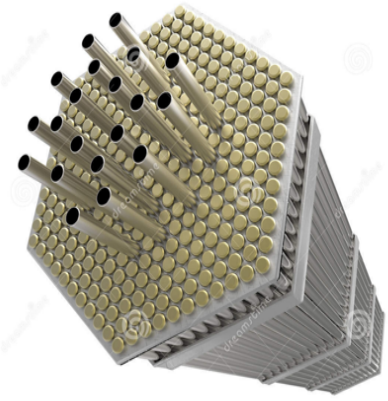


(a) PWR core

(b) PWR fuel assembly



(c) VVER core



(d) VVER fuel assembly

Figure 3.1: PWR and VVER-type reactors. Sources: www.nuclear-power.net/nuclear-power-plant/nuclear-fuel/, www.dreamstime.com

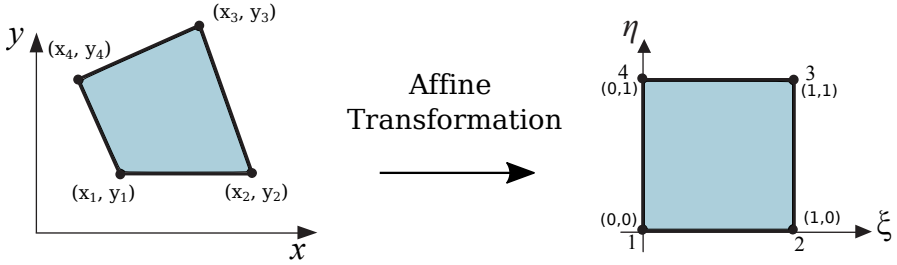


Figure 3.2: Affine transformation between the physical element and the reference element.

3.1.1 Lagrange finite elements

Lagrange finite elements are chosen (Zienkiewicz et al., 1977) for the discretization. These elements have their nodes distributed based on Gauss-Lobatto support points. Their shape functions are defined with Lagrange polynomials for every dimension. These basis functions are determined by the Kronecker delta function such that they are equal to one at the corresponding nodal point and zero at the other nodes. They satisfy all inter-element continuity conditions. Lagrange polynomials (or one-dimensional elements) can be expressed as

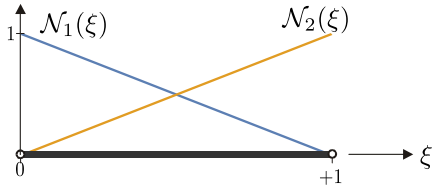
$$\mathcal{N}_i(\xi) = l_i^p(\xi) = \prod_{\substack{k=0 \\ k\neq i}}^{p+1} \frac{\xi - \xi_k}{\xi_i - \xi_k}\,, \tag{3.4}$$

where p is the degree of polynomial of the expansion which characterizes the finite element method, and ξ_i is the position of every node in the element. Multidimensional expansions of these elements are obtained by tensor product of one-dimensional elements. Thus, the two-dimensional elements are given by

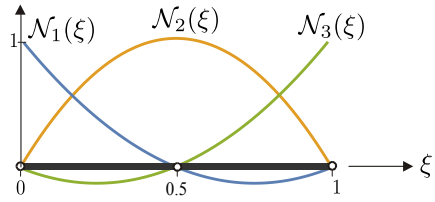
$$\mathcal{N}_{i,j}(\xi,\eta) = l_i^p(\xi) l_j^p(\eta)\,. \tag{3.5}$$

Figure 3.3 shows some examples of one-dimensional Lagrange elements. Figure 3.4 displays a two-dimensional Lagrange element. The number of nodes in each cell (known as the degrees of freedom) is determined by the order p. The degrees of freedom of the problem (N_{dofs}) are computed by multiplying the number of nodes per cell by the number of cells and removing the repeated nodes in the interface between cells.

Lastly, Gauss-Legendre quadrature is used (Golub and Welsch, 1969) to compute the integrals of the weak formulation in each cell. The degree of the quadrature is selected with $p+1$ quadrature points (in each space direction) in order to ensure an exact integration of polynomial shape functions.



$$\mathcal{N}_1(\xi) = (1-\xi)$$
$$\mathcal{N}_2(\xi) = \xi$$



$$\mathcal{N}_1(\xi) = (1-2\xi)(1-\xi)$$
$$\mathcal{N}_2(\xi) = 4\xi(1-\xi)$$
$$\mathcal{N}_3(\xi) = \xi(2\xi-1)$$

Figure 3.3: Example of unidimensional shape functions used: linear and quadratic.

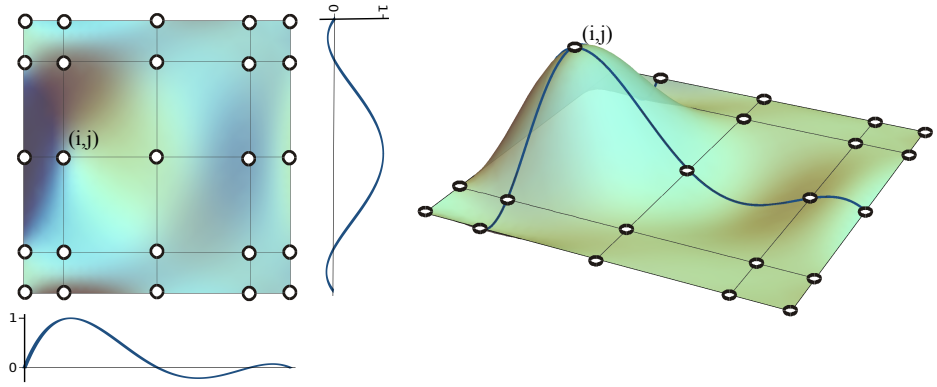


Figure 3.4: A shape function for a 2D Lagrangian element, ($i = 1$, $j = 2$, $p = 4$).

3.2 Spatial discretization for the modes problems associated with neutron diffusion equation

Let us consider the λ-modes problem for the neutron diffusion equation in the approximation of two energy groups to explain the finite element discretization method. A similar process is applied to obtain the algebraic problems associated with other approximations of the neutron transport equations. This equation

can be expressed as

$$\begin{pmatrix} -\vec{\nabla}(D_1\vec{\nabla}) + \Sigma_{a1} + \Sigma_{s12} & 0 \\ -\Sigma_{s12} & -\vec{\nabla}(D_2\vec{\nabla}) + \Sigma_{a2} \end{pmatrix} \begin{pmatrix} \phi_1^\lambda \\ \phi_2^\lambda \end{pmatrix} = \frac{1}{\lambda} \begin{pmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \phi_1^\lambda \\ \phi_2^\lambda \end{pmatrix}. \tag{3.6}$$

The weak formulation of this equation is obtained by pre-multiplying by a test function, $\varphi = (\varphi_1, \varphi_2)$, and integrating over the domain, V, that defines the reactor core as

$$\int_V \begin{pmatrix} \varphi_1 & \varphi_2 \end{pmatrix} \begin{pmatrix} -\vec{\nabla}(D_1\vec{\nabla}) + \Sigma_{a1} + \Sigma_{12} & 0 \\ -\Sigma_{s12} & -\vec{\nabla}(D_2\vec{\nabla}) + \Sigma_{a2} \end{pmatrix} \begin{pmatrix} \phi_1^\lambda \\ \phi_2^\lambda \end{pmatrix} \mathrm{d}V$$

$$= \frac{1}{\lambda} \int_V \begin{pmatrix} \varphi_1 & \varphi_2 \end{pmatrix} \begin{pmatrix} \nu\Sigma_{f1} & \nu\Sigma_{f2} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \phi_1^\lambda \\ \phi_2^\lambda \end{pmatrix} \mathrm{d}V\,. \tag{3.7}$$

Then, by taking $u\left(\vec{\nabla}\cdot\vec{\nabla}v\right) = \vec{\nabla}\cdot\left(u\vec{\nabla}v\right) - \left(\vec{\nabla}u\right)\cdot\left(\vec{\nabla}v\right)$, one can rewrite Equation (3.7) as

$$\begin{aligned} &\int_V \vec{\nabla}\varphi_1 D_1 \vec{\nabla}\phi_1^\lambda \mathrm{d}V - \int_V \vec{\nabla}\cdot\left(\varphi_1 D_1 \vec{\nabla}\phi_1^\lambda\right)\mathrm{d}V + \int_V \varphi\left(\Sigma_{a1} + \Sigma_{s12}\right)\phi_1^\lambda \mathrm{d}V \\ &+ \int_V \vec{\nabla}\varphi_2 D_2 \vec{\nabla}\phi_2^\lambda \mathrm{d}V - \int_V \vec{\nabla}\cdot\left(\varphi_2 D_2 \vec{\nabla}\phi_2^\lambda\right)\mathrm{d}V + \int_V \varphi_2 \Sigma_{a2}\phi_2^\lambda \mathrm{d}V \\ &- \int_V \varphi_2 \Sigma_{s12}\phi_1^\lambda \mathrm{d}V = \frac{1}{\lambda}\left(\int_V \varphi_1 \nu\Sigma_{f1}\phi_1^\lambda \mathrm{d}V + \int_V \varphi_1 \nu\Sigma_{f2}\phi_2^\lambda \mathrm{d}V\right). \end{aligned} \tag{3.8}$$

Now, we remove the second order derivatives by the Gauss Divergence theorem, that says, under some assumptions that normally are satisfied in reactor domains, that $\int_V \vec{\nabla}\cdot\vec{F}\mathrm{d}V = \int_\Gamma \vec{F}\mathrm{d}\vec{S}$, where $\Gamma = \partial V$ is the boundary of V. The expression obtained is

$$\begin{aligned} &\int_V \vec{\nabla}\varphi_1 D_1 \vec{\nabla}\phi_1^\lambda \mathrm{d}V - \int_\Gamma \varphi_1 D_1 \vec{\nabla}\phi_1^\lambda \mathrm{d}\vec{S} + \int_V \varphi_1\left(\Sigma_{a1} + \Sigma_{s12}\right)\phi_1^\lambda \mathrm{d}V \\ &+ \int_V \vec{\nabla}\varphi_2 D_2 \vec{\nabla}\phi_2^\lambda \mathrm{d}V - \int_\Gamma \varphi_2 D_2 \vec{\nabla}\phi_2^\lambda \mathrm{d}\vec{S} + \int_V \varphi_2 \Sigma_{a2}\phi_2^\lambda \mathrm{d}V \\ &- \int_V \varphi_2 \Sigma_{s12}\phi_1^\lambda \mathrm{d}V = \frac{1}{\lambda}\left(\int_V \varphi_1 \nu\Sigma_{f1}\phi_1^\lambda \mathrm{d}V + \int_V \varphi_1 \nu\Sigma_{f2}\phi_2^\lambda \mathrm{d}V\right). \end{aligned} \tag{3.9}$$

The last step is dividing the reactor domain V into cells or subdomains V_c such that $V = \cup_{c=1,\dots,N_c} V_c$ where it is assumed that the nuclear cross sections remain constant due to a previous spatial homogenization strategy. The cross sections for each cell c are denoted by the superscript ^{c}. Similarly, Γ_c is defined as the corresponding subdomain surface which is part of the boundary Γ. Therefore, Equation (3.9) is expressed as

$$\begin{aligned}\sum_{c=1}^{N_c} \Bigg(& D_1^c \int_{V_c} \vec{\nabla}\varphi_1 \vec{\nabla}\phi_1^\lambda \mathrm{d}V - D_1^c \int_{\Gamma_k} \varphi_1 \vec{\nabla}\phi_1^\lambda \mathrm{d}\vec{S} + (\Sigma_{a1}^c + \Sigma_{s12}^c) \int_{V_c} \varphi_1 \phi_1^\lambda \mathrm{d}V \\ & + D_2^c \int_{V_c} \vec{\nabla}\varphi_2 \vec{\nabla}\phi_2^\lambda \mathrm{d}V - D_2^c \int_{\Gamma_k} \varphi_2 \vec{\nabla}\phi_2^\lambda \mathrm{d}\vec{S} + \Sigma_{a2}^c \int_{V_c} \varphi_2 \phi_2^\lambda \mathrm{d}V \\ & - \Sigma_{s12}^c \int_{V_c} \varphi_2 \phi_1^\lambda \mathrm{d}V \Bigg) = \frac{1}{\lambda} \sum_{c=1}^{N_c} \left(\nu\Sigma_{f1}^c \int_{V_c} \varphi_1 \phi_1^\lambda \mathrm{d}V + \nu\Sigma_{f2}^c \int_{V_c} \varphi_1 \phi_2^\lambda \mathrm{d}V \right). \end{aligned} \tag{3.10}$$

Note that there are several surface integrals over the boundaries, Γ_c, that may depend on the boundary conditions. They will be studied in Section 3.2.1. To solve the integrals over the subdomains, V_c, the function ϕ_g^λ is approximated through an usual trial solution as sum of shape functions, $\mathcal{N}_a$, multiplied by the unknown expansion coefficients, $\tilde{\phi}_{ga}$,

$$\phi_g^\lambda \approx \sum_{a=0}^{N_{\mathrm{dofs}}} \mathcal{N}_a \, \tilde{\phi}_{ga} \,, \tag{3.11}$$

For test functions, continuous Galerkin approximation (Zienkiewicz et al., 1977) is used. In that sense, the test space is the same that the space defined by the basis of shape functions, that is $\mathcal{N}$.

Using these expressions in Equation (3.10) and removing redundant coefficients to obtain continuous solutions, yields to an algebraic eigenvalue problem

$$A^\lambda \tilde{\phi}^\lambda = \lambda B^\lambda \tilde{\phi}^\lambda \,, \tag{3.12}$$

where

$$A^\lambda = \begin{pmatrix} F_{11} & F_{12} \\ 0 & 0 \end{pmatrix}, \qquad B^\lambda = \begin{pmatrix} L_{11} & 0 \\ S_{21} & L_{22} \end{pmatrix}, \qquad \tilde{\phi}^\lambda = \begin{pmatrix} \tilde{\phi}_1^\lambda \\ \tilde{\phi}_2^\lambda \end{pmatrix}, \tag{3.13}$$

and the matrices elements (a, b) are given by

$$
\begin{aligned}
L_{11(ab)} &= \sum_{c=1}^{N_c} D_1^c \int_{V_k} \vec{\nabla}\mathcal{N}_a \vec{\nabla}\mathcal{N}_b \mathrm{d}V - D_1^c \int_{\Gamma_k} \mathcal{N}_a \vec{\nabla}\mathcal{N}_b \mathrm{d}\vec{S} + (\Sigma_{a1}^c + \Sigma_{12}^c) \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V \,, \\
L_{22(ab)} &= \sum_{c=1}^{N_c} D_2^c \int_{V_k} \vec{\nabla}\mathcal{N}_a \vec{\nabla}\mathcal{N}_b \mathrm{d}V - D_2^c \int_{\Gamma_k} \mathcal{N}_a \vec{\nabla}\mathcal{N}_b \mathrm{d}\vec{S} + \Sigma_{a2}^c \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V \,, \\
S_{21(ab)} &= \sum_{c=1}^{N_c} -\Sigma_{12}^c \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V \,, \\
F_{11(ab)} &= \sum_{c=1}^{N_c} \nu\Sigma_{f1}^c \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V \,, \\
F_{12(ab)} &= \sum_{c=1}^{N_c} \nu\Sigma_{f2}^c \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V \,.
\end{aligned}
\tag{3.14}
$$

These integrals only are different from zero if shape functions $\mathcal{N}_i$ and $\mathcal{N}_j$ collide inside the same cell. Therefore, sparse matrices are obtained. The Galerkin approximation also guarantees that the block matrices obtained are symmetric. Moreover, it can be proved the block matrices are also positive definite because the cross sections coefficients are always positive.

A similar process for the spatial discretization of the rest of modes problems leads to other block generalized eigenvalue problems. From the approximation of two energy groups associated with γ-modes problem, it is obtained

$$
A^\gamma \tilde{\phi}^\gamma = \gamma B^\gamma \tilde{\phi}^\gamma \,, \tag{3.15}
$$

where

$$
A^\gamma = \begin{pmatrix} F_{11} & F_{12} \\ -S_{21} & 0 \end{pmatrix} , \qquad B^\gamma = \begin{pmatrix} L_{11} & 0 \\ 0 & L_{22} \end{pmatrix} , \qquad \tilde{\phi}^\gamma = \begin{pmatrix} \tilde{\phi}_1^\gamma \\ \tilde{\phi}_2^\gamma \end{pmatrix} , \tag{3.16}
$$

and the block matrices are defined in Equation (3.14).

The α-modes discretization gives the algebraic eigenvalue problem

$$
A^\alpha \tilde{\phi}^\alpha = \hat{\alpha} B^\alpha \tilde{\phi}^\alpha \,, \tag{3.17}
$$

where

$$\hat{\alpha} = \frac{1}{\alpha}\,, \qquad A^{\alpha} = \begin{pmatrix} [V]_{11}^{-1} & 0 \\ 0 & [V]_{22}^{-1} \end{pmatrix}, \quad B^{\alpha} = \begin{pmatrix} L_{11} + F_{11} & F_{12} \\ -S_{21} & L_{22} \end{pmatrix}, \tag{3.18}$$

$$\tilde{\phi}^{\alpha} = \begin{pmatrix} \tilde{\phi}_1^{\alpha} \\ \tilde{\phi}_2^{\alpha} \end{pmatrix}, \quad [V]_{11} = v_1 M\,, \qquad [V]_{22} = v_2 M\,,$$

and the rest of block matrices are defined in Equation (3.14). The matrix M, referred to as the mass matrix, is defined as

$$M_{(ab)} = \sum_{c=1}^{N_c} \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V\,. \tag{3.19}$$

That is different to the identity matrix because the basis of Legendre polynomials is not orthonormal.

In the following, as an abuse of the notation, we denote the algebraic fluxes $\tilde{\phi^{\lambda}}$, $\tilde{\phi^{\gamma}}$, $\tilde{\phi^{\alpha}}$ with the same expressions as the continuous fluxes ϕ^{λ}, ϕ^{γ}, ϕ^{α}, respectively, by removing the tildes from the original notation. The superindex ^{c} is also removed from the cross-section notation.

3.2.1 Boundary conditions

The boundary conditions considered for the neutron diffusion equation are albedo, vacuum, zero-current and zero-flux boundary conditions.

The albedo boundary conditions are of the form,

$$\vec{n}\vec{\nabla}\phi_g^{\delta}(\vec{r}_0) + \frac{1}{D_g}\frac{1}{2}\left(\frac{1-\beta_{\mathrm{al}}}{1+\beta_{\mathrm{al}}}\right)\phi_g^{\delta}(\vec{r}_0) = 0\,, \qquad \vec{r}_0 \in \Gamma\,, \quad \delta = \lambda, \gamma, \alpha\,, \tag{3.20}$$

where $\vec{n}$ is a outgoing normal vector to the boundary, β_{al} is the albedo factor going from 0, leading to vacuum boundary conditions, to 1, giving zero-current boundary conditions.

Albedo boundary conditions are treated in a weak form by pre-multiplying the condition by the test function and integrating over the surface of the domain as

$$-D_g \int_{\Gamma} \varphi_g \vec{\nabla}\phi_g^{\delta} \mathrm{d}\vec{S} = \frac{1}{2}\frac{1-\beta_{\mathrm{al}}}{1+\beta_{\mathrm{al}}} \int_{\Gamma} \varphi_g \phi_g^{\delta} \mathrm{d}\vec{S}\,. \tag{3.21}$$

Henceforth, the integrals related to the boundaries that appear in equation (3.14) are substituted by,

$$\sum_{c=1}^{N_c} -D_g \int_{\Gamma_c} \mathcal{N}_a \vec{\nabla} \mathcal{N}_b \mathrm{d}\vec{S} = \sum_{c=1}^{N_c} \frac{1}{2} \frac{1-\beta_{\mathrm{al}}}{1+\beta_{\mathrm{al}}} \int_{\Gamma_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}\vec{S}, \qquad g = 1, 2, \tag{3.22}$$

where N_c is the number of faces belonging to the reactor boundary.

If zero-current boundary conditions are considered, the surface integral terms are equal to zero and the finite element formulation takes into account these conditions without restrictions in the nodes as

$$\int_{\Gamma_c} \mathcal{N}_a \vec{\nabla} \mathcal{N}_b \mathrm{d}\vec{S} = 0. \tag{3.23}$$

Zero-current boundary conditions also keeps symmetry in the neutron diffusion equation.

Zero-flux boundary conditions are assumed if the nodal values on the boundary are exactly fixed to zero. Thus, their associated shape functions do not appear in problem (3.12).

3.2.2 Normalization

The fluxes (eigenvectors) are not well determined by solving the eigenvalue problem and it is necessary to establish a normalization criterion. For that, we define the neutron power.

The thermal power is proportional to the neutron power generated by the reactor if it is assumed that every fission generates a constant average amount of energy, κ. The neutron power is defined as a weighted sum of neutron fluxes

$$P^{\delta} = \sum_{g=1}^{G} \kappa \Sigma_{fg} |\phi_g^{\delta}| \,. \tag{3.24}$$

The absolute value is introduced in this definition to extend the neutron power to subcritical modes where the fluxes have positive and negative values.

One of the most used criteria is to force that the integral over the reactor domain of the neutron power must be equal to the total volume of the reactor, V_t. That can be written as

$$\frac{1}{V_t} \int_V P^{\delta} \mathrm{d}V = \frac{1}{V_t} \sum_{g=1}^{G} \int_V \Sigma_{fg} \left| \phi_g^{\delta} \right| \mathrm{d}V = 1 \,. \tag{3.25}$$

In the normalization process, several considerations are taken into account. If values for Σ_{fg} are not available it is common to use $\nu\Sigma_{fg}$ in equation (3.25) instead to recover the energy in terms on the flux. This approximation considers that ν is constant for all the materials in the core and all energy groups. The energy per fission κ is not considered in the normalization because it is considered a global constant.

3.3 Spatial discretization for the SP_N equations

The SP_N approximation is nothing more than a set of diffusion-like equations whose solutions are unknown flux moments (Equations (2.90)). Because of that, a similar process as the one used in the finite element discretization for the spatial modes problems of the neutron diffusion equation (Section 3.2), can be developed without major changes. Likewise, an algebraic generalized eigenvalue problem is obtained from applying a continuous Galerkin finite element method to equation (2.92).

To simplify the notation, only one group of energy is considered in the following formulas. The discretized SP_N equations can be written as a generalized algebraic eigenvalue problem of the form

$$\mathbf{S}\tilde{U} = \lambda\mathbf{T}\tilde{U}. \tag{3.26}$$

For instance, the matrices obtained for the discretized SP_3 equations are

$$\mathbf{S} = \begin{pmatrix} \mathbf{S_{00}} & \mathbf{S_{01}} \\ \mathbf{S_{10}} & \mathbf{S_{11}} \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} \mathbf{T_{00}} & \mathbf{T_{01}} \\ \mathbf{T_{10}} & \mathbf{T_{11}} \end{pmatrix}, \quad \tilde{U} = \begin{pmatrix} \tilde{u}^1 \\ \tilde{u}^2 \end{pmatrix}, \tag{3.27}$$

For the SP_5 equations, we have

$$\mathbf{S} = \begin{pmatrix} \mathbf{S_{00}} & \mathbf{S_{01}} & \mathbf{S_{02}} \\ \mathbf{S_{10}} & \mathbf{S_{11}} & \mathbf{S_{12}} \\ \mathbf{S_{20}} & \mathbf{S_{21}} & \mathbf{S_{22}} \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} \mathbf{T_{00}} & \mathbf{T_{01}} & \mathbf{T_{02}} \\ \mathbf{T_{10}} & \mathbf{T_{11}} & \mathbf{T_{12}} \\ \mathbf{T_{20}} & \mathbf{T_{21}} & \mathbf{T_{22}} \end{pmatrix}, \quad \tilde{U} = \begin{pmatrix} \tilde{u}^1 \\ \tilde{u}^2 \\ \tilde{u}^3 \end{pmatrix}. \tag{3.28}$$

In the previous expressions, $\tilde{u}^1$, $\tilde{u}^2$ and $\tilde{u}^3$ are the algebraic vectors representing u^1, u^2 and u^3 and the matrix blocks elements are defined as

$$(\mathbf{T_{nn}})_{ab} = \sum_{c=1}^{N_c} \mathbb{D}_{nn} \int_{V_c} \vec{\nabla}\mathcal{N}_a \vec{\nabla}\mathcal{N}_b \, \mathrm{d}V - \mathbb{D}_{nn} \int_{\Gamma_c} \vec{\nabla}\mathcal{N}_a \mathcal{N}_b \mathrm{d}V + \mathbb{A}_{nn} \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V,$$

$$(\mathbf{T_{nn'}})_{ab} = \sum_{c=1}^{N_c} \mathbb{A}_{nn'} \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V, \qquad \text{for} \quad n' \neq n,$$

$$(\mathbf{S_{nn'}})_{ab} = \sum_{c=1}^{N_c} \mathbb{F}_{nn'} \int_{V_c} \mathcal{N}_a \mathcal{N}_b \mathrm{d}V,$$

where the matrices $\mathbb{D}_{nn}$, $\mathbb{A}_{nn'}$ and $\mathbb{F}_{nn'}$ were previously defined in Chapter 2.2 (Section 2.6). The Lagrange polynomial $\mathcal{N}_a$ is the shape function associated with the a-th degree of freedom defined in Section 3.1.1.

In the $\mathrm{SP_N}$ case, the solution fluxes are normalized by forcing

$$1 = \frac{1}{V_t} \sum_{g=1}^{G} \int_V \Sigma_{fg} |\phi_0^g| \, \mathrm{d}V, \tag{3.29}$$

where ϕ_0^g is the scalar flux.

In the following, as for the spatial modes fluxes, we denote the algebraic fluxes $\tilde{\phi}$, $\tilde{u}$ with the same expression as the continuous fluxes ψ, u, by removing the tilde.

3.4 Numerical results

This section is divided in two parts. First, the performance of the finite element method described above is studied to determine the λ, the γ and the α-modes associated with the neutron diffusion equation. Moreover, the differences between the different spatial modes problems are analyzed. In this part three different tridimensional benchmark problems have been considered for different purposes: an homogeneous reactor, the Langenbuch reactor, and the NEACRP reactor. In the second part, the finite element method is studied for the $\mathrm{SP_N}$ equations for the C5G7 benchmark. This part includes the analysis of different types of meshes, refinement sizes and degree of polynomials in the finite element method.

The solution of the eigenvalue problems are computed with the solvers that will be presented in Chapter 4.

The code to compute the different spatial modes has been implemented in `C++` language. The numerical results are executed in a computer with an Intel®

Core™ i7-4790 @3.60GHz×8 processor with 32Gb of RAM running Ubuntu GNU/Linux 16.04.

3.4.1 Homogeneous reactor

A 3D prismatic reactor with homogeneous material is considered because it can be solved analytically for every type of eigenvalue problem. The analytical solution for this benchmark is defined in Appendix B.1. The mesh considered for the discretization of the reactor is composed of 36 cells of size 50×50 cm^2 per 6 planes of height 75 cm, having a total of 216 cells. The material cross sections for the prismatic reactor are displayed in Table B.1.

First, we validate the results obtained with the code against the analytical solution. For that, we need to choose several type of errors. The *Eigenvalue error* in pcm is defined as

$$\varepsilon_{\delta} = 10^5 \times \frac{|\delta_m - \delta_m^*|}{|\delta_m^*|}, \tag{3.30}$$

where δ_m is the m-th computed eigenvalue ($\delta_m = \lambda_m, \gamma_m, \alpha_m$) and δ_m^* is the analytical eigenvalue.

Table 3.1 shows the first two λ, γ and α analytical and numerical modes computed with different finite element degrees. This Table shows the convergence of the finite element method and that good approximations are obtained with polynomial degree in the finite element method, p, equal to 2 or larger. Furthermore, it is observed that for the same degree of polynomial, the error in eigenvalues is lower for λ and γ-modes than for α-modes. Several relative errors related to the eigenfunctions are analyzed but the values obtained are negligible.

3.4.2 Langenbuch reactor

The Langenbuch benchmark (Langenbuch et al., 1977) in steady state is chosen to compare the different modes in a benchmark with several materials. It has 1170 different assemblies including 545 cells modelling the reflector. The geometry and the definition of the 5 different materials and their cross sections at $t = 0.0$ s are defined in Appendix B.3.

The 5 first modes are displayed in Table 3.2. This Table shows that the reactor is critical, since we have forced to start in critical state. Note that, the first α-mode is not equal to zero, since for the calculation of these modes we have imposed that the reactor was not exactly critical. If not, computational errors are obtained. Moreover, the second and the third eigenvalues are degenerated due to the spatial symmetry of the reactor (Tommasi et al., 2016). If one compares the

Table 3.1: Modes and errors for the homogeneous reactor.

	First eigenvalue		Second eigenvalue	
p	δ_1	$\varepsilon_\delta(pcm)$	δ_2	$\varepsilon_\delta(pcm)$
λ-modes				
1	1.003649	37.27	0.993831	140.29
2	1.004022	0.12	0.995208	1.98
3	1.004023	0.02	0.995227	0.01
Anal. solut.:	1.004024		0.995227	
γ-modes				
1	1.003619	134.92	0.993831	348.44
2	1.002266	0.07	0.997295	1.10
3	1.002267	0.02	0.997306	0.01
Anal. solut.:	1.002266		0.997306	
α-modes				
1	160.6970	9313	-271.5494	29248
2	177.1258	41	-210.9691	410
3	177.1995	0.06	-210.1068	3.1
Anal. solut.:	177.1995		-210.1002	

spectrum of the different kind of modes, this is more clustered in the γ-modes computation.

Table 3.2: Eigenvalues in Langenbuch reactor at critical state.

Modes	**1st**	**2nd**	**3rd**	**4th**	**5th**
λ-modes	1.000000	0.968020	0.968020	0.951963	0.937756
γ-modes	1.000000	0.981841	0.981841	0.972668	0.964403
α-modes	-0.00026	-867.189	-867.189	-1277.52	-1710.17

Figure 3.5 represents the thermal flux in the middle plane to compare the spatial distribution of the modes. The distributions corresponding to the first eigenvalues, that represent the flux of the reactor in steady-state are identical for the three kind of modes. The second and third modes (degenerated between them) present large differences in the spatial shapes in comparison with the other kind of modes. The next modes represented are not significantly different.

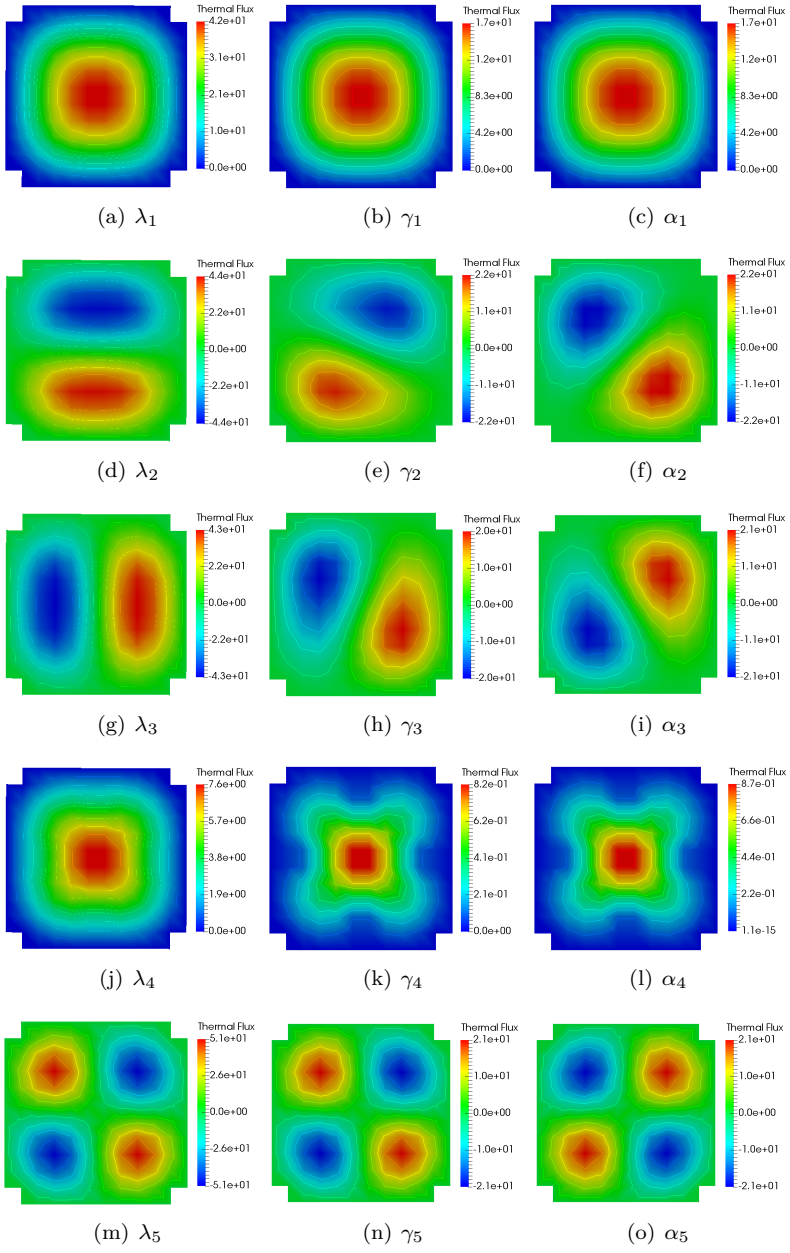


Figure 3.5: Radial distribution of thermal flux for the Langenbuch reactor.

3.4.3 NEACRP reactor

A critical configuration of the NEACRP (case A1) benchmark (Finnemann and Galati, 1991) is chosen to compare the different modes in a more realistic case. It is composed of 3978 different assemblies. The definition of the benchmark can be found in Appendix B.4. Some subcritical configurations are also defined by dividing the fission cross sections of the different materials by 1.1 (Perturbation I) and by 1.2 (Perturbation II).

Table 3.3 displays the results for the first four eigenvalues for the three configurations of reactor considered. It is observed that the reactor without perturbations is quasi-critical since the dominant λ and γ are near 1, and α is near to 0. In the Perturbation I and II, the reactor is subcritical with $k_{\text{eff}} = 0.90$ and $k_{\text{eff}} = 0.83$, respectively.

Table 3.3: Eigenvalues at initial state of NEACRP reactor.

	Eigenvalues		
T. modes	1st	2nd	3th=4th
Critical State			
λ-modes	1.0002	0.9886	0.9854
γ-modes	1.0001	0.9937	0.9919
α-modes	7.6341	-442.32	-573.62
Perturbation I			
λ-modes	0.9093	0.8907	0.8958
γ-modes	0.9490	0.9429	0.9412
α-modes	-3462.14	-3861.18	-4006.92
Perturbation II			
λ-modes	0.8335	0.8238	0.8212
γ-modes	0.9048	0.8990	0.8974
α-modes	-6252.36	-6610.89	-6754.83

Figure 3.6 shows the radial and axial profiles for the fast flux associated with the first three modes in the for critical configuration of the reactor. The radial profiles of fast flux functions are approximately equal for the first and second eigenvalues, observing small differences for the third one. In the last row of the Figure, only the axial profiles associated with the first three λ-modes are shown because the axial profile obtained for the γ and α-modes are very close to these ones. Furthermore, it is deduced that the fast flux for the first modes

is positive and has radial and axial symmetry, whereas the second modes are antisymmetric in axial profiles and symmetric in the radial ones. The third modes are antisymmetric in the radial and the axial profiles.

The power average profiles (axial and radial) are used to compare the modes between them in the three configurations of the reactor. The axial profiles are computed as

$$P_{\mathrm{ax}}(z) = \frac{\frac{1}{L_x}\frac{1}{L_y}\int_{L_x}\int_{L_y}\Sigma_{f1}|\xi_1(\vec{r})| + \Sigma_{f2}|\xi_2(\vec{r})|\ \mathrm{d}y\ \ \mathrm{d}x}{\frac{1}{V_t}\int_V(\Sigma_{f1}|\xi_1| + \Sigma_{f2}|\xi_2|)\ \mathrm{d}V},$$

where V_t, L_x and L_y are the total volume, the width and the depth of the reactor core, respectively. The functions ξ_i, $i = 1, 2$ are the corresponding fast and thermal fluxes for the different kind of modes. The radial profiles are computed in a similar way. These profiles are shown in Figure 3.7. For critical configuration, there are no differences between the profiles of the modes. In the same way, there are no differences between the profiles of λ and γ-modes in Perturbation I and Perturbation II. However, when the reactor becomes more subcritical the α-modes power shape changes with respect to its shape in the critical configuration and the other modes shapes in the radial profile.

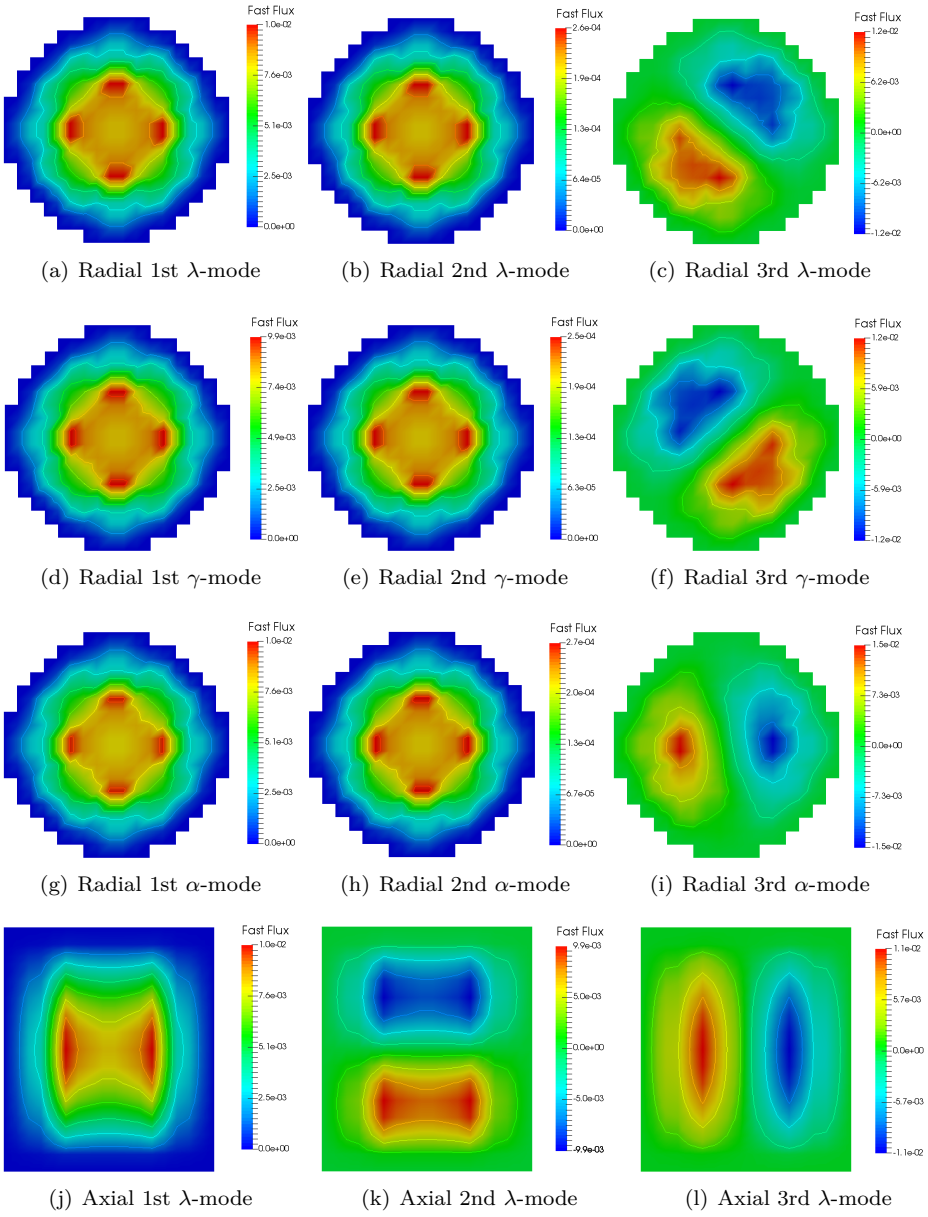
Spectral index

To demonstrate the variation in the neutron energy spectrum, the spectral index I is studied. This index is defined as the ratio of the fluxes integrated in the core volume (Ronen et al., 1976),

$$I = \frac{\int_\Omega \xi_1\ \mathrm{d}V}{\int_\Omega \xi_2\ \mathrm{d}V}, \tag{3.31}$$

where ξ_1, ξ_2, are the fast and thermal fluxes of λ, α and γ-modes.

16 configurations of NEACRP benchmark have been considered to study the spectral index behavior of the λ, γ and α-modes. These configurations are obtained modifying the position of central control rod, since in the case A1 of NEACRP benchmark, the central control rod is moved. In Figure 3.8, the differences between the spectral indexes (I) and the spectral index when the reactor is in critical configuration (I_c) are represented in each configuration as a function of the λ-eigenvalue. Figure 3.8(a), also shows that near criticality these differences in each mode are nearly equal and increasing. However, from $\lambda = 1.001$ these functions are separated as λ increases and when $\lambda = 1.003$ they become decreasing functions. Moreover, in Figure 3.8(b), we observe that



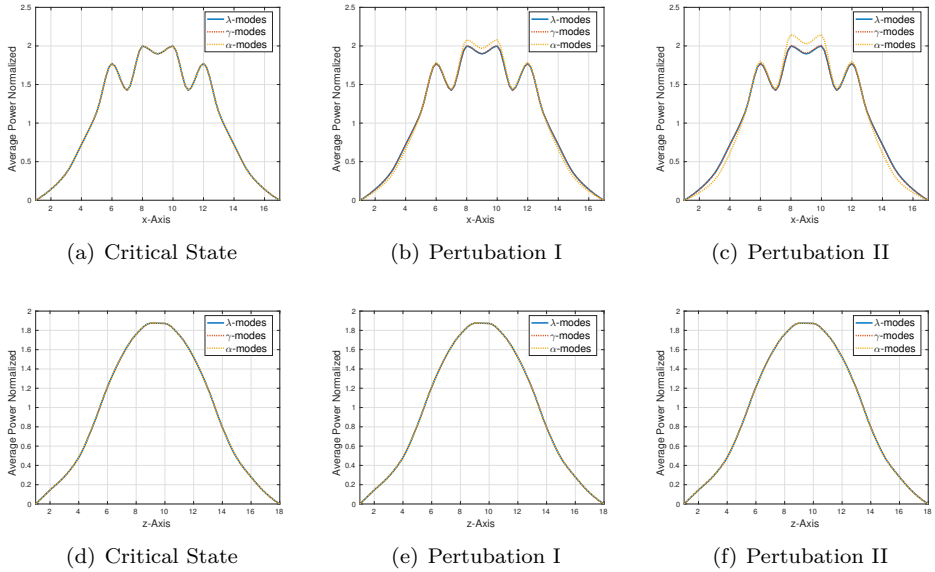
(a) Radial 1st λ-mode (b) Radial 2nd λ-mode (c) Radial 3rd λ-mode

(d) Radial 1st γ-mode (e) Radial 2nd γ-mode (f) Radial 3rd γ-mode

(g) Radial 1st α-mode (h) Radial 2nd α-mode (i) Radial 3rd α-mode

(j) Axial 1st λ-mode (k) Axial 2nd λ-mode (l) Axial 3rd λ-mode

Figure 3.6: Radial and axial fast flux profiles for NEACRP in critical configuration.

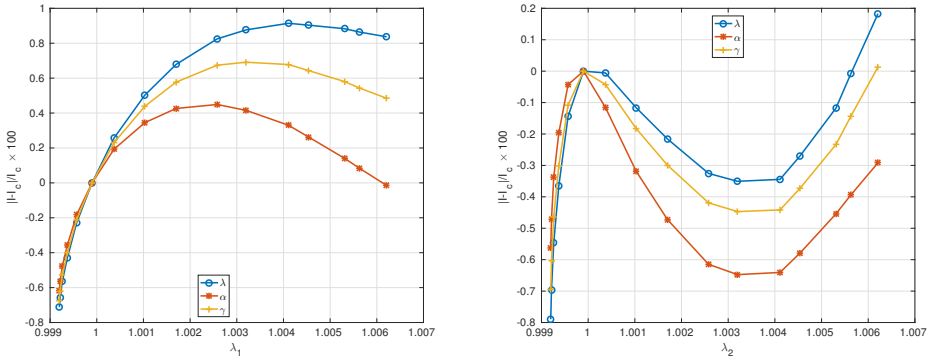


(a) Critical State (b) Pertubation I (c) Pertubation II

(d) Critical State (e) Pertubation I (f) Pertubation II

Figure 3.7: Average power profiles for the two configurations of NEACRP reactor.

functions are similar when the first λ-eigenvalue is close to 1. For the second mode, a relative maximum is observed for $\lambda_2 = 1$. So, the spectral indexes for λ, γ and α-modes have the same behavior.



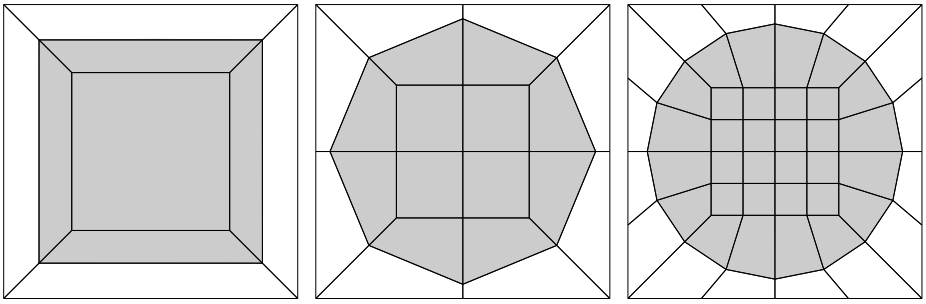
(a) First mode (b) Second mode

Figure 3.8: Spectral indexes in NEACRP reactor.

3.4.4 2D C5G7 benchmark

First, the performance of the FEM for the SP_N equations is studied by the two dimensional version of the C5G7 fuel assembly benchmark introduced by the Nuclear Energy Agency (NEA) in (Lewis et al., 2001). The characteristics of this benchmark are that it uses 7 energy groups and the cross-sections are given at homogenized pin level. The benchmark has been analyzed with several deterministic codes and a very precise solution was obtained as reference using the Monte Carlo method. The description of this benchmark can be found in Appendix B.5.

Figure 3.9 shows the proposed meshes used to discretize the pin cell by depending on the radial refinement parameter r_r. All meshes maintain the area of the fuel region (in grey color) to provide a more accurate model of the problem.



(a) $r_r = 0$ (b) $r_r = 1$ (c) $r_r = 2$

Figure 3.9: Meshes considered for the pin structure.

Tables 3.4 and 3.5 show the results for the fundamental eigenvalue and its corresponding eigenvector associated with diffusion theory (SP_1) and the SP_3 equations, respectively. These results are computed for different meshes by depending on the finite element polynomial degree p and the mesh refinement parameter, r_r. Errors in the eigenvalue are given by Equation (3.30) where $\lambda_{\text{ref}} = 1.186550$ is the reference eigenvalue given by the benchmark authors. To assess the eigenvector, the following collective per cent error measures were selected: average pin power per cent error (AVG) and mean relative pin power per cent error (MRE) of the pin power per cent error distribution (Smith et al.,

2003),

$$\mathrm{AVG} = \frac{1}{N_c} \sum_{i=1}^{N_c} |e_i|, \tag{3.32}$$

$$\mathrm{MRE} = \frac{\sum_{i=1}^{N_c} |e_i| \, P_i}{\sum_{i=1}^{N_c} P_i}, \tag{3.33}$$

where N_c is the number of fuel pin cells and e_i is the calculated per cent error for the i-th pin neutron power, P_i. It can be seen that the results are spatially converged for $r_r = 1$ and $p = 2$. Furthermore, it is observed that the SP_3 equations improve the accuracy of the results with respect to the SP_1 mainly for the eigenvector.

Table 3.4: Accuracy results for SP_1.

r	p	Number of Cells	Number of DoFs	Eigenvalue λ_1	ε_{λ_1}	AVG (%)	MRE (%)
0	1	11 849	83 664	1.185 11	144	2.26	1.88
0	2	11 849	333 207	1.185 12	143	2.65	2.02
0	3	11 849	748 636	1.185 11	144	2.26	1.88
1	1	28 900	203 735	1.183 81	274	1.52	1.25
1	2	28 900	812 063	1.183 35	320	1.43	1.21
1	3	28 900	1 824 991	1.183 30	325	1.42	1.21
2	1	78 608	553 119	1.183 73	282	1.46	1.24
2	2	78 608	2 206 743	1.183 26	329	1.42	1.21
2	3	78 608	4 960 879	1.183 25	330	1.42	1.21

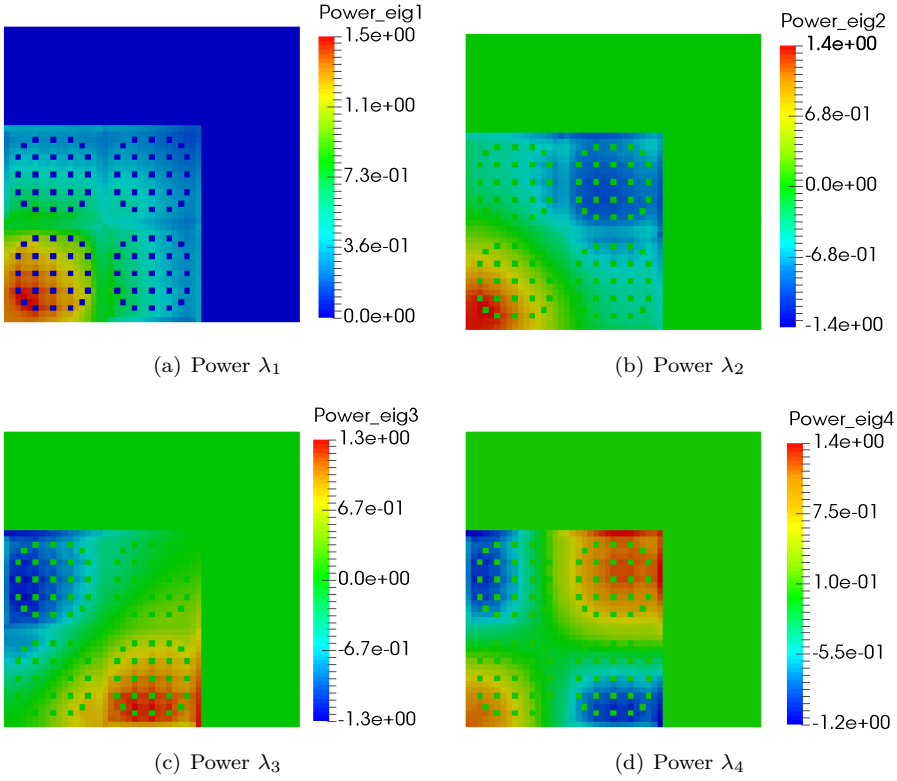
Table 3.5: Accuracy results for SP_3.

r	p	Number of Cells	Number of DoFs	Eigenvalue λ_1	$\Delta\lambda_1$	AVG (%)	MRE (%)
0	1	11 849	167 328	1.185 40	115	1.69	1.46
0	2	11 849	666 414	1.183 75	280	0.85	0.74
1	1	28 900	407 470	1.183 57	298	0.86	0.72
1	2	28 900	1 624 126	1.182 61	394	0.72	0.65
2	1	78 608	1 106 238	1.183 47	308	0.81	0.72

Figure 3.10 shows the spatial distribution of four power harmonic modes, defined as

$$P_m = \sum_{g=1}^{G} \Sigma_{fg} \phi_{0,m}^{g} \,. \tag{3.34}$$

Note that, they are not exactly the first four modes because, we have considered a quarter of the reactor for the computation. The first distribution coincides with the spatial distribution of the reactor in steady-state.



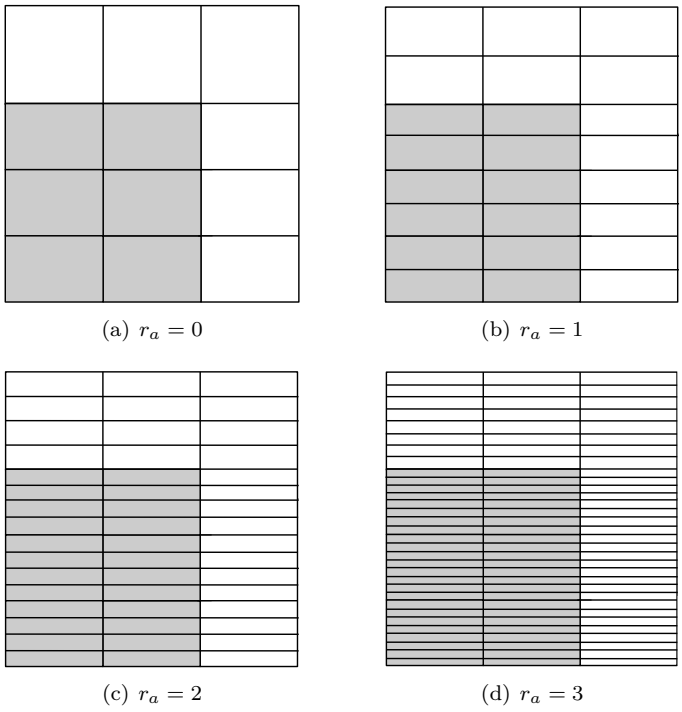
(a) Power λ_1 (b) Power λ_2

(c) Power λ_3 (d) Power λ_4

Figure 3.10: Distribution of neutron power for the C5G7 reactor.

3.4.5 3D C5G7 benchmark

The 3D-C5G7 problem (Smith et al., 2003) has been solved to test the finite element methodology in a challenging three dimensional problem. The mesh has 264 992 finite element cells and 2 343 865 degrees of freedom. This problem has the same radial configuration as the two dimensional version (Appendix B.5); then, the discretization used in this direction has been $r_r = 1$ (Figure 3.9). The axial discretization is done by extruding the two dimensional mesh by axial plane (Figure B.7). The discretizations considered in the axial direction are represented in Figure 3.11. The configuration of the finite element method to compute the solution for this case has been $p = 2$. In this case, the number of eigenvalues requested is 1.



(a) $r_a = 0$ (b) $r_a = 1$

(c) $r_a = 2$ (d) $r_a = 3$

Figure 3.11: Discretization considered axially.

Table 3.6 displays the accuracy results for the first eigenvalue and its corresponding eigenvector for SP_1 and the SP_3 equations for different axial refinement parameters, r_a. One can see that it is necessary $r_a = 2$ to get spatially convergent

results. Thus, the pin power error is less than 1% and about 400 pcm in the k_{eff} for the SP_3 equations.

Finally, one can compare these problem sizes with the bidimensional version of this reactor (with $r_r = 1$ and $p = 2$) by means of the number of degree of freedoms (Dofs) for the SP_3 approximation (Table 3.5). It is noted that the size of the problem is multiplied from more than 10 times, for $r_a = 0$, to more than 75 times, for $r_a = 3$. This implies for the 3D benchmark large expenses of computational resources in terms of computational time and memory.

Table 3.6: Accuracy results for 3D-C5G7 benchmark.

Eq.	$\mathbf{r_a}$	$\mathbf{r_r}$	$\boldsymbol{p}$	**Number of Cells**	**Number of DoFs**	**Eigenvalue** λ_1	$\Delta\lambda_1$	**AVG** (%)	**MRE** (%)
SP_1	0	1	2	132 496	8 964 375	1.140 10	298	4.26	3.35
	1	1	2	264 992	16 407 055	1.138 76	432	1.98	1.60
	2	1	2	529 984	31 292 415	1.138 19	489	1.42	1.19
	3	1	2	1 059 968	61 063 135	1.138 13	495	1.41	1.17
SP_3	0	1	2	132 496	17 928 750	1.140 67	241	4.40	3.51
	1	1	2	264 992	32 814 110	1.139 36	372	1.93	1.54
	2	1	2	529 984	62 584 830	1.138 80	428	0.98	0.84
	3	1	2	1 059 968	122 126 270	1.138 70	438	0.87	0.77

CHAPTER 4

EIGENVALUE COMPUTATION METHODS

The spatial discretization of the different steady-state problems associated with the approximations of the neutron transport equation gives a generalized eigenvalue problem. The largest eigenvalue shows the criticality of the system and its associated eigenfunction the neutron distribution in steady-state. The computation of the next eigenvalues has interest, for example, to develop modal methods (Chapter 5) or to study BWR instabilities (Miró et al., 2000).

Calculation of the dominant mode has traditionally used the classical power iteration method which, although robust, converges slowly for dominance ratios near one, as occurs in some practical nuclear reactor problems. Henceforth, acceleration techniques have been needed to improve the convergence of the power iteration method. Some simple improvements in diffusion theory are, for instance, Chebyshev iteration (Hageman and Young, 2012) and Wielandt shift (Sutton, 1988).

Other alternative methods have been also studied to solve this kind of reactor problems in an attempt to improve upon the performance of accelerated power iteration methods. For instance, the inverse power method for the computation of one eigenvalue (Allen and Berry, 2002). When a set of dominant modes have to be computed, other methods have been used, such as the subspace iteration method (Verdú et al., 1994; Vidal et al., 1998; Warsa et al., 2004), the classical Arnoldi method, the Implicit Restarted Arnoldi method (IRAM) (Warsa et al., 2004; Verdú et al., 1999) or, more recently, the Krylov-Schur method (Vidal-

Ferrandiz et al., 2014; Bernal et al., 2017). The application of these methods requires to transform the generalized problem

$$Ax = \delta Bx,$$

into an ordinary eigenvalue problem or to apply a shift and invert technique. In both cases, it is necessary to solve numerous linear systems associated with large matrices and, consequently, the time of convergence of the methods can be slow. One solution of this problem is using the Jacobi-Davidson method, (Verdú et al., 2005), that also makes use of a shift and invert strategy, but it does not need to solve as many linear systems as the previous ones.

Other methods to solve eigenvalue problems associated with nonsymmetric matrices are the gradient type methods, that do not require solving linear systems involving the full operator. However, if there are clustered or degenerate eigenvalues, these methods may have problems to find all the eigenvalues. In practical situations of reactor analysis, the dominance ratio corresponding to the dominant eigenvalues is often near unity, resulting in a slow convergence. In such cases, block methods with several initial approximated eigenvalues and eigenvectors are an alternative since their convergence behaviour depends only on the separation of the group of target eigenvalues from the rest of the spectrum. One of these methods is the Generalized Davidson that has been successfully used for the computation of the modes in other approximations of the neutron transport equation such as the multigroup SP_N equations (Hamilton and Evans, 2015). Other method, is the inverse-free Krylov subspace method introduced by Golub for symmetric and definite matrices in (Golub and Ye, 2002) and with a block implementation in (Quillen and Ye, 2010). This method improves the traditional steepest descent method by expanding the search direction to a Krylov subspace with the advantage of better approximation properties offered by Krylov subspaces. In this thesis, the performance of the block inverse-free Krylov subspace method is analyzed for the steady-state problems associated with the neutron transport approximations that are not always symmetric.

On the other hand, Newton's methods have been shown very efficient in the computation of eigenvalues in neutron diffusion theory. For instance, the modified block Newton method (Lösche et al., 1998) has been considered to solve an ordinary λ-modes problem associated with the original generalized problem, (González-Pintor et al., 2011). In this thesis, several generalizations of this method are proposed and analyzed. All of these methods are very sensitive to the initial guess and good approximations are needed to initialize them. In this context, hybrid methods have been developed using slow convergence methods to initiate Newton's methods. As an example, the use of Jacobian-Free Newton-

Krylov methods has been studied using traditional methods as preconditioners (Gill and Azmy, 2009; Knoll et al., 2011; Gill and Azmy, 2011) such as the IRAM (Gill and Azmy, 2009; Mahadevan and Ragusa, 2008). In this thesis, a hybrid method has been proposed that combines a block inverse-free preconditioned Arnoldi method (BIFPAM) with a generalization of the modified block Newton method.

This Chapter exposes different strategies to solve the algebraic eigenvalue problems obtained in the Chapter 3. Section 4.1 collects and describes a selection of the most commonly used eigensolvers for neutron computations. Moreover, this Section presents the block inverse-free preconditioned Arnoldi method (BIFPAM), two generalizations of the modified Newton method and a hybrid eigenvalue solver which is based on the two previous methods. Then, several methodologies to improve the implementation of the code are studied. Iterative methods improve their convergence if a suitable initial guess is given. For this purpose, Section 4.2 studies several initialization techniques. Another point that accompanies the solver is the preconditioner used. Thus, several preconditioners are exposed in Section 4.3. Section 4.4 briefly outlines the matrix-free strategy used as an optimization of the code because this avoids the assembly of the matrices involved in the problem. To finish this Chapter, some numerical results are presented to test and to compare all strategies for the SP_N equations in the C5G7 benchmark reactor. This chapter rewrites and synthesizes the methods and results presented in (Carreño et al., 2017a; Carreño et al., 2017b; Carreño et al., 2018a; Carreño et al., 2018b; Carreño et al., 2019c; Vidal-Ferràndiz et al., 2019; Carreño et al., 2019a; Carreño et al., 2019b).

4.1 Eigenvalue problem solvers

In this section, several well-known eigenvalue solvers are described. A scheme of them can be found in the Figure 4.1. Even thought, this list is not intended to be exhaustive, as several other eigenvalue solvers appear in the neutron transport computations or in general, in mathematics literature. Furthermore, although in this thesis only the solution for the diffusion equations and SP_N equations are computed, the block structure of the problem is the same for other angular approximations of the neutron transport equation, such as the S_N equations and P_N equations (even if the linear operators themselves are different). Therefore, the following methodology is equally applicable to other approximations.

Thus, it is supposed that we have a generalized partial eigenvalue problem of the form

$$AX = BX\Lambda\,, \tag{4.1}$$

where the matrices $A \in \mathbb{R}^{\mathrm{N_{dofs}} \times \mathrm{N_{dofs}}}$ and $B \in \mathbb{R}^{\mathrm{N_{dofs}} \times \mathrm{N_{dofs}}}$ have a block structure with blocks symmetric and positive definite and they come from the spatial discretization using a FEM. The matrix $\Lambda \in \mathbb{R}^{q \times q}$ is a diagonal matrix whose elements are the first q dominant eigenvalues (largest in magnitude) ($q \lll \mathrm{N_{dofs}}$) and $X \in \mathbb{R}^{\mathrm{N_{dofs}} \times q}$ has the corresponding q eigenvectors in its columns.

We assume that the eigenvalues an their corresponding eigenvectors are real, even though this is proved only under restrictive conditions (e.g. monoenergetic transport). This theoretical assumption is nevertheless supported by numerical evidence on benchmark problems (Carney et al., 2014).

In some cases, we refer to the ordinary (or standard) eigenvalue problem associated with the generalized problem to

$$CX = X\Lambda\,, \text{ where } C \equiv B^{-1}A\,. \tag{4.2}$$

To use the ordinary eigenvalue problem, the matrix B^{-1} is not constructed explicitly, but it is multiplied by vectors solving linear systems.

One special case is the λ-modes problem associated with the neutron diffusion equation in the two energy groups approximation, without considering up-scattering and one eigenvalue

$$\begin{pmatrix} L_{11} & 0 \\ S_{21} & L_{22} \end{pmatrix} \begin{pmatrix} \phi_1^\lambda \\ \phi_2^\lambda \end{pmatrix} = \frac{1}{\lambda} \begin{pmatrix} F_{11} & F_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \phi_1^\lambda \\ \phi_2^\lambda \end{pmatrix}. \tag{4.3}$$

For that case, it is very common to reduce the generalized problem to the ordinary eigenvalue problem

$$L_{11}^{-1} \left(F_{11} - F_{12}\, L_{22}^{-1}\, S_{21}\right) \phi_1^\lambda = \lambda\, \phi_1^\lambda\,, \tag{4.4}$$

and then the thermal group is computed as

$$\phi_2^\lambda = -L_{22}^{-1} S_{21} \phi_1^\lambda\,. \tag{4.5}$$

In this thesis, we only use the previous consideration when the Krylov-Schur method is applied, because this method is only available in the library SLEPc (Hernandez et al., 2005) for ordinary eigenvalue problems.

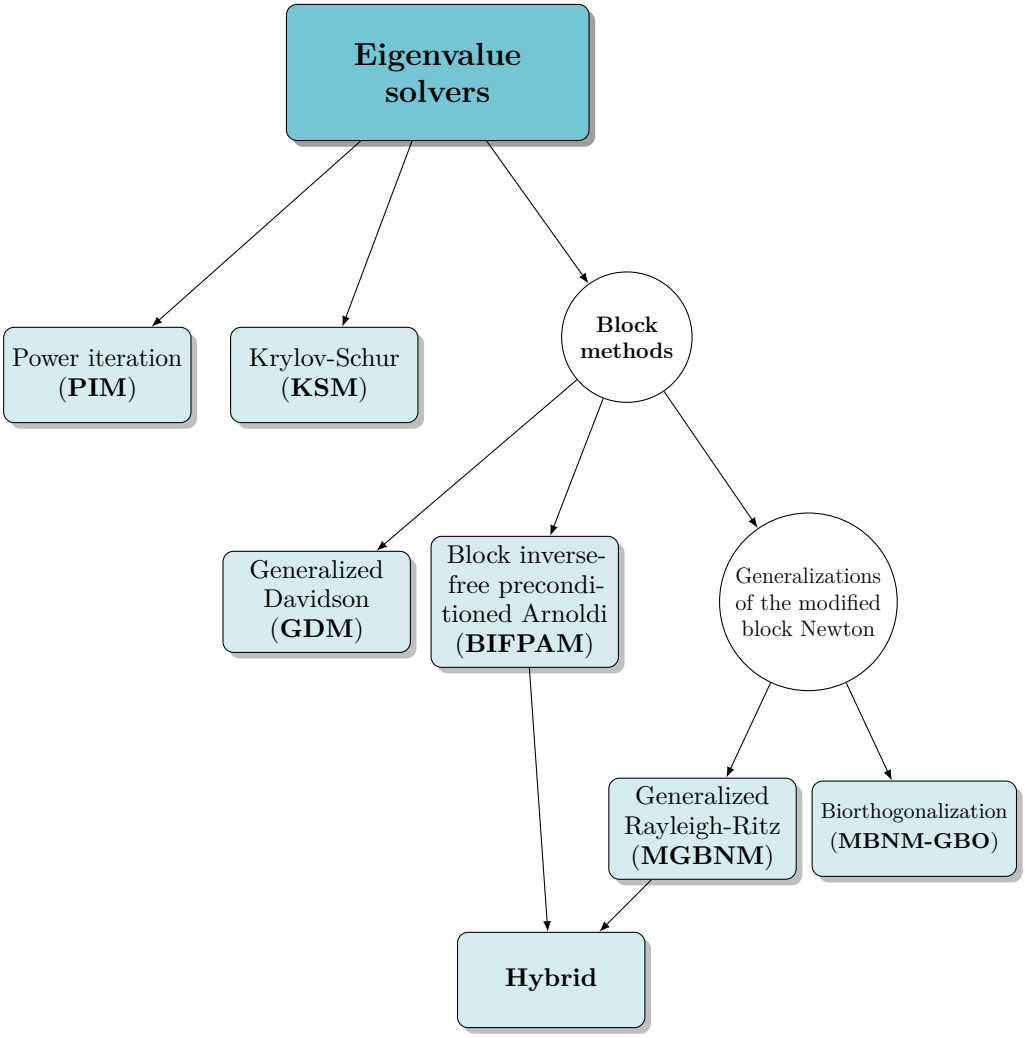


Figure 4.1: Scheme for the eigenvalue solvers.

4.1.1 The power iteration method

One of the oldest techniques for solving eigenvalue problems is the so-called power iteration method (PIM). Given the generalized eigenvalue problem for one eigenvalue

$$Ax = \delta Bx,$$

the classical power iteration method needs previously to transform it into an ordinary eigenvalue problem as

$$Cx = \delta x, \text{ where } C \equiv B^{-1}A\,.$$

This method consists simply on generating the sequence of vectors for the ordinary eigenvalue problem (4.2), $C^i x^{(0)}$ where $x^{(0)}$ is some nonzero initial vector. It can be proved that this sequence of vectors, normalized appropriately and under reasonable conditions, converges to the dominant eigenvector (Saad, 2003). The normalization most commonly used is to ensure that the largest component of the current iterate is equal to one. This yields to the Algorithm 1.

Algorithm 1 Power Iteration Method (PIM)

Input: Matrices A and B, initial approximation $x^{(0)}$.
Output: Largest eigenvalue δ_1 and its corresponding eigenvector x_1.

1: Compute $\delta^{(0)}$
2: **while** res_g>tol and $i < maxits$ **do**
3: Compute $x^{(i)} = 1/\delta^{(i-1)} B^{-1} A\, x^{(i-1)}$
4: Obtain $\delta^{(i)} = \delta^{(i-1)} \|Bx^{(i)}\| / \|Bx^{(i-1)}\|$
5: **end while**

The proof of the convergence for this method shows that the convergence factor of the method is given by (Saad, 2003)

$$\rho_{\text{PIM}} = \frac{|\delta_2|}{|\delta_1|}.$$

It is a common situation in reactor problems that the dominant eigenvalues δ_2 and δ_1 are very close one from the other. As a result, convergence may be extremely slow.

The eigenvalue can be also estimated by using the Rayleigh quotient

$$\delta^{(i)} = \delta^{(i-1)} \frac{\langle Ax^{(i)}, Bx^{(i)} \rangle}{\langle Bx^{(i)}, Bx^{(i)} \rangle}\,,$$

where $\langle , \rangle$ is a discrete inner product, leading to the Rayleigh quotient method, that in some cases has a faster convergence.

The previous algorithm, for the special case of the λ-modes problem in the approximation of two energy groups (4.3), can found as a fission source iteration

scheme (Stacey, 2018). From the notation of Equation (4.3), the iteration process can be described as:

- Approximate $f_1^{(i-1)} = F_{11}\phi_1^{\lambda,(i-1)} + F_{12}\phi_2^{\lambda,(i-1)}$.
- Solve $L_{11}\phi_1^{\lambda,(i)} = 1/\lambda^{(i-1)} f_1^{(i-1)}$.
- Compute $s_2^{(i)} = -S_{21}\phi_1^{\lambda,(i)}$.
- Solve $L_{22}\phi_2^{\lambda,(i)} = s_2^{(i)}$.
- Compute $\lambda^{(i)} = F\phi^{\lambda,(i)}/(L+S)\phi^{\lambda,(i)}$.

Note that this iteration is equivalent to the Algorithm 1 (except in the estimation of the largest eigenvalue) when one block Gauss-Seidel iteration (Algorithm 8) is applied to solve the linear systems where the matrix B is

$$B = \begin{pmatrix} L_{11} & 0 \\ S_{21} & L_{22} \end{pmatrix}. \tag{4.6}$$

The power iteration algorithm is designed to obtain the largest eigenvalue. To estimate the rest of eigenvalues to apply a deflation technique is needed. This procedure consists of applying a rank one modification to the original matrix to displace the eigenvalue δ_1 of the matrix spectrum, while keeping all other eigenvalues unchanged. The rank one modification is chosen so that the eigenvalue δ_2 becomes the one with largest modulus and therefore, the power iteration can be applied to the new matrix to estimate the pair (δ_2, x_2). Traditional deflation techniques are the Wielandt deflation and the Schur Wielandt deflation (Saad, 2003). In this thesis, the power iteration method is only used to compute the dominant eigenvalue because other works related to diffusion theory have shown that it is not a very efficient method to compute more than one eigenvalue (Bernal et al., 2017)

4.1.2 Subspace expansion methods

The power iteration method or some expansion as the shifted PIM are fixed-point methods, i.e., the next estimate of the solution depends only on the estimate immediately preceding it. As alternative to fixed-point iteration methods are subspace eigenvalue solvers in which information from several vectors is used to generate the next approximate solution. The vast majority of subspace solvers are built on two basic principles: extracting an approximating of the solution from a given subspace and adding a new vector to the current subspace.

In the solution extraction phase, given the generalized partial eigenvalue problem

$$AX = BX\Lambda,$$

or the associated ordinary eigenvalue problem

$$CX = X\Lambda, \quad \text{where} \quad C \equiv B^{-1}A,$$

one can estimate the desired eigenvectors from a linear combination of the subspace basis vectors Z as $X = ZU$, where U is an orthonormal matrix. The process is almost invariably achieved through a **Rayleigh-Ritz procedure** by solving the projected eigenvalue problem

$$A\underbrace{ZU}_{X} = B\underbrace{ZU}_{X}\Lambda \iff Z^{\mathsf{T}}AZU = Z^{\mathsf{T}}BZU\Lambda$$

or for an ordinary eigenvalue problem

$$C\underbrace{ZU}_{X} = \underbrace{ZU}_{X}\Lambda \iff Z^{\mathsf{T}}CZU = U\Lambda,$$

where Z contains a set of typically orthonormal basis vectors for the current subspace. For an appropriate selection of Z, the eigenvalues of the projected problem will closely approximate the eigenvalues of the original system, and the vectors ZU will approximate the corresponding eigenvectors X. The approximate eigenvalues and eigenvectors obtained from the Rayleigh-Ritz procedure are generally referred to as Ritz values and Ritz vectors, respectively. All eigensolvers used in this thesis use the Rayleigh-Ritz procedure to extract an approximate solution from the subspace. The generalized Rayleigh-Ritz method can be summarized in Algorithm 2.

Algorithm 2 Generalized Rayleigh-Ritz

Input: Matrices A and B, initial approximation of the invariant subspace Z.
Output: Dominant eigenvalue δ_1 and its approximated eigenvector x_1.

1: Form projection $A_z = Z^{\mathsf{T}}AZ$, $B_z = Z^{\mathsf{T}}BZ$
2: Compute dominant eigenpair (δ_1, u_1) of $A_z u = \delta B_z u$
3: Compute $x_1 = Zu_1$

The method of subspace expansion is what distinguishes the majority of subspace eigensolvers.

4.1.3 The Krylov-Schur method

Power iteration method works with two vectors. This corresponds to a Krylov subspace of dimension one. If now, the power iteration is applied to q vectors simultaneously, the result would be that the q vectors would all converge to the dominant eigenvector. If the vectors were orthonormalized at every iteration they would instead converge to the q eigenvectors that correspond to the q largest eigenvalues. This approach is called **subspace iteration**. This method operates on only the most recently computed vectors ignoring the information provided from the previous iterations.

Another typical subspace expansion method is the **Arnoldi method** (Saad, 2003), for ordinary eigenvalue problems

$$Cx = \delta x\,, \text{ where} \qquad C \equiv B^{-1}A, \tag{4.7}$$

that constructs a Krylov subspace from the vector sequence generated by power iteration of matrix C satisfying a Galerkin orthogonality condition. The Galerkin condition is satisfied through the Arnoldi decomposition of order d_k

$$CZ_{d_k} = Z_{d_k}H_{d_k} + h_{k+1,k}z_{k+1}e_{d_k}^{\mathsf{T}}, \tag{4.8}$$

where z_{d_k+1} is the result of orthonormalizing Cz_{d_k} with respect to previous columns and H_{d_k} is an upper Hessenberg matrix. That leads to a relatively small projected (upper Hessenberg) matrix from which the approximate eigenvalues and eigenvectors can be calculated easily. The efficiency of this methods improves when the dimension of the Krylov subspace d_k increases as well as the computational cost. The method can incorporate a restarting technique to avoid to have a high number of vectors obtaining the implicit restarted Arnoldi method, IRAM, (more details in (Lehoucq, 2001)) .

An improvement of Arnoldi method is the Krylov-Schur method proposed by Stewart in (Stewart, 2002) for ordinary eigenvalue problems. The Krylov-Schur method is defined by generalizing the Arnoldi decomposition (4.8) to obtain a so-called Krylov decomposition of order k,

$$CZ_{d_k} = Z_{d_k}L_{d_k} + z_{d_k+1}b_{d_k+1}^{\mathsf{T}}, \tag{4.9}$$

in which matrix L_{d_k} is not restricted to be upper Hessenberg and b_{d_k+1} is an arbitrary vector.

Krylov decompositions are invariant under (orthogonal) similarity transformations, so that

$$CZ_{d_k}Q = Z_{d_k}Q(Q^TL_{d_k}Q) + z_{d_k+1}b_{d_k+1}^TQ,$$

with $Q^{\mathsf{T}}Q = I$, is also a Krylov decomposition. In particular, one can choose Q in such way that $S_{d_k} = Q^T L_{d_k} Q$ is in a (real) Schur form, that is, upper quasi-triangular with the eigenvalues in the 1×1 or 2×2 diagonal blocks. This particular class of relation, called Krylov-Schur decomposition, can be written in block form as

$$C \begin{pmatrix} \tilde{Z}_1 & \tilde{Z}_2 \end{pmatrix} = \begin{pmatrix} \tilde{Z}_1 & \tilde{Z}_2 \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} + z_{k+1} \begin{pmatrix} \tilde{b}_1^{\mathsf{T}} & \tilde{b}_2^{\mathsf{T}} \end{pmatrix},$$

and has the nice feature that it can be truncated, resulting in a smaller Krylov-Schur decomposition,

$$C\tilde{Z}_1 = \tilde{Z}_1 S_{11} + z_{d_k+1} \tilde{b}_1^T,$$

that can be extended again to order d_k.

Subspaces expansion methods based on Krylov subspaces are widely used to compute several eigenvalues of ordinary eigenvalue problems. For generalized eigenvalue problems (4.1), linear systems are required to be solved in each iteration to solve the corresponding ordinary eigenvalue problem (4.2). This increases the computational cost, but also large errors in the convergence process can be obtained if the matrix B is ill-conditioned.

To solve the ordinary eigenvalue problems with the Krylov-Schur method, the implementation provided by the library SLEPc (Hernandez et al., 2005) has been used. There are other implementations of the Krylov-Schur method that allow a treatment of the eigenvalues in block (Baker et al., 2009), but we have not studied them.

4.1.4 The generalized Davidson method

Davidson type methods may present better performance in generalized eigenproblems

$$Ax = \delta Bx.$$

These methods are expansions of the classical Davidson method (Davidson, 1975). The basic idea behind subspace expansion in the Davidson method is, given an approximate eigenvalue, $\delta^{(i)}$ in the iteration i, and the corresponding eigenvector $x^{(i)}$, one should seek a correction, $t^{(i)}$ such that the eigenvalue correction equation given by

$$A(x^{(i)} + t^{(i)}) = \delta^{(i)} B(x^{(i)} + t^{(i)}),$$

is satisfied. Reordering the terms of this equation yields to

$$(A - \delta^{(i)} B)t^{(i)} = -(A - \delta^{(i)} B)x^{(i)} = -r^{(i)},$$

where $r^{(i)}$ is the residual of the eigenvalue problem. This equation implies that a linear system with the matrix $(A - \delta^{(i)}B)$ must be solved at each iteration. A wide variety of developments in the subspace expansion have been studied. Morgan and Scott in (Morgan and Scott, 1986) introduces the first expansion: the Generalized Davidson Method (GDM) that applies a preconditioner, P, of the matrix $(A - \delta^{(i)}B)$ to the residual to improve the approximation in the direction of the desired eigenvector as

$$Pt^{(i)} = -r^{(i)} .$$

The main advantage of this method is that no linear system is needed to be solved involving the full problem operator; only the application of a preconditioner to approximate the solution of a linear system is required. However, it may occur that the iterated vector is almost collinear to the approximated eigenvector, leading to the stagnation of the method.

Other Davidson methods, although there will not be considered in this thesis, are the **Jacobi-Davidson method** (Sleijpen et al., 1996) or the **Olsen method** (Olsen et al., 1990). The most popular is the Jacobi-Davidson whose correction is given by

$$(I - x^{(i)}x^{(i)\,\mathsf{T}})(A - \delta^{(i)}B)(I - x^{(i)}x^{(i)\,\mathsf{T}})t^{(i)} = -r^{(i)} ,$$

where the projection operator $(I - x^{(i)}x^{(i)\,\mathsf{T}})$ prevents the stagnation of the method. Even if, these methods were developed for symmetric eigenvalue problem, later work extended the theory to nonsymmetric matrices.

In this thesis, the block implementation of the library SLEPc is used to compute the eigenvalues of a generalized eigenvalue problem with the generalized Davidson method.

4.1.5 The block inverse-free preconditioned Arnoldi method

Other possibility to avoid solving linear systems for generalized eigenvalue problems is the block inverse-free preconditioned Arnoldi method (BIFPAM) where the subspace expansion is obtained from a Krylov subspace corresponding to residual matrices, $A - \delta_m B$. The BIFPAM was originally presented for A and B symmetric matrices and B positive definite (see (Golub and Ye, 2002)). Nevertheless, this thesis shows that this methodology works efficiently to compute the modes associated with some reactor problems, where matrices A and B are not symmetric. The performance of this method was studied for the neutron diffusion equation in (Carreño et al., 2018a; Carreño et al., 2018b; Carreño et al., 2019a) and for $\mathrm{SP_N}$ equations in (Vidal-Ferràndiz et al., 2019).

We start with the problem for one eigenvalue

$$Ax = \delta Bx,$$

the aim is maximizing the Rayleigh quotient

$$\delta(x) := \frac{x^\mathsf{T} Ax}{x^\mathsf{T} Bx}, \tag{4.10}$$

where x is on a certain subspace and it is assumed that $x^\mathsf{T} Bx$ never vanishes.

Following the steepest descent method and starting from an initial approximation $(\delta^{(0)}, x^{(0)})$ leads to that the approximate eigenvector $x^{(i+1)}$ in the i-th iteration can be chosen from the $\text{span}\{x^{(i)}, r^{(i)}\}$, where

$$r^{(i)} = \frac{(A - \delta^{(i)} B)x^{(i)}}{x^{(i),\mathsf{T}} Bx^{(i)}},$$

is the gradient of $\delta(x)$ evaluated in $x^{(i)}$, and $\delta^{(i)}$ is the approximation of the eigenvalue in the i-th iteration.

This can also be considered as the Rayleigh-Ritz quotient method on the subspace

$$K_1(A - \delta^{(i)} B, x^{(i)}) := \text{span}\{x^{(i)}, (A - \delta^{(i)} B)x^{(i)}\}.$$

A way to extend this approach is finding a new $x^{(i+1)}$ from the d_k-order Krylov subspace

$$K_{d_k}^{(i)}(A - \delta^{(i)} B, x^{(i)}) := \text{span}\{x^{(i)}, (A - \delta^{(i)} B)x^{(i)}, \ldots, (A - \delta^{(i)} B)^{d_k} x^{(i)}\},$$

and then using the Rayleigh-Ritz projection method exposed in Algorithm 2. The matrix Z is a basis of $K_{d_k}^{(i)}(A - \delta^{(i)} B, x^{(i)})$. The dominant eigenvalue is obtained from the first Ritz value as $\delta^{(i+1)} = \delta_1$ and its eigenvector is obtained from the first Ritz vector $x^{(i+1)} = Zu_1$. Arnoldi method is used to construct the basis $K_{d_k}^{(i)}$.

This method can be dealt as an iteration with a block of vectors that allows computing several eigenvalues simultaneously (Quillen and Ye, 2010). If we are interested on computing q eigenvalues of problem (4.1),

$$AX = BX\Lambda\,, \tag{4.11}$$

we can accelerate the convergence by using the subspace $\mathcal{K}_{d_k}^{(i)}$ with

$$\mathcal{K}_{d_k}^{(i)} =: \bigcup_{m=1}^{q} K_{d_k,m}^{(i)}(A - \delta_m^{(i)} B, x_m^{(i)}),$$

where $\delta_m^{(i)}$ denotes the m-th eigenvalue computed in the i-th iteration and $x_m^{(i)}$ its associated eigenvector.

That means that, one may construct a basis for this subspace, Z, by the union of the q bases, Z_m for $1 \leq m \leq q$, of the m Krylov subspaces $K_{d_k,m}^{(i)}$, $1 \leq m \leq q$. Then, the original generalized eigenvalue problem (4.1) is projected onto the basis Z, and it is solved for the first q dominant eigenvalues to obtain the new eigenvalues and corresponding eigenvectors.

The rate of convergence of this method improves as the dimension of subspace, d_k, increases. However, the computational cost is also increased considerably. In this way, alternatively the method is accelerated with an equivalent transformation of the original problem by means of a preconditioner.

Golub in (Golub and Ye, 2002) proved that the rate of convergence of the block inverse-free Arnoldi method depends on the spectral distribution of $R = A - \delta B$, where δ is the desired eigenvalue. Thus, the idea of preconditioning is to construct an equivalent problem so that when we apply the block inverse-free Arnoldi method to the new problem, the new matrix associated to the equivalent problem, $\hat{R}$, has a better spectral distribution.

With an approximate eigenpair $(\delta_m^{(i)}, x_m^{(i)})$, one consider for some matrices $L_m^{(i)}$, $U_m^{(i)}$ the transformed eigenvalue problem

$$(L_m^{(i),-1} A U_m^{(i),-1})x = \delta (L_m^{(i),-1} B U_m^{(i),-1})x \Leftrightarrow \hat{A}_{m,i}\hat{x} = \delta \hat{B}_{m,i}\hat{x}, \tag{4.12}$$

which has the same eigenvalues as the original problem. The relation between the eigenvector of the original eigenvalue problem, $x_{m,i}$, and the corresponding approximate eigenvector for the transformed problem 4.12, $\hat{x}_{m,i}$, is $\hat{x}_m^{(i)} = U_m^{(i)} x_m^{(i)}$. This transformation is called preconditioning. The rate of convergence after applying one step of the inverse-free Arnoldi method to the problem ((4.12)) will be determined by the eigenvalues of

$$\hat{R}_m^{(i)} =: \hat{A}_m^{(i)} - \delta_m^{(i)} \hat{B}_m^{(i)} = L_m^{(i),-1}(A - \delta_m^{(i)} B) U_m^{(i),-1}. \tag{4.13}$$

Different preconditioning transformations can be constructed by using for instance, different factorizations of the matrix $A - \delta_m^{(i)} B$ in order to obtain a favorable distribution of the eigenvalues of matrix $\hat{R}_m^{(i)}$.

The preconditioned iteration of the block inverse-free Arnoldi method (BIFPAM) can be implemented implicitly, i.e., without explicitly forming the transformed problem $\hat{R}_m^{(i)}$. Due to the relation between the eigenvector of the transformed

problem and the original problem, obtaining a basis of $K^{(i)}_{d_k,m}$ requires only to construct a basis for the subspace $U^{(i),-1}_m \hat{K}^{(i)}_{d_k,m}$, where

$$\hat{K}^{(i)}_{d_k,m} =: \text{span}\{U^{(i)}_m x^{(i)}_m, \hat{R}^{(i)}_m U^{(i)}_m x^{(i)}_m, (\hat{R}^{(i)}_m)^2 U^{(i)}_m x^{(i)}_m, \ldots, (\hat{R}^{(i)}_m)^{d_k} U^{(i)}_m x^{(i)}_m\},$$

or equivalently of the subspace

$$\begin{aligned} K^{(i)}_{d_k,m} = U^{(i),-1}_m \hat{K}^{(i)}_{d_k,m} =: \text{span}\{&U^{(i),-1}_m L^{(i),-1}_m x^{(i)}_m, U^{(i),-1}_m L^{(i),-1}_m R^{(i)}_m x^{(i)}_m, \\ &U^{(i),-1}_m L^{(i),-1}_m (R^{(i)}_m)^2 x^{(i)}_m, \ldots, U^{(i),-1}_m L^{(i),-1}_m (R^{(i)}_m)^k x^{(i)}_m\}. \end{aligned}$$

It is observed that $(L^{(i)}_m U^{(i)}_m)^{-1}$ is only needed to premultiply the vectors when the subspace is built. In order words, we only need a method to multiply a preconditioner of the matrix $R^{(i)}_m$ by a vector that does not necessarily come from the factorization of a matrix. It can be an approximate solution of the linear system that involves the matrix $R^{(i)}_m$. Moreover, in practice, we use constants $L^{(i)}_m = L^{(1)}_1$ and $U^{(i)}_m = U^{(1)}_1$ obtained from a preconditioner for $A - \delta^{(1)}_1 B$, where $\delta^{(1)}_1$ is a first approximation of the first eigenvalue. The strategy to precondition the BIFPAM is using the preconditioners designed for linear systems that will be described in Section 4.3.

The block inverse-free preconditioned Arnoldi method (BIFPAM) is summarized in Algorithm 3.

4.1.6 The modified block generalized Newton method

Other methods different to the subspace type methods are Newton's methods. In particular, the modified block Newton method is studied. This type of Newton method is proposed in (Lösche et al., 1998) for ordinary eigenvalue problems. In this thesis, it will be extended for generalized eigenvalue problems in two different ways. These generalizations were published in (Carreño et al., 2017b).

From the generalized Rayleigh-Ritz (MBNM-GRR or MGBNM)

Given the generalized eigenvalue problem

$$AX = BX\Lambda\,, \tag{4.14}$$

it is assumed that the eigenvectors can be factorized as

$$X = ZS, \tag{4.15}$$

Algorithm 3 BIFPAM

Input: Matrices A and B, initial approximation $X^{(0)} = [x_1^{(0)}, \ldots, x_q^{(0)}]$.
Output: Diagonal matrix of eigenvalues Λ and matrix X with the eigenvectors as its columns.

1: Compute $\delta_m^{(0)} = (x_m^{(0),\mathsf{T}} A x_m^{(0)})/(x_m^{(0),\mathsf{T}} B x_m^{(0)})$, $1 \leq m \leq q$
2: **while** res_g>tol and $i < maxits$ **do**
3: Obtain the basis $Z_m^{(i)}$ of $K_{d_k,m}^{(i)}(A - \delta_m^{(i)} B, x_m^{(i)})$, $1 \leq m \leq q$ ▷ Arnoldi
4: Construct $Z^{(i)} := [Z_1^{(i)}, \ldots, Z_q^{(i)}]$
5: Form projection $A_{d_k} = Z^{(i),\mathsf{T}} A Z^{(i)}$, $B_{d_k} = Z^{(i),\mathsf{T}} B Z^{(i)}$
6: Compute q dominant eigenpairs $(\delta_m^{(i+1)}, u_m)$ of $A_{d_k} U = B_{d_k} U \Lambda$
7: Compute $x_m^{(i+1)} = Z u_m$, $1 \leq m \leq q$
8: **end while**

where $Z^{\mathrm{T}} Z = I_q$. In this way, the problem (4.14) can be rewritten as

$$AX = BX\Lambda \Rightarrow AZS = BZS\Lambda \Rightarrow AZ = BZS\Lambda S^{-1} \Rightarrow AZ = BZK. \quad (4.16)$$

This problem is undetermined since the eigenvectors are defined up to a constant. To determine the problem, the biorthogonality condition $W^{\mathrm{T}} Z = I_q$ is introduced, where W is a fixed matrix of rank q. Thus, Newton's method is used to solve the following problem

$$F(Z, \Lambda) := \begin{pmatrix} AZ - BZK \\ W^{\mathsf{T}} Z - I_q \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (4.17)$$

A new iterated solution arises as,

$$Z^{(i+1)} = Z^{(i)} - \Delta Z^{(i)}, \qquad K^{(i+1)} = K^{(i)} - \Delta K^{(i)}, \quad (4.18)$$

where $\Delta Z^{(i)}$ and $\Delta K^{(i)}$ are solutions of the system

$$\begin{cases} A\Delta Z^{(i)} - B\Delta Z^{(i)} K^{(i)} - BZ^{(i)} \Delta K^{(i)} = AZ^{(i)} - BZ^{(i)} K^{(i)}, \\ W^{\mathsf{T}} \Delta Z^{(i)} = W^{\mathsf{T}} Z^{(i)} - I_q, \end{cases} \quad (4.19)$$

which is obtained by substituting (4.18) into (4.17) and removing second order terms.

The system (4.19) is coupled, since the matrix K is not necessarily diagonal. To decouple the system, the Modified block-Newton method applies two previous

steps. The first step consists of an orthogonalization to the matrix $Z^{(i)}$ using the modified Gram-Schmidt Orthogonalization. Once $Z^{(i)}$ is an orthonormal matrix, i.e., ${Z^{(i)}}^{\mathrm{T}} Z^{(i)} = I_q$, as a second step, a Rayleigh-Ritz procedure for generalized eigenvalue problems is applied (Algorithm 2), which consists of obtaining the eigenvectors $S^{(i)}$ and their corresponding eigenvalues $\Lambda^{(i)}$ that satisfy

$$ {Z^{(i)}}^{\mathrm{T}} A Z^{(i)} S^{(i)} = {Z^{(i)}}^{\mathrm{T}} B Z^{(i)} S^{(i)} \Lambda^{(i)}. \tag{4.20} $$

Defining $\bar{Z}^{(i)} := Z^{(i)} S^{(i)}$, we have, from (4.20), that $\Lambda^{(i)}$ is a diagonal matrix whose elements, δ_m are the Ritz values and $\bar{Z}^{(i)}$ are the approximated Ritz eigenvectors, satisfying the equation

$$ {Z^{(i)}}^{\mathrm{T}} (A\bar{Z}^{(i)} - B\bar{Z}^{(i)}\Lambda^{(i)}) = 0. \tag{4.21} $$

At each iteration, the matrix W in Equation (4.19) is chosen as the previous approximation for the invariant subspace, that is, $W = \bar{Z}^{(i)}$. From the definition of $K^{(i)}$ on Equation (4.16), the system (4.19) is decoupled into the q linear systems

$$ \begin{pmatrix} A - B\delta_m^{(i)} & B\bar{Z}^{(i)} \\ {\bar{Z}^{(i)}}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \Delta\bar{z}_m^{(i)} \\ -\Delta\delta_m^{(i)} \end{pmatrix} = \begin{pmatrix} A\bar{z}_m^{(i)} - B\bar{z}_m^{(i)}\delta_m^{(i)} \\ 0 \end{pmatrix}, \; m = 1, \ldots, q, \tag{4.22} $$

where $\Delta\bar{z}_m^{(i)}$ is the i-th column of $\Delta\bar{Z}^{(i)}$. Vectors $Z^{(i+1)}$ are updated according to Equation (4.18) and the eigenvalues $\delta_m^{(i)}$ are obtained from the Ritz values of (4.20). To solve the previous linear systems the GMRES solver is used, preconditioned with the ILU factorization. Furthermore, a dynamic procedure is used to set the tolerance in each linear system.

The modified generalized block Newton method can be summarized in Algorithm 4.

From biorthogonalization process (MBNM-GBO)

Given the generalized eigenvalue problem (4.14), as in the previous method, the eigenvectors are expressed as

$$ X = ZS, \tag{4.23} $$

but now vectors Z are chosen to satisfy $H^{\mathrm{T}} B Z = I_q$ for some $H \in \mathbb{R}^{N_{\mathrm{dofs}} \times q}$. Now, the problem (4.14) is rewritten as

$$ AZ = BZK. \tag{4.24} $$

Algorithm 4 MGBNM

Input: Initial approximation $U = [u_1, \ldots, u_q]$ for the eigenvectors.
Output: Diagonal matrix of eigenvalues Λ and matrix X with the eigenvectors as its columns.

1: Compute the Ritz approximations Λ, X ▷ Rayleigh-Ritz gen.
2: **while** $\text{res}_g > tol$ and $i < maxits$ **do**
3: Compute $\Delta Z = [\Delta z_1, \ldots, \Delta z_q]$ ▷ Correc. Newton (Eq.(4.22))
4: $Z = X - \Delta Z$
5: Orthonormalize(Z) ▷ Modified Gram-Schmidt
6: Compute the Ritz approximations (Λ, X) ▷ Rayleigh-Ritz gen.
7: **end while**

The problem is determined, in this case, by imposing the biorthogonality condition $W^{\mathsf{T}}BZ = I_q$, where W is a fixed matrix of rank q. Thus it is defined the problem

$$F(Z, \Lambda) := \begin{pmatrix} AZ - BZK \\ W^{\mathrm{T}}BZ - I_q \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.25}$$

From Newton's method, a new iterated solution arises as

$$Z^{(i+1)} = Z^{(i)} - \Delta Z^{(i)}, \qquad K^{(i+1)} = K^{(i)} - \Delta K^{(i)}, \tag{4.26}$$

where $\Delta Z^{(i)}$ and $\Delta K^{(i)}$ are solutions of the system

$$\begin{cases} A\Delta Z^{(i)} - B\Delta Z^{(i)}K^{(i)} - BZ^{(i)}\Delta K^{(i)} = AZ^{(i)} - BZ^{(i)}K^{(i)}, \\ W^{\mathsf{T}}B\Delta Z^{(i)} = W^{\mathsf{T}}BZ^{(i)} - I_q. \end{cases} \tag{4.27}$$

To decouple system (4.27), this generalization of the modified block Newton method applies two previous steps. The first step is to apply to matrix $Z^{(i)}$ an algorithm, based on a biorthogonalization process (Adrover et al., 2005), to obtain $Z^{(i)}$ and $H^{(i)}$ such that ${H^{(i)}}^{\mathrm{T}}BZ^{(i)} = I_q$, (see Algorithm 5), where $H^{(i)}$ is initiated as $Z^{(i)}$.

Once $Z^{(i)}$ and $H^{(i)}$ have been obtained, as a second step, a Rayleigh-Ritz procedure is applied, which consists of obtaining the eigenvectors $S^{(i)}$ and their corresponding eigenvalues $\Lambda^{(i)}$ that satisfy

$$AZ^{(i)}S^{(i)} = BZ^{(i)}S^{(i)}\Lambda. \tag{4.28}$$

Algorithm 5 Biorthogonalization process

Input: $\bar{H}$, $\bar{Z}$
Output: $H = [H_1, \ldots, H_q]$, $Z = [Z_1, \ldots, Z_q]$ such that $H^{\mathrm{T}} B Z = I$

1: $H = \bar{H}$
2: $Z = \bar{Z}$
3: **for** m=1 to q **do**
4: $\quad H_m = H_m / (H_m^{\mathsf{T}} B Z_m)$
5: $\quad hb = H_m^{\mathsf{T}} B$
6: $\quad bz = B Z_m$
7: $\quad$ **for** j=m+1 to q **do**
8: $\quad\quad Z_j = Z_j - (hb Z_j) Z_m$
9: $\quad\quad H_j = H_j - (H_j^{\mathsf{T}} bz) H_m$
10: $\quad$ **end for**
11: **end for**

Making use of the relation of matrix $Z^{(i)}$ and $H^{(i)}$, the following equation is obtained

$$H^{(i)^{\mathsf{T}}} A Z^{(i)} S^{(i)} = S^{(i)} \Lambda^{(i)}, \tag{4.29}$$

which is a small generalized eigenvalue problem.

Defining $\bar{Z}^{(i)} := Z^{(i)} S^{(i)}$, it is satisfied that

$$H^{(i)^{\mathsf{T}}} (A \bar{Z}^{(i)} - B \bar{Z}^{(i)} \Lambda^{(i)}) = 0. \tag{4.30}$$

At each iteration, the matrix W is chosen as $W = H^{(i)}$, then the system (4.27) is decoupled into the q linear systems

$$\begin{pmatrix} A - B\delta_m^{(i)} & BZ^{(i)} \\ H^{(i)^{\mathsf{T}}} B & 0 \end{pmatrix} \begin{pmatrix} \Delta z_m^{(i)} \\ -\Delta \delta_m^{(i)} \end{pmatrix} = \begin{pmatrix} A z_m^{(i)} - B z_m^{(i)} \delta_m^{(i)} \\ 0 \end{pmatrix}, \; m = 1, \ldots, q. \tag{4.31}$$

As in the previous method, only the eigenvectors are updated with $\Delta z_i^{(i)}$ and the eigenvalues are obtained from the solution of the small problem (4.29). To solve the previous linear systems, also the GMRES solver is used, preconditioned with the ILU factorization; and a dynamic procedure is used to set the tolerance.

4.1.7 Block hybrid eigenvalue solver

Numerical results show that the BIFPAM with a 'good' preconditioner converges efficiently. However, the method for very accurate approximations exhibits a slower convergence history or the convergence rate decreases after a certain number of iterations. One the other hand, we know that theoretically the MGBNM converges quadratically. However, it needs a suitable initial guess, otherwise in the first iterations has a slow convergence. Thus, it is proposed to use a hybrid scheme combining both methods. This hybrid scheme is based on using the BIFPAM in the first iterations until a given tolerance of res $= 10^{-2}$ is reached, and then the MGBNM is applied. This cut off value for the tolerance is reactor dependent and it has been chosen by observing the convergence history of the methods. The implementation of this block hybrid method can be summarized in the Algorithm 6. This method for the neutron diffusion equation was presented in (Carreño et al., 2018a).

Algorithm 6 Block hybrid method (BIFPAM-MGBNM)

Input: Initial $X^{(0)}$
Output: Approximated eigenvalues Λ and eigenvectors X

1: Initialize X with $X^{(0)}$
2: **while** $res \geq 10^{-2}$ **do**
3: Solve $AX = BX\Lambda$ to obtain X^{BIFPAM} ▷ BIFPAM
4: **end while**
5: Initialize X with $X^{(0)} = X^{\text{BIFPAM}}$
6: **while** $res \geq tol$ **do**
7: Solve $AX = BX\Lambda$ to obtain X^{MGBNM} ▷ MGBNM
8: $X = X^{\text{MGBNM}}$
9: **end while**

4.1.8 Numerical results for the computation of the spatial modes

This section compares the computation of the λ-modes, γ-modes and α-modes with the Krylov-Schur method in two benchmarks: a homogeneous reactor and the NEACRP reactor. Then, a strategy to compute efficiently the γ-modes and α-modes is analyzed for the NEACRP reactor. It is composed of two steps: first an approximation of the λ-modes is obtained with the Krylov-Schur method and then, one generalization of the modified Block Newton method is applied. The

approximation of the λ-modes are taken as initial guess for the Newton's method. The two generalizations of the modified block Newton method are tested.

All iterative methods need a stopping criterion to establish when it has a sufficiently accurate approximation. For this purpose, two type of residual errors are defined.

Given the generalized eigenvalue problem

$$A^\delta \phi_m = \delta_m B^\delta \phi_m, \quad m = 1, \ldots, q,$$

where q is the total number of eigenvalues computed, A^δ and B^δ are the matrices associated with the δ-modes problem ($\delta = \lambda, \gamma, \alpha$), δ_m the m-th eigenvalue and ϕ_m its corresponding eigenvector, the residual error in the generalized eigenvalue problem is defined as

$$\text{res}_g = \max_{m=1,\ldots,q} \frac{\|B^\delta \phi_m - \delta_m A^\delta \phi_m\|_2}{\|\phi_m\|_2}. \tag{4.32}$$

On the order hand, the residual error considered for the ordinary eigenvalue problems associated is defined as

$$\text{res}_o = \max_{m=1,\ldots,q} \frac{\|(B^\delta)^{-1} A^\delta \phi_m - \delta_m \phi_m\|_2}{\|\phi_m\|_2}. \tag{4.33}$$

Homogeneous reactor

This homogeneous reactor tests the Krylov-Schur method to compute the different types of modes. It is presented in Appendix B.1 and their modes are described in Chapter 3 (Section 3.4.1). The algebraic problem for each type of mode is the one obtained from the spatial discretization using a polynomial degree equal to $p = 3$ in the finite element method. The number of eigenvalues computed has been 4. As Krylov-Schur method works with the ordinary eigenvalue problems, the stopping criterion has been $\text{res}_o < 10^{-8}$. The method to solve the linear systems has been GMRES, with ILU(0) preconditioner and a Cuthill-McKee reordering to reduce the bandwidth of the matrices. The dimension of the Krylov subspace has been set to 19.

Computational data related to Krylov-Schur method are displayed in Table 4.1. First, it is observed that more iterations are needed by the Krylov-Schur method to reach the tolerance for γ-modes than for the other modes. This is because the

spectrum of these modes is more clustered. For instance, the dominance ratio in this reactor is $\gamma_1/\gamma_2 = 1.004973$ in comparison with $\lambda_1/\lambda_2 = 1.008839$ for the λ-modes and $\alpha_1/\alpha_2 = 0.843404$ for the α-modes. The number of iterations for the Krylov-Schur method is larger as this ratio is closer to 1.

However, if the mean number of the iterations to solve the linear systems with GMRES method is compared, the lowest value is obtained for the γ-modes. This is because the matrix B^γ is symmetric (see Equation (3.16)). In this comparison, the mean number of iterations needed for the α-modes is much larger when compared with the other modes. The reason is that the matrix B^α, that comes from the discretization of $-(\mathcal{L}+\mathcal{S})+\mathcal{F}$, is ill-conditioned. In fact, the condition number of each one of B^δ, cond(B^δ), is estimated. This value for the matrix corresponding to the α-modes problem is two orders of magnitude larger than the one for matrices corresponding to the other modes. This can be explained because the desired α-modes are close to 0. Thus, it is very expensive to converge the solution of linear systems associated with B^α. Moreover, the residual error obtained for the generalized problem for the α-modes is very high ($\text{res}_g \approx 10^{-2}$ in comparison with the error in the associated ordinary eigenvalue problem, that is $\text{res}_o \approx 10^{-8}$. In this way, to compute a solution of α-modes with a residual error $\text{res}_g \approx 10^{-6}$, we will need to request approximately a tolerance in the corresponding ordinary eigenvalue problem of $\text{res}_o \approx 10^{-12}$. These characteristics of the modes imply that the λ-modes are the cheapest modes to compute.

Table 4.1: Data of eigenvalue problem for homogeneous reactor solved with the Krylov-Schur method.

Type of Modes	**Its. Krylov**	**Mean its. GMRES**	**cond(B^δ)**	**δ_1/δ_2**	**res$_g$**	**CPU Time(s)**
λ-modes	13	6.99	2.59e+02	1.009	3.75e-07	3.1
γ-modes	23	4.00	2.13e+02	1.005	4.17e-06	12.8
α-modes	3	30.48	5.71e+05	0.843	6.85e-02	15.3

NEACRP reactor

NEACRP reactor (Appendix B.4) is a more realistic tridimensional benchmark reactor to compare the computation of the different types of spatial modes. The dominant modes for this reactor are displayed and represented in Section 3.4.3.

First, the Krylov-Schur method is applied to solve the ordinary eigenvalue problems associated with the spatial modes (Equations (4.2) for the γ and α-modes and (4.4) for the λ-modes). The Krylov subspace dimension is set to 21. The stopping criterion is $\text{res}_o = 10^{-8}$ for the λ and the γ-modes problems, and $\text{res}_o = 10^{-12}$ for the α-modes. This distinction is done to obtain good approximations (with residual errors less than $\text{res}_g = 10^{-5}$) in the generalized eigenvalue problem (see more details in the previous Section 4.1.8). The number of eigenvalues computed has been 4.

Table 4.2 displays the data obtained with the Krylov-Schur method for the computation of the dominant modes in three configurations of reactor: a critical state and two subcritical states (Perturbation I and Perturbation II). The computational times show that the computation of the λ-modes is much quicker than the one for the other modes for similar residual errors. One reason is that the ordinary λ-modes problem has half-size (see Equation (4.4)). The other reasons are due to the characteristics of the rest of the spatial modes. High times for the computation of γ-modes are due to the high number of iterations needed to converge the Krylov method since the spectrum of eigenvalues is clustered, as it has been already discussed for the homogeneous reactor. In the case of α-modes, different reasons justify this behaviour depending on the configuration of the reactor. For the critical configuration, the mean number of iterations of GMRES method is very high since the matrix B^α is ill-conditioned, but the number of iterations of the Krylov method is low since the eigenvalues are relatively spaced. When the reactor is more subcritical the matrix B^α becomes better conditioned (since the eigenvalues are away from 0) and this is reflected in the number of iterations needed by the GMRES method. Nevertheless, the number of iterations of Krylov method is also increased because the eigenvalues are more clustered. Consequently, the computational times needed to compute the α-modes are reduced as the reactor becomes more subcritical, but in any case, the times remain larger than the times needed to compute the λ-modes.

Table 4.2: Eigenvalues in initial state of NEACRP reactor computed with Krylov-Schur method.

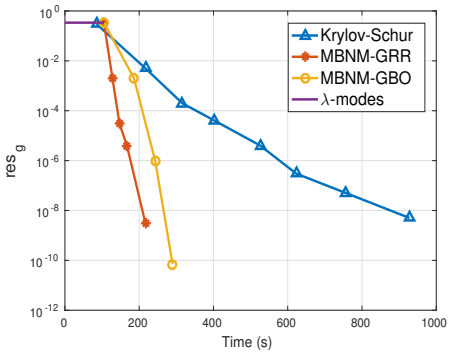
Type of modes	**its. Krylov**	**mean its. GMRES**	**res$_g$**	**CPU Time**
Critical State				
λ-modes	13	14.9	1.5e-7	105 s
γ-modes	22	10.6	6.1e-7	661 s
α-modes	3	109.7	2.2e-6	735 s
Perturbation I				
λ-modes	13	14.9	1.5e-7	106 s
γ-modes	23	10.6	5.0e-7	669 s
α-modes	4	31.9	3.7e-5	438 s
Perturbation II				
λ-modes	13	14.9	1.5e-7	106 s
γ-modes	23	10.6	5.0e-7	679 s
α-modes	6	24.6	3.6e-5	471 s

As the computational time necessary to obtain the different modes with the Krylov-Schur method is very different (for α and γ-modes is much larger than for λ-modes) and near of criticality the eigenfunctions are similar, it is proposed computing α-modes and γ-modes by using an alternative methodology that uses the λ-eigenvectors as an initial approximation. The proposition is initializing, with the solution of the λ-modes problem, a block generalized method that avoids to solve many linear systems. To use the λ-modes eigenvectors as initial guess, a Gram-Schmidt orthogonalization and the generalized Rayleigh-Ritz process (Algorithm 2) is applied. As block method, we have used the two versions of the modified generalized block Newton method: MBNM-GRR and MBNM-GBO. The solution of the linear systems that are needed to be solved with the Newton's method are computed with the GMRES method preconditioned with ILU(0).

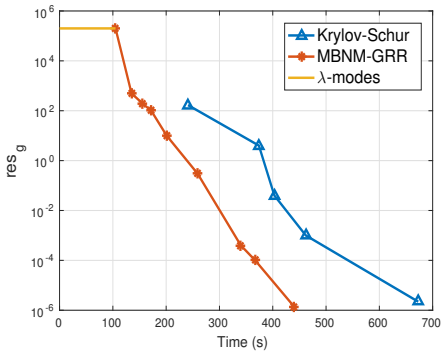
This strategy is compared with the application of the Krylov-Schur method. Many settings have been taken by using the Krylov-Schur method modifying the required tolerance in the ordinary eigenvalue problem (res$_o$), computing the residual errors for the generalized eigenvalue problem (res$_g$) and the computational time necessary to converge the problem. For the block Newton methods, we obtain, in each iteration, the computational time needed for the convergence and their residual error, res$_g$. The time needed to compute the λ-modes to initiate

the block methods has been added in CPU time. Converged λ-modes have been used to initialize the MBNM-GRR with a residual error $\text{res}_g = 10^{-4}$.

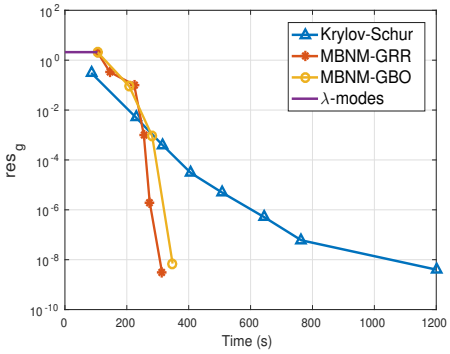
Figure 4.2 shows the convergence histories. For γ-modes, in critical configuration, the block Newton methods are faster than the Krylov-Schur method. In the perturbed configurations, the initial errors obtained with the approximations of λ-modes are larger, but the block-Newton methods are more efficient than the Krylov-Schur method to obtain error lower than $\text{res}_g = 10^{-3}$. Concerning to the block Newton methods, the convergence of MBNM-GRR is slightly faster than the one of MBNM-GBO method. For the α-modes there are more differences. The results for MBNM-GBO are not included since this method does not converge for these modes. For the critical configuration, the MBNM-GRR is more efficient than the Krylov-Schur method. There is a difference of 200 seconds between both methods to obtain a residual error of $\text{res}_g = 10^{-6}$. In Perturbation I, the speed of the method is similar. However, for residual errors lower than $\text{res}_g = 10^{-1}$ the MBNM-GRR computes the solution in less time than Krylov-Schur method. In Perturbation II, the shape of α-modes is more different from the shape of the λ-modes and this makes that the MBNM-GRR is not as fast as Krylov-Schur method.



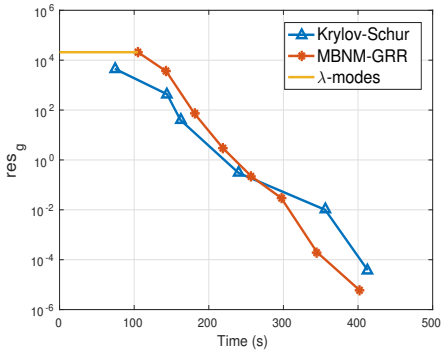
(a) γ-modes: critical configuration



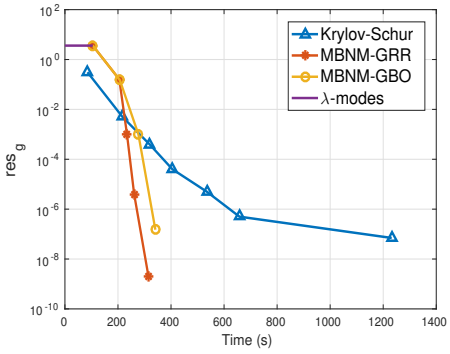
(b) α-modes: critical configuration



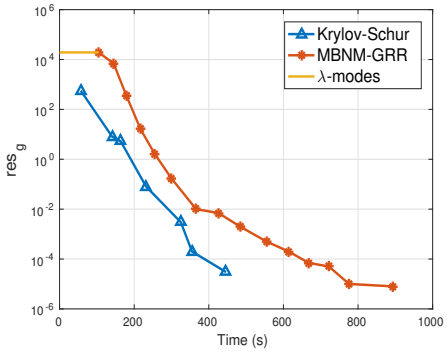
(c) γ-modes: Perturbation I



(d) α-modes: Perturbation I



(e) γ-modes: Perturbation II



(f) α-modes: Perturbation II

Figure 4.2: Residual error against CPU time for the NEACRP reactor with Krylov-Schur, MBNM-GRR and MBNM-GBO methods.

4.2 Initialization techniques

Iterative methods improve their convergence if a suitable initial guess is given. For this purpose, this Section studies several initialization techniques to initialize the block methods with a set of approximate eigenvalues. In particular, it is proposed the λ-modes initialization, several multilevel initializations and a Krylov initialization.

Let us consider the eigenvalue problem (4.1)

$$AX = BX\Lambda\,.$$

By using an iterative method, one typically calculates the successive approximations to the exact solution, X, of the form

$$X^{\text{new}} = G(A, B, X^{\text{old}}),$$

where G is some expression involving the old assignment of X. One natural way to accelerate this process is to get better initial guesses (closer to the solution) at a computational cost as low as possible.

Before the description of initialization strategies, we must remark that only the block methods can be initialized with a set of eigenvectors. In the methods that are presented in the numerical results section, only the Krylov-Schur method is not a solver of this type and consequently, only the first eigenvector is initialized with an all-ones vector.

λ-modes initialization. This initialization can be used to solve the γ and the α-modes problem as we have shown in Section 4.1.8. Numerical results show that the computation of λ-modes is faster than for the rest of modes. This property can be used to obtain an initial approximation for the eigenvalue solver. Moreover, the adjoint λ-modes problem can be computed by using this type of initialization. Usually, the adjoint modes are used together with the direct modes as we show in Chapter 5 for the modal methods. Thus, one can use the direct modes as initial guesses for the adjoint modes.

Even though, the solution of the λ-modes can be used directly as initial guess, it is recommended to project these vectors over the generalized eigenvalue problem with a Rayleigh-Ritz procedure (Algorithm 2) and then, apply the Gram-Schmidt orthogonalization.

Multilevel initialization. Several block multilevel techniques are proposed to obtain good initial approximations of several eigenvectors: *multilevel-mesh*,

multilevel-fem, *multilevel-spn*. All strategies follow the same scheme that is represented in Figure 4.3. First, a small problem, related with the original problem in some way, is defined. Then, this small problem is solved for the largest q eigenvalues with an eigenvalue solver that is not too dependent on the initial guess. In the next step, the obtained solution is projected (or prolonged) onto the original problem. Finally, these vectors are used to initialize some block method.

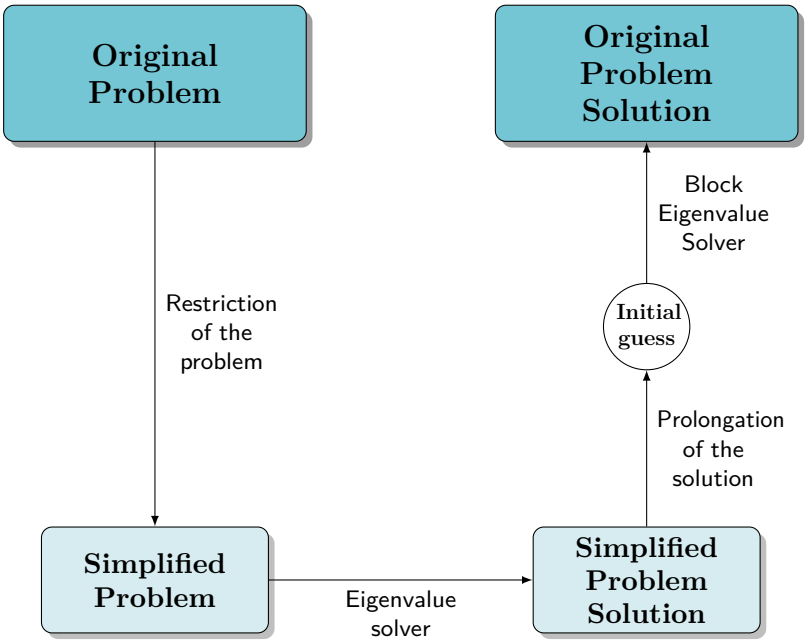


Figure 4.3: Scheme for the multilevel initialization.

A complete description of the multilevel concepts can be found, for instance, in (McCormick, 1987; Sampath and Biros, 2010; Hackbusch, 2013). However, some basic concepts on multilevel techniques are reviewed.

The first technique is based on two meshes (*multilevel-mesh*). The fine mesh, which is the final mesh considered to solve the problem. It comes from the spatial discretization of the problem. And then, a coarse mesh, with less number of cells, constructed by coarsening the fine mesh. This is used to obtain an initial approximation for the problem. This multilevel method will strongly depend on the geometry of the underlying meshes. The application of this strategy for the neutron diffusion equation was published in (Carreño et al., 2017a).

Let us consider the original algebraic partial eigenvalue problem

$$A^f X^f = B^f X^f \Lambda^f, \tag{4.34}$$

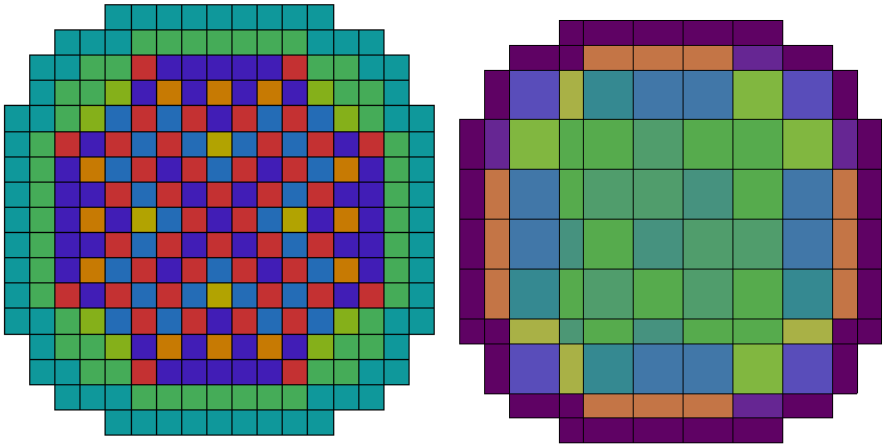
where A^f and B^f are the matrices that arise from the discretization of a given domain V by using the fine mesh V^f. From this mesh a coarser mesh, V^c, is constructed and a new eigenvalue problem

$$A^c X^c = B^c X^c \Lambda^c, \tag{4.35}$$

is considered, where A^c and B^c are the matrices associated with the spatial discretization using the mesh V^c. This algebraic problem has a smaller dimension than the initial one. To assembly matrices A^c and B^c, besides the coarsening of the initial spatial discretization, the cross sections must be homogenized. In each coarse cell the value of each cross section Σ_d is computed as a volume average,

$$\Sigma_d = \frac{1}{V_d} \sum_{i=1}^{N_c} V_{d_i} \Sigma_{d_i}, \tag{4.36}$$

where the coarse cell d is equal to the union of fine cells $d_1, \ldots, N_c$, i.e. $d = \cup_{i=1}^{N_c} d_i$. Σ_{d_i} is the value of the cross section in cell d_i. V_d is the volume of the coarse cell d and V_{d_i} the volume of the cell d_i. Figure 4.4 displays an example of fine mesh and coarse mesh for a two-dimensional benchmark. Each colour represents one different homogenized material.



(a) Fine mesh (b) Coarse mesh

Figure 4.4: Example of meshes for the *multilevel-mesh* initialization.

To use the eigenvectors X^c as an initial guess for the problem associated with the finer grid V^f, we need to define a prolongation operator, $\mathcal{P}$, that interpolates vectors defined on the coarse mesh onto the fine mesh. For that, the solution at the nodes of the coarse problem is interpolated in the nodes of the FEM associated with the original problem. It is performed by setting the same values when both nodes are in the same position, and considering an interpolation for the nodes that appear in the fine mesh but not in the coarse mesh. In the other way, one can define the restriction operator, $\mathcal{R}$.

The second initialization has a similar performance. For structured meshes, given an initial mesh, it is not really difficult to define a coarse mesh from it. However, there are some cases where the problem implies to use other type of meshes and this task can become complicated. As solution, the simplified problem can be defined by considering a lower degree of the polynomial in the finite element method than the original degree of polynomial (*multilevel-fem*). The same mesh is considered, but the simplified problem associated has smaller number of nodes. Note that this initialization only makes sense when the problem is solved with a degree of polynomials higher than 1 in the finite element method. The definition of this simplified problem avoids redefining the cross-sections in each cell.

As the previous case, a prolongation operator must be defined to use the solution of the simplified problem as initial guess for the original problem. This interpolation is the same as the presented for two meshes, where the nodes of the finite element method with a low polynomial degree are interpolated over the nodes in the original problem.

Finally, a multilevel initialization is proposed based on the different approximations of the neutron transport equation (*multilevel-spn*). This type is used to obtain the solution of the $\mathrm{SP_N}$ equations where $N > 1$. The idea behind of this process is using the solutions $U^1, \ldots, U^{(M+1)/2}$ of the $\mathrm{SP_M}$ approximation to initialize the vectors $\hat{U}^1, \ldots, \hat{U}^{(N+1)/2}$ of the $\mathrm{SP_N}$ approximation where $N > M$ as

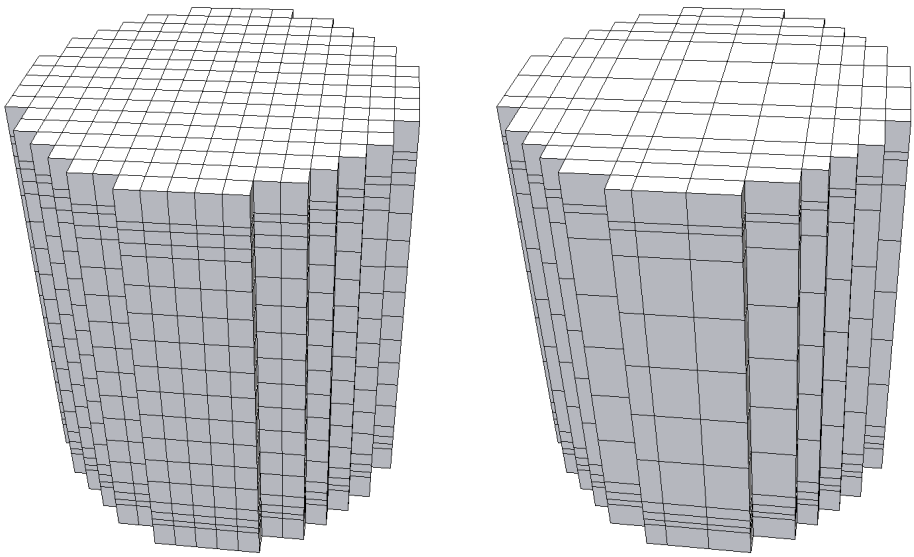
$$\begin{cases} \hat{U}^q = U^q, & \text{if } q \leq (M+1)/2, \\ \hat{U}^q = 0, & \text{if } q > (M+1)/2. \end{cases}$$

Krylov subspace initialization. That estimates the q initial vectors using the Krylov subspace generated by the matrix $B^{-1}A$ acting on an initial vector (we use an all-ones vector). The Arnoldi method has been used to obtain this subspace. The dimension of the subspace depends on the number of required vectors q. Before using that initialization, the resulting system of vectors is orthonormalized by using the modified Gram-Schmidt process. Then, the Rayleigh-Ritz algorithm for the generalized eigenvalue problem is applied.

4.2.1 Numerical results for the initialization techniques

The initialization strategies are studied for several block methods in the computation of 4 λ-modes associated with the neutron diffusion equation for the NEACRP reactor. In particular, the *multilevel-mesh*, the *Krylov* and a *Random* initialization are compared.

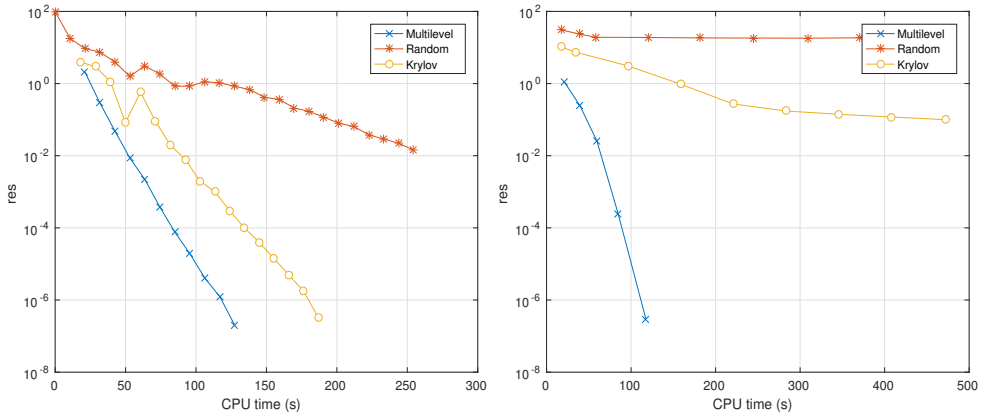
In the multilevel initialization, the Krylov-Schur method with the GMRES method and the ILU(0) preconditioner has been used to compute the solution in the simplified problems, since this solver without initialization converges faster than the rest, for this type of problems. The tolerance for the simplified problem has been $\text{res}_g < 10^{-3}$. Figure 4.5(a) shows the fine mesh used for the spatial discretization to solve the problem and Figure 4.5(a) represents the coarse mesh used to apply the *multilevel-mesh* initialization. In *Krylov* initialization, the dimension of Krylov subspace has been $d_k = 10$. The *Random* initialization generates the q vectors using random numbers on the interval $[-1, 1]$ and then, the Gram-Schmidt orthogonalization and the generalized Rayleigh-Ritz process are applied.



(a) Fine mesh (n. cells = 3978) (b) Coarse mesh (n. cells =1308)

Figure 4.5: Meshes for NEACRP reactor.

Figure 4.6 shows the convergence histories for the BIFPAM and the MBGNM with the different initializations. In the *multilevel-mesh* times, the CPU time to assemble and solve the eigenvalue problem in the coarse mesh, that is 21s, has been included. In the CPU times obtained with the *Krylov* initialization, the time to compute the subspace (18s) has been added. Both graphics reflect that the *multilevel-mesh* initialization, although it takes more time to obtain the initial guess than the other initializations, is a better strategy to initialize the block methods. Moreover, for the Newton's method, it is difficult to obtain a convergent performance without a 'good' initialization.



(a) BIFPAM (b) MGMNM

Figure 4.6: Convergence histories for the BIFPAM and the MGBNM using different initializations for the computation of the λ-modes of the NEACRP problem.

4.3 Preconditioned strategies for linear systems

In general terms, preconditioning is simply transforming the original problem into one which has the same solution, but which is easier to be solved with an iterative solver (Saad, 2003).

The application of one of the eigenvalue solvers proposed in this thesis requires either solving linear systems, in the case of Krylov-Schur method (KSM) and the modified generalized block Newton method (MGBNM); or to use directly a preconditioner, in the generalized Davidson method (GDM) and the block inverse-free preconditioned Arnoldi method (BIFPAM). Usually, the solutions of the linear systems are obtained with the GMRES method (Saad and Schultz, 1986), but the application of a preconditioner is mandatory to improve the

condition number of the matrices, and then, the convergence rate of the GMRES method.

In the BIFPAM, preconditioners for linear systems are applied as one strategy to improve the spectral distribution of matrices. In the GDM, the preconditioner to solve the linear systems is used as an approximation of the matrix because the preconditioners have smaller bandwidth than the original matrices.

The classical preconditioning of matrices derived from FEM discretization are based on an incomplete matrix factorization, such as the ILU decomposition or the ICC decomposition (Saad, 2003). Nevertheless, the computation of such factorizations demands to store the sparse matrices in memory, in addition to the computed preconditioners, which results in large requirements of memory resources. These memory requirements can be lowered by different fill-ins or threshold criteria for the preconditioner, although the minimum memory requirement remains large if a good preconditioner wants to be used.

On the other hand, one can use the Jacobi, the Gauss-Seidel or the SOR method as preconditioners. Although they are efficient, its implementation implies to access to matrix elements and it can be computationally very expensive. However, in this thesis, a block version of the Gauss-Seidel (Saad, 2003), used mainly in parallel computations, has been implemented to develop a block preconditioner for the matrices related to the neutron transport approximations.

Another technique, to precondition the solution of linear systems, is the polynomial preconditioning. This preconditioner, for a linear system with the matrix A, is any polynomial $P = P_n(A)$ normalized to $P(0) = 1$ such that minimizes $\|I - P^{-1}A\|$. For instance, Chebyshev polynomials minimizes this problem when the infinity norm is chosen (Saad, 2003). They are very common because they present better performance in parallel computations than, for example, the block Gauss-Seidel (Adams et al., 2003).

Recently, the multilevel methods are becoming increasingly popular. Different coarse levels are obtained either from a linear finite element discretization on the original grid (Kronbichler and Wall, 2018) or as the multigrid methods by combining simple iterative schemes on a hierarchy of coarser meshes (Bastian et al., 2019) or with several levels of energy groups (Cornejo et al., 2019). The smoother proposed in (Kronbichler and Wall, 2018) has been the Chebyshev smoother and in the work (Bastian et al., 2019), the authors used block smoothers such as the block Jacobi, the block Gauss-Seidel and the block SOR.

In this Section, classical block preconditioners for linear systems are briefly reviewed and more sophisticated strategies to reduce the computational memory resources based on multilevel preconditioners are proposed.

4.3.1 Block preconditioners

These preconditioners are very common in parallel computation, but they also are a part of multilevel methods such as the one described in (Bastian et al., 2012). They are designed for preconditioning block systems of the form

$$\begin{pmatrix} B_{11} & \cdots & B_{1n_b} \\ \vdots & & \vdots \\ B_{n_b1} & \cdots & B_{n_bn_b} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_{n_b} \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_{n_b} \end{pmatrix},$$

where n_b is the number of blocks. In our case, they are interesting due to the block structure of the matrices that are obtained from the discretization of the multigroup neutron diffusion equation and $\mathrm{SP_N}$ equations. Two classical block methods are:

Block Jacobi (BJACOBI). This is applied with one iteration of the traditional Jacobi method for solving linear systems where the elements are substituted by the blocks of the matrix. Algorithm 7 exposes an outline of a possible implementation. Note that, the division of the diagonal elements in traditional method is substituted by the multiplication of the inverse of the diagonal block matrix. This inverse, in some cases, will be substituted directly by a preconditioner of the block matrix and, in other cases, a linear system is solved.

Algorithm 7 Block Jacobi preconditioner

Input: Matrix B and vector $x = [x_1; \ldots; x_{nb}]$.
Output: Vector $y = [y_1; \ldots; y_{nb}]$.

1: Make $y^{\mathrm{old}} = x$
2: **for** $i = 1$ to nb **do**
3: Compute

$$y_i = B_{ii}^{-1}\Big(x_i - \sum_{\substack{j=1 \\ j\neq i}}^{n_b} B_{ij} y_j^{\mathrm{old}}\Big)$$

4: **end for**

Block Gauss-Seidel (BGS). Following a similar strategy as the previous case leads to the block Gauss-Seidel preconditioner. This method typically converges

faster than the Block Jacobi method, so the preconditioner is better. From the storage point of view, Gauss-Seidel is more economical because the new approximation can be overwritten over the same vector. The implementation can be observed in Algorithm 8

Algorithm 8 Block Gauss-Seidel preconditioner

Input: Matrix B and vector $x = [x_1; \ldots; x_{nb}]$.
Output: Vector $y = [y_1; \ldots; y_{nb}]$.

1: Make $y^{\text{old}} = x$
2: **for** $i = 1$ to nb **do**
3: Compute

$$y_i = B_{ii}^{-1}(x_i - \sum_{j=1}^{i-1} B_{ij} y_j - \sum_{j=i+1}^{nb} B_{ij} y_j^{\text{old}})$$

4: **end for**

4.3.2 Multilevel preconditioners

The multilevel preconditioners are based on the classical V-cycle multigrid method with two meshes (the original mesh and a coarser mesh) known in many works as geometric multigrid preconditioner (GMG). This strategy is characterized by the method used to smoother the solution and by the definition of simplified problem. The smoothing is done with a Gauss-Seidel iterative method, similar to the strategy developed in (Knyazev and Neymeyr, 2003) when we have access to the matrix elements. In other cases, such as in parallel computations or in matrix-free implementations, Chebyshev polynomial smoothers are used. Now, depending on the definition of the simplified problem, we will distinguish several preconditioners.

Geometric Multigrid Preconditioner (GMG). In this case, as the original multigrid method, the simplified problem is obtained from the discretization of the problem but with a coarser mesh. If the *multilevel-mesh* initialization is used to estimate an initial guess for the method, this preconditioner can be implemented without extra cost by using the matrices and operators that have already been defined previously. By using the notation introduced in Section 4.2 in the *multilevel-mesh* initialization, the application of this preconditioner can be summarized in the Algorithm 9. In our implementation, the number of iterations for the smoother, its_s, is set to $its_s = 3$. The application of this preconditioner for the neutron diffusion equation was published in (Carreño et al., 2018a).

Algorithm 9 Geometric multigrid preconditioner (GMG)

Input: Vector x, matrix B and matrix in the coarse mesh B_c.
Output: Vector $y = P^{-1}x$ con P preconditioner of B.

1: Pre-smooth with its_s iterations of Gauss-Seidel on $By = x$
(Initialize the iterative method with $y = 0$)
2: Restrict the residual $r = By - x$ to the coarse mesh by $r_c = \mathcal{R}(r)$
3: Solve $B_c e_c = r_c$
4: Prolongate e_c by $e = \mathcal{P}(e_c)$
5: Correct $y = y + e$
6: Post-smooth with its_s iterations of Gauss-Seidel on $By = x$
(Initialize the iterative method with y)

Multilevel preconditioner from FEM and SPN (MLFE or MLSPN). In a similar way as it is done for the multilevel initialization technique, one can use the algorithm of the geometric multigrid preconditioner, but in this case the simplified problem is not related to the original problem by coarsening the original mesh. In this way, one can define a multilevel preconditioner with several degrees in the polynomial of the FEM (MLFE), with several SP_N approximations (MLSPN) or with a combination of both (MLFE-SPN). The matrices, the restriction and prolongation operators are the same that are ones defined for *multilevel-fem* and *multilevel-spn* (Section 4.2).

4.3.3 Block preconditioner for the MGBNM

Alternative to the preconditioners defined previously, a special preconditioner is studied for the modified generalized block Newton method to solve the linear systems. The matrix that appears in the linear systems arising in this method has a special block structure, with one diagonal block equal to zero, which makes it unfeasible to apply the above preconditioning methods. Thus, several different strategies for preconditioning this type of matrices are proposed. This methodology was published in (Carreño et al., 2019b).

The first choice for a preconditioner is assembling the matrix involved in the system

$$S = \begin{pmatrix} A - \delta_m B & BZ \\ Z^{\mathsf{T}} & 0 \end{pmatrix},$$

and then, constructing the full preconditioner associated with the matrix. The ILU(0) preconditioner is used since S is a non-symmetric matrix. Nevertheless, there are no significant differences if the preconditioner obtained for the matrix associated with the first eigenvalue is used for all eigenvalues in the same iteration because in the matrix S only changes the value of δ_m and usually, the eigenvalues in reactor problems are clustered. This preconditioner is denoted by P.

To devise an alternative preconditioner without the necessity of assembling the matrix S, we write the explicit inverse of A, by using its block structure,

$$S^{-1} = \begin{pmatrix} R^{-1}(I - C_1(C_2^{\mathsf{T}}C_1)^{-1}C_2^{\mathsf{T}}) & R^{-1}C_1(C_2^{\mathsf{T}}C_1) \\ (C_2^{\mathsf{T}}C_1)^{-1}C_2^{\mathsf{T}} & -(C_2^{\mathsf{T}}C_1)^{-1} \end{pmatrix},$$

where

$$R = A - \delta_m B, \quad C_1 = BZ, \quad C_2^{\mathsf{T}} = Z^{\mathsf{T}}R^{-1}.$$

We desire a preconditioner for S by suitably approximating S^{-1}. Let us call P_R a preconditioner for R, where $\delta_m := \delta_1$. For instance, $P_R = (LU)^{-1}$, where L, U are the incomplete L and U factors of R. Thus, we can define, after setting $C_2^{\mathsf{T}} = Z^{\mathsf{T}}P_R$, the preconditioner of S as

$$\hat{P}_R = \begin{pmatrix} P_R(I - C_1(C_2^{\mathsf{T}}C_1)^{-1}C_2^{\mathsf{T}}) & P_R C_1(C_2^{\mathsf{T}}C_1) \\ (C_2^{\mathsf{T}}C_1)^{-1}C_2^{\mathsf{T}} & -(C_2^{\mathsf{T}}C_1)^{-1} \end{pmatrix}.$$

The previous preconditioner does not need to assemble the entire matrix S, but it needs to assemble the matrix R to build its ILU(0) preconditioner. Therefore, the next alternative that we propose is using a preconditioner of $-B$ instead of the $R = A - \delta_1 B$. This preconditioner works well because in the discretization process, the B matrix comes from the discretization of the differential matrix that has the gradient operators and the diffusion terms. In addition, in nuclear calculations, δ_1 is near 1.0. Thus, we can build a preconditioner of $-B$ instead of the matrix R. We denote by $\hat{P}_B$ to the preconditioner $\hat{P}_R$ where the preconditioner of $-B$ is used to precondition the block R.

4.3.4 Numerical results for the block methods and preconditioner strategies

The NEACRP reactor (Appendix B.4) is chosen to study the performance of the block methods and the efficiency of different preconditioners to compute the λ-modes associated with the neutron diffusion equation. First, the performance of the BIFPAM method is studied. Then, the block preconditioners for the MGBNM are analyzed. Finally, the hybrid method results are shown and all eigenvalue solvers are compared.

Residual errors have been, for all methods, $\text{res}_g < 10^{-6}$. The initial guess has been obtained from *multilevel-mesh* initialization. Figure 4.5(a) shows the fine mesh considered to solve the problem and Figure 4.5(b) shows the coarse mesh to define the simplified problem. They have 3978 and 1308 cells, respectively. For cubic polynomials in the FEM, the problem associated with the fine mesh has 230 120 degrees of freedom and the one associated with the coarse mesh has 78 440 degrees of freedom. The number of eigenvalues computed has been 4.

Numerical results compare the BIFPAM by using the ILU(0) and GMG preconditioners and without preconditioning, for dimensions of the Krylov subspace $d_k = 4$ and $d_k = 8$. The GMG preconditioner is constructed with the same mesh as the *multilevel-mesh* initialization. Figure 4.7 displays the convergence histories for the configurations considered. It is observed that the preconditioner improves the rates of convergence and attenuates the oscillations produced in the convergence histories. Between the different preconditioners used, the Figure shows that the ILU preconditioner (with $d_k = 8$) and the GMG preconditioner (with $d_k = 4$) reach the tolerance almost at the same time. However, note that the GMG preconditioner works efficiently even when the dimension of the Krylov subspace is $d_k = 4$, which implies to use less computational memory. This is because when we apply the GMG preconditioner, there are not big differences in the number of iterations and it is expensive to apply. Thus, better results in terms of CPU time are obtained with a low Krylov subspace. In opposite, the ILU preconditioner is a cheaper preconditioner, but the number of iterations to reach the convergence differs greatly between two Krylov subspaces with different dimensions. The use of these preconditioners with the other values considered for the dimension (d_k) is less efficient.

On the other hand, one can consider the MGBNM to solve the eigenvalue problem. The performance of the block preconditioners for the MGBNM described in Section 4.3.3 is tested. First, we show the convergence history of the MGBNM to obtain the solution of the eigenvalue problem. Figure 4.8 shows the number of

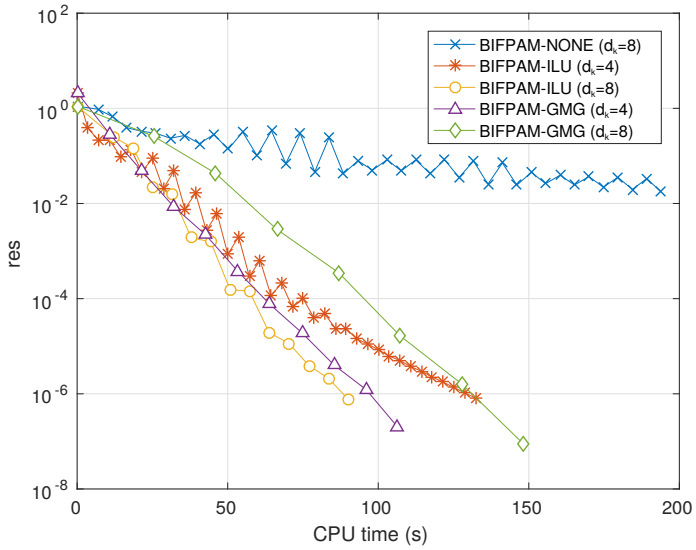


Figure 4.7: Residual errors against CPU time (s) of the BIFPAM with different configurations for the NEACRP problem.

iterations against the residual error for the NEACRP reactor. It is observed that the MGBNM only needs 4 iterations to reach a minor residual error than 10^{-6}.

The most relevant data to compare the preconditioners considered in this work are exposed in Table 4.3. This includes the average number of iterations that GMRES method needs to reach the residual error, the time for assembling the matrices and building the preconditioners (Setup time (s)) and the computational memory. The first row shows the data of applying directly the ILU(0) preconditioner of S. Even though the number of iterations is not very high, the time spent to assemble the matrix and to construct the preconditioner increases the total CPU time considerably. It is necessary to build in each iteration a new preconditioner for S because of the columns related to the block Z change considerably in each updating. The second row displays these data related to $\hat{P}_R$ that uses the ILU preconditioner for approximating the inverse of $R = A - \lambda_1 B$. In this case, only it is necessary to assemble once the matrix $R = A - \lambda_1 B$ in the first iteration. The number of iterations of the GMRES preconditioned with $\hat{P}_R$ is larger than in the previous case, but the total CPU time of using this block preconditioner has been reduced in 26s in comparison with the full preconditioner P. The third row represents the data related to $\hat{P}_R$, but in this case, we have used the geometric multigrid (GMG) preconditioner to approximate the inverse of R. The

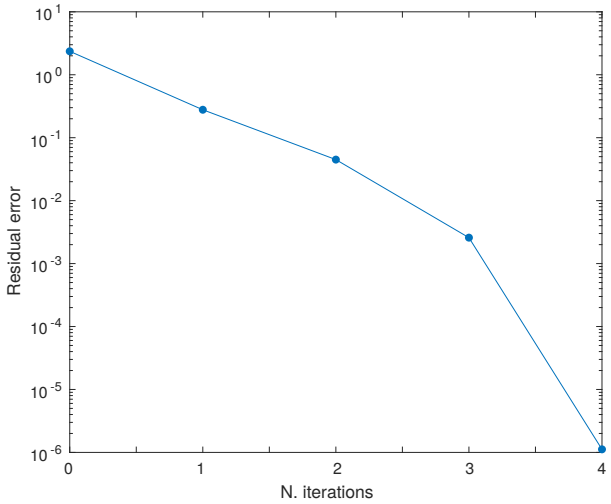


Figure 4.8: Convergence history of the MGBNM for the NEACRP reactor.

results show that in spite of the total number of iterations and the setup time are much lower for the GMG, the total computational time is much higher. This is due to the application of the GMG preconditioner is more expensive than the application of the ILU preconditioner. The next results are obtained by using the block preconditioner, but approximating the R^{-1} by the ILU(0) preconditioner of $-B$ ($\hat{P}_B$).

If all preconditioners for the MGBNM are compared between them, the number of iterations increases when worse approximations of the inverse of R are considered, but the setup time that needs each preconditioner becomes smaller. Moreover, the maximum CPU memory is also reduced significantly. The total CPU times show that the block preconditioner ($\hat{P}$), in all its versions, improves the times obtained of applying directly the ILU preconditioner of S. Between the possibilities for obtaining a preconditioner of R, there are no big differences in the computational times but there is an important saving up in the computational memory. The best results are obtained by the $\hat{P}_B$ if the computational memory consumption is taken into account.

Figure 4.9 compares the convergence histories of the MGBNM with the ILU preconditioner for S and the BIFPAM with the ILU and GMG preconditioner. It is deduced that the desired tolerance is reached quicker with the MGBNM. However, we would like to highlight that the convergence behaviour of BIFPAM-

Table 4.3: Data obtained of using different preconditioners for the MGBNM in the NEACRP reactor.

Precond.	Mean Its. GMRES	Setup Time	Total CPU time	Max. CPU memory
P^{ILU}	18.1	48.0 s	96.8 s	2062 Mb
$\hat{P}_R{}^{\mathrm{ILU}}$	21.5	6.6 s	70.0 s	1418 Mb
$\hat{P}_R{}^{\mathrm{GMG}}$	12.2	2.5 s	151.3 s	1118 Mb
$\hat{P}_B{}^{\mathrm{ILU}}$	24.5	4.4 s	73.2 s	787 Mb

ILU is very similar to the one of BIFPAM-GMG and when the residual becomes smaller the convergence of the Newton method becomes faster.

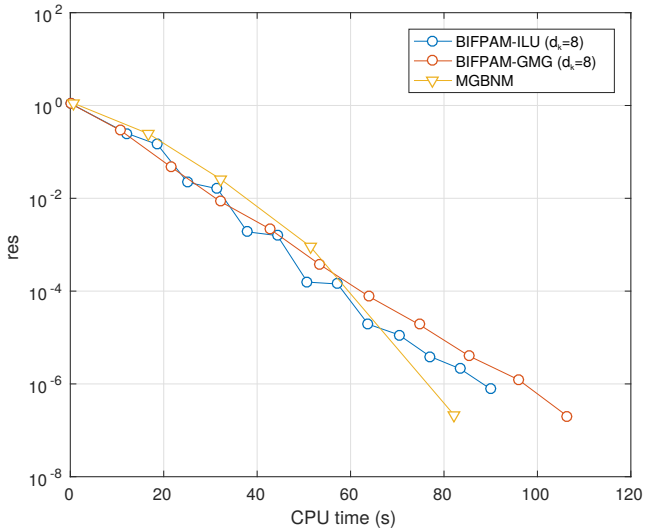


Figure 4.9: Convergence history for the fourth dominant eigenvalues of the NEACRP problem using the MGBNM and the BIFPAM.

Thus, it is proposed to initialize the algorithm with the BIFPAM method until $\mathrm{res}_g = 10^{-2}$ and then, the MGBNM is applied. The BIFPAM has been set with the ILU preconditioner and dimension of the Krylov subspace $m = 8$. Figure 4.10 compares the hybrid scheme with the MGBNM and the BIFPAM with ILU preconditioner. It is showed that the hybrid algorithm is an efficient scheme to compute several eigenvalues of the NEACRP problem.

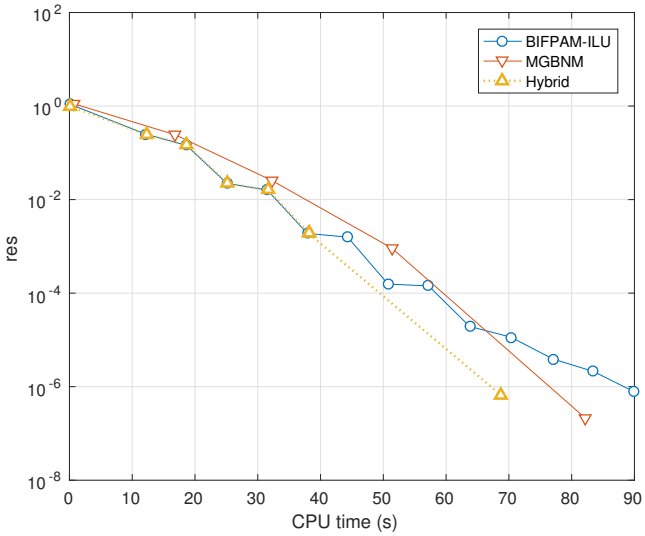


Figure 4.10: Convergence history of the BIFPAM with ILU preconditioner, the MGBNM and the hybrid method.

4.4 Matrix-free implementation

The computation of eigenvalues by using the previous algorithms evaluate the discrete operators through matrix-vector products. This makes that these operations take up the most time-consuming component in the code. On the other hand, one of the main inconvenient in the finite element method is the quantity of memory used to store the matrix elements, in spite of they are saved in a sparse way. This waste increases considerably as more accurate approximations of the neutron transport equations are used.

As alternative to assemble and save the matrices, matrix-free techniques are developed where the matrix-vector multiplications are computed on the fly in a cell-based interface. Instead of assembling a global sparse matrix, matrix-free strategy stores the unit cell shape function, the enumeration of degrees of freedom, and the map that transforms the unit cell to the real cell. Recently, cell-based strategies without explicit matrix storage have been considered for GPU programming (Komatitsch et al., 2010).

For instance, we can consider that a finite element Galerkin approximation that leads to the block matrix A takes a vector u as input and computes the integrals of the operator multiplied by trial functions, and the output vector is v. The

operation can be expressed as a sum of N_c cell-based operations,

$$v = Au = \sum_{c=1}^{N_c} M_c^{\mathsf{T}} A^{N_c} M_c u, \tag{4.37}$$

where M_c denotes the matrix that defines the location of cell-related degrees of freedom in the global vector and A^c denotes the submatrix of A on cell N_c. This sum is optimized through *sum-factorization*. Details about the implementation are explained in (Kronbichler and Kormann, 2012). This strategy does not only minimizes the memory used by the matrix elements, but it also can improve matrix-vector multiplication runtimes for higher orders of FEM, as we show in the numerical results.

The main difficulty of the matrix-free implementation is to obtain efficient algebraic solvers that only use matrix-vector multiplications. The eigenvalue solvers are based on matrix-vector multiplications, but classical preconditioners need to access to the matrix elements. Different strategies of preconditioning are used according to the implementation of the matrices involved in the problem.

In this thesis, three matrix storage schemes are used. The first one, allocated all the block matrices in *CRS* format (Saad, 2003). For this implementation, one can use any type of preconditioner. The second one stores only the diagonal block matrices of B in a sparse way to permit the computation of an incomplete LU factorization of these blocks. The rest of the blocks are implemented with the matrix-free operator (*non-diagonal*). This type of implementation is justified because the block preconditioners only need to apply a preconditioner or to solve a system related to the diagonal block matrices. For this case, a block preconditioner is used such as the block Gauss-Seidel where each inverse of the diagonal block is substituted either by the ILU(0) preconditioner (BGS-ILU) or by solving linear systems with the conjugate gradient and the ILU preconditioner (BGS-CG-ILU). Finally, all block matrices can be implemented with the matrix-free technique in the *full matrix-free* scheme. In this last case, one can choose a block preconditioner where their inverse diagonal blocks are solved with the conjugate gradient method preconditioned with a multilevel preconditioner. The *multilevel-fem* preconditioner has to be smoother with a Chebyshev polynomial that it does not need to access to the matrix elements. This preconditioner is called BGS-CG-MLFE. The matrix-free performance for the multigroup neutron diffusion equation was published in (Carreño et al., 2019a) and for the $\mathrm{SP_N}$ equations in (Vidal-Ferràndiz et al., 2019).

4.4.1 Numerical results for the matrix-free performance

In this Section, the matrix-free performance for the different block eigenvalue solvers are tested and compared for the solution of the λ-modes associated with the neutron diffusion equation. As the problems obtained from the discretization of the neutron diffusion approximation are not excessively large, only the *CSR* and the *non-diagonal* allocation are compared in the NEACRP reactor and more realistic reactor: the Ringhals reactor.

NEACRP reactor

In this part, the NEACRP reactor (Appendix B.4) is used to test the matrix-free implementation. In particular, the *non-diagonal* matrix-free implementation is compared against the *CSR* allocation. Note that, the algorithms of the methods are not changed, but different preconditioners must be considered. Polynomials of degree 3 is used in the FEM.

For the BIFPAM, alternative to use the ILU or the GMG preconditioner in the BIFPAM of the matrix $R = A - \lambda_1 B$, one can consider the preconditioner of $-B$. Thus, one avoids to assemble the matrix A. As it is explained for the block preconditioner of the MGBNM, this approximation works because the B has the discretization of the gradient operators. Thus, the block Gauss-Seidel preconditioner (BGS) is considered to the matrix B and the diagonal blocks, $(B_{11})^{-1}$ and $(B_{22})^{-1}$ are approximated by their ILU preconditioners. In this way, one can use the *non-diagonal* matrix-free strategy. Table 4.4 shows the reduction of the memory for allocating the matrices with the matrix-free implementation, but also the reduction of the CPU times because the matrix-vector multiplications is also accelerated.

Table 4.4: Data of the matrix-free performance for the block inverse-free preconditioned Arnoldi method in the NEACRP reactor.

Allocation	**Precond.**	$\mathbf{d_k}$	**N. its.**	**Matrix CPU Mem.**	**CPU Time**
CSR	ILU	8	15	825 Mb	91 s
CSR	GMG	4	11	825 Mb	108 s
Non-diagonal	BGS	8	11	330 Mb	48 s

For the MGBNM, one can also use the block Gauss-Seidel to precondition the matrix B instead of the ILU preconditioner. Table 4.5 shows the timings and the memory spent in the matrix allocation by using the matrix-free technique or

without using this strategy. The results show that not only the matrix memory consumption and the time to assemble are reduced but also the time spent to compute the matrix-vector products. That implies that the matrix-free strategy reduces the total CPU time by about 30%.

Table 4.5: Data obtained of using different matrix implementations for the MGBNM in the NEACRP reactor.

Allocation	Precond.	Matrix CPU Memory	Setup CPU Time	Total CPU Time
CSR	ILU	787 Mb	4 s	73 s
CSR	BGS	687 Mb	1 s	74 s
Non-diagonal	BGS	319 Mb	1 s	52 s

Finally, the *non-diagonal* matrix-free and *CSR* implementation is compared for all eigenvalue solvers described in this thesis. The preconditioner used for all block eigenvalue solvers has been the block Gauss-Seidel with ILU preconditioner in their diagonal blocks. In the Krylov-Schur method (KSM), the preconditioners for the GMRES method has been simply the ILU factorization. The initialization for the block methods has been the *multilevel-mesh* initialization.

Table 4.6 displays the computational times needed by the code to compute several sets of eigenvalues. The *non-diagonal* implementation, in addition to reducing the CPU memory, it also decreases the CPU times, especially for the methods that apply more matrix-vector products. In the matrix-free implementation, the best results are obtained for the hybrid method. Even though, for a small number of eigenvalues, the BIFPAM is also very efficient.

Table 4.6: Computational times (s) obtained for the NEACRP reactor using the KSM method, the GDM, the BIFPAM, the MGBNM and the Hybrid method for different set of eigenvalues q.

Allocation	q	PIM	KSM	GDM	BIFPAM	MGBNM	Hybrid
CSR	1	241	29	32	30	66	27
Non-diagonal	1	146	32	26	20	43	20
CSR	4	-	64	107	99	119	82
Non-diagonal	4	-	72	92	57	80	53
CSR	6	-	93	166	245	125	118
Non-diagonal	6	-	106	135	131	82	78

Ringhals reactor

A critical configuration of Ringhals reactor is considered for a realistic application of the block methods and the matrix-free implementation for the λ-modes problem associated with the neutron diffusion equation. Particularly, we have chosen the C9 point of the Ringhals I stability benchmark, which corresponds to a point of operation that degenerated in an out-of-phase oscillation (Lefvert, 1996). Its geometry, with 19 656 cells, is represented in Figure 4.11. The finite element degree considered have been 3 to obtain a total number of 1 106 180 degrees of freedom.

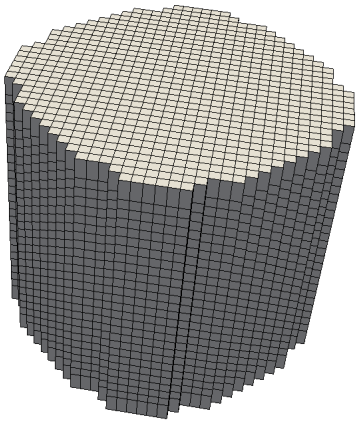


Figure 4.11: Mesh for Ringhals reactor

The BIFPAM method has been used to compute 4 eigenvalues. The tolerance in the generalized eigenvalue problem has been $\text{res}_g < 10^{-6}$. The preconditioner has been the block Gauss-Seidel where the diagonal blocks are substituted by the ILU preconditioner. Two types of matrix implementations are compared: the *CSR* and the *non-diagonal* strategy. Moreover, some values of the dimension of the Krylov subspace d_k are tested. Table 4.7 shows the most relevant data. In the comparison of the dimensions, it is observed that using a high number of vectors improves the convergence of the BIFPAM. Between the matrix implementations, the Table shows that the *non-diagonal* implementation improves the computational efficiency of the method.

Now, the performance of the MGBNM is tested for several block preconditioners and matrix allocations to compute 4 eigenvalues and with $\text{res}_g < 10^{-6}$. Table 4.8 collects the average number of iterations for the GMRES method for each iteration of MGBNM, the time to assemble the matrices and build the precondi-

Table 4.7: Data of the matrix-free performance for the block inverse-free preconditioned Arnoldi method in the Ringhals reactor.

Allocation	d_k	N. its.	Matrix CPU Memory	CPU Time
CSR	4	26	4.5 Gb	1465 s
CSR	8	21	4.5 Gb	700 s
Non-diagonal	4	26	1.8 Gb	845 s
Non-diagonal	8	21	1.8 Gb	418 s

tioners, the total time of the MGBNM to reach the tolerance and the maximum computational memory requested to assemble the matrices. This Table deduces similar conclusions as the ones obtained for the NEACRP reactor. The number of iterations is not reduced when block preconditioners are used, but the total CPU time and the maximum memory decrease considerably. For the Ringhals reactor, the most efficient option, in terms of computational memory, is also to apply the block preconditioner $\hat{P}_B{}^{\text{BGS-ILU}}$ with the *non-diagonal* matrix implementation. As the size of this reactor is much larger, the differences between the preconditioners for computational memory are much higher.

Table 4.8: Data obtained of using different preconditioners for the Ringhals reactor.

Precond.	Allocation	Its. GMRES	CPU memory	Setup Time	Total CPU time
P^{ILU}	*CSR*	71.5	12.5 Gb	155 s	693 s
$\hat{P}_R{}^{\text{ILU}}$	*CSR*	81.0	9.3 Gb	39 s	562 s
$\hat{P}_B{}^{\text{ILU}}$	*CSR*	85.2	6.2 Gb	36 s	591 s
$\hat{P}_B{}^{\text{BGS-ILU}}$	*CSR*	88.2	5.5 Gb	26 s	625 s
$\hat{P}_B{}^{\text{BGS-ILU}}$	*Non-diagonal*	88.2	3.7 Gb	8 s	523 s

Finally, Figure 4.12 compares all eigenvalue solvers with the *non-diagonal* matrix implementation to compute one eigenvalue. The BIFPAM and the generalized Davidson method (GDM) have been implemented with the BGS-ILU preconditioner and the MBGNM with the $\hat{P}_B{}^{\text{BGS-ILU}}$ preconditioner. The *multilevel-mesh* initialization is set for these block methods. In the CPU time, the time to assemble the matrices and to construct the preconditioners are included. The results show that the hybrid and the MGBNM are the most efficient methodd in terms of the computational time to compute one λ-mode.

Figure 4.13 shows the convergence histories for the BIFPAM, the MGBNM and the Hybrid method. One can check, in this case, that the rate of convergence of MGBNM is higher for low residuals in comparison with the BIFPAM. However, the hybridization of BIFPAM-MGBNM (hybrid method) gives a more efficient solver than applying each one separately. The Krylov-Schur method and the Generalized Davidson method are not included because they use a deflation technique and the global residual error is not computed by SLEPc in each iteration. However, using both methods with a tolerance of $\text{res}_g < 10^{-6}$ leads to computational times of 575 s for the Krylov-Schur method and 866 s for the Generalized-Davidson method. Both values are higher than the CPU time obtained with the hybrid method.

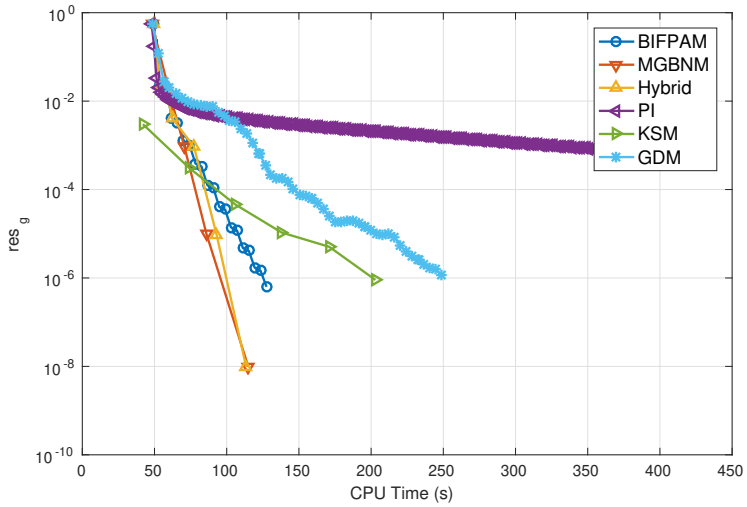


Figure 4.12: Convergence histories for the block eigenvalue solvers in the computation of 1 eigenvalue for the Ringhals reactor.

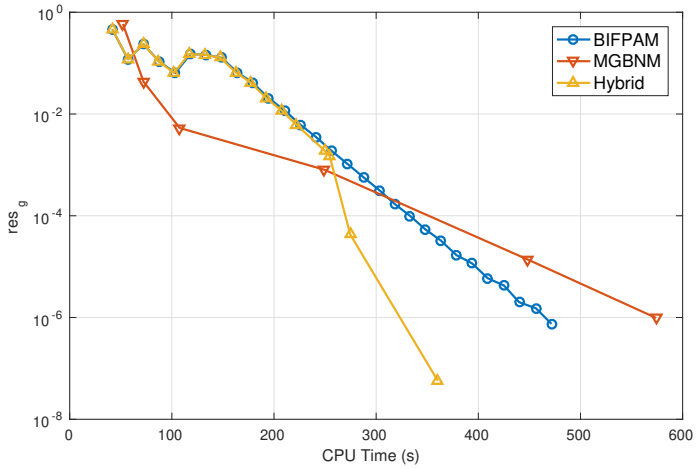


Figure 4.13: Convergence histories for the block eigenvalue solvers in the computation of 4 eigenvalue for the Ringhals reactor.

4.5 Numerical results to compute the λ-modes associated with the SP_N equations

This Section analyzes the methodology proposed to compute the λ-modes associated with the SP_N equations in the C5G7 benchmark (Appendix B.5). In particular, it is considered two versions of the two-dimensional case and the three-dimensional case. The finite element discretization is studied in Chapter 3 (Section 3.4.4) for several values of finite element degrees and meshes. In the two-dimensional and three-dimensional case, one obtains an eigenvalue problem with the block structure

$$\begin{pmatrix} A_{11} & \cdots & A_{17} \\ A_{21} & \cdots & A_{27} \\ A_{31} & \cdots & A_{37} \\ A_{41} & \cdots & A_{47} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \lambda \begin{pmatrix} B_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ B_{21} & B_{22} & 0 & 0 & 0 & 0 & 0 \\ B_{31} & B_{32} & B_{33} & 0 & 0 & 0 & 0 \\ B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & 0 & 0 \\ 0 & 0 & 0 & B_{54} & B_{55} & B_{56} & 0 \\ 0 & 0 & 0 & 0 & B_{65} & B_{66} & B_{67} \\ 0 & 0 & 0 & 0 & 0 & B_{76} & B_{77} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix}. \tag{4.38}$$

The calculations compare the different eigenvalue solvers described in this thesis: the power iteration (PIM), the Krylov-Schur (KSM), the Generalized Davidson (GDM), the block inverse-free preconditioned Arnoldi method (BIFPAM), the

modified generalized block Newton method (MGBNM) and the Hybrid method. The problems are solved with a tolerance of $\text{res}_g < 10^{-7}$. Note that, the size of the problem increases considerably with respect to the two energy groups diffusion approximation. For that reason, the matrices are not full allocated explicitly and *non-diagonal* matrix implementation is used. Thus, the preconditioner for the eigenvalue solvers has been the block Gauss-Seidel preconditioner (BGS) where the inverse of each diagonal block is approximated by solving linear systems with the conjugate gradient method preconditioned with the ILU(0) factorization and residual tolerance of 10^{-5}. It is denoted as BGS-CG-ILU preconditioner.

4.5.1 2D-C5G7 reactor

In the following computations, $r_r = 1$ (Figure 4.14) and $p = 2$ has chosen to obtain an eigenvalue problem of the form (4.38) where each block has a size of 116 009 degrees of freedom for the SP_1 approximation and 232 018 degrees of freedom for the SP_3 approximations. The type of discretization in the pin structure and the degree of the polynomial in the FEM has been discussed in Section 3.4.4.

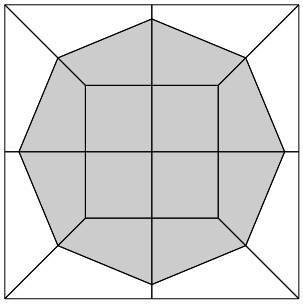


Figure 4.14: Mesh $r_r = 1$ considered for the pin structure.

To check the efficiency of the multilevel initialization the BIFPAM method is used. Table 4.9 displays the residual error and the performance of the BIFPAM solver for different initialization strategies: *Random*, that generates the initial vectors by using random numbers on the interval $[-1,\ 1]$, the *Krylov subspace*, the *multilevel-fem* and the *multilevel-spn* initialization . The *multilevel-mesh* has not been considered in this case due to type of mesh used in the spatial discretization. Note that for SP_1 and SP_3, the *multilevel-fem* with $p = 1$ is considered, but the *multilevel-spn* initialization only has sense for the computation of SP_3 equations. The simplified initialization problems are solved with the Generalized Davidson method until a tolerance of 10^{-4}.

The results conclude that the proposed *multilevel-fem-spn* initialization (SP_1, $p = 1$) minimizes the initialization error taking only 6 seconds. It must be noted that the multilevel SP_3 initialization does not improve the initialization error of the multilevel SP_1 initialization to solve the SP_3 problem. Then, it does not worth the computational overhead. Furthermore, the *Krylov Subspace* initialization takes more CPU time than the SP_1, $p = 1$ initialization and it gets one order of magnitude less accuracy. *Rando*m initialization does not spend any CPU time to obtain the initial guess but this initialization has a large error.

Table 4.9: Performance results of different initialization procedures to compute 1 eigenvalue with BIFPAM solver by using $r_r = 1$ and $p = 2$.

Eq.	Init.	Init. Error	Init. Time	CPU Time
SP_1	*Random*	5.04	0 s	56 s
	Krylov Subspace	2.0e-2	7 s	51 s
	Multilevel (SP_1, $p = 1$)	2.1e-3	6 s	25 s
SP_3	*Random*	5.06	0 s	104 s
	Krylov Subspace	2.3e-2	24 s	102 s
	Multilevel (SP_1, $p = 1$)	2.5e-3	6 s	65 s
	Multilevel (SP_3, $p = 1$)	2.6e-3	11 s	71 s

Tables 4.10 and 4.11 show the performance of the different eigenvalue solvers for SP_1 and SP_3 problems to obtain a $\text{res}_g < 10^{-7}$. The initial guess is obtained from the *multilevel-fem-spn* initialization with SP_1 and $p = 1$. Note that the SP_1 problem uses about 1200 Mb of RAM and the SP_3 case uses about 2500 Mb. The 6 seconds of initialization are included in the CPU times in Tables. The numerical results show that the proposed BIFPAM method with the *multilevel-fem-spn* initialization is more efficient than the other methods studied for one eigenvalue calculations, even though this problem is advantageous to the power iteration method as $\delta = 0.763$. These Tables also show that the proposed multilevel initialization is also convenient for the rest of methods. Note that for the computation of the λ-modes associated with the SP_N equations the hybrid method does not improve the results of BIFPAM. This is because the rate of convergence for the MGBNM (for low residuals) is not higher that this value for the BIFPAM, contrary to the two energy group diffusion case. Even though, the hybrid method with initialization does improve the results of the Krylov-Schur method, the Generalized Davidson and the power iteration method.

Figure 4.15 shows the residual norm, $\text{res}_g(k)$, at the k-th iteration for the first eigenvalue. Figure 4.16 also shows the convergence history of $\text{res}_g(k)$ for the SP_3 problem. In these Figures it can be noted that the proposed initialization reduces

Table 4.10: Performance results for different eigenvalue solvers for SP_1 problem with $r_r = 1$ and $p = 2$.

Solver Method	Init	n eigs.	CPU Time	its
Power Iteration	No	1	163 s	44
Power Iteration	Yes	1	85 s	20
Krylov-Schur	No	1	1646 s	16
Generalized Davidson	No	1	70 s	14
Generalized Davidson	Yes	1	46 s	8
BIFPAM	No	1	56 s	5
BIFPAM	Yes	1	25 s	3
MGBNM	Yes	1	40 s	4
Hybrid	No	1	123 s	4
Hybrid	Yes	1	39 s	2
Krylov-Schur	No	4	1955 s	19
BIFPAM	Yes	4	179 s	4
Generalized Davidson	Yes	4	220 s	42
MGBNM	Yes	4	210 s	3
Hybrid	Yes	4	185 s	2

the starting residual norm, and for some cases, it changes the convergence ratio of the methods.

As alternative, one can implement the matrices with a *full matrix-free* implementation where any matrix is allocated in memory. The BIFPAM method is used to make this comparison. However, different preconditioners must be considered instead of the ILU preconditioner for preconditioning the conjugate gradient in the diagonal blocks of the BGS preconditioner (BGS-CG-ILU). In this thesis, a multilevel with the finite element method preconditioner is proposed where the smoother is a Chebyshev polynomial. This is known as BGS-CG-MLFE preconditioner. The performance for the SP_1 equations is analyzed. Similar conclusions are obtained with the SP_3 equations. The simplified problem for the *multilevel-fem* initialization and preconditioner is obtained considering $p = 1$. The number of eigenvalues computed has been 4 with a residual error in the generalized eigenvalue problem less than 10^{-7}. The degree in the FEM has been set equal to $p = 2$ and $p = 3$ to compare the conclusions.

Table 4.12 shows that the *CSR* strategy is outperformed by the rest of the methodologies in terms of memory consumption. We can observe that the BGS-

Table 4.11: Performance results for different eigenvalue solvers for SP_3 problem with $r_r = 1$ and $p = 2$.

Solver Method	Init	n eigs.	CPU Time	its
Power Iteration	No	1	286 s	52
Power Iteration	Yes	1	166 s	26
Krylov-Schur	No	1	2729 s	16
Generalized Davidson	No	1	155 s	17
Generalized Davidson	Yes	1	105 s	12
BIFPAM	No	1	104 s	5
BIFPAM	Yes	1	65 s	3
MGBNM	Yes	1	109 s	3
Hybrid	No	1	243 s	4
Hybrid	Yes	1	86 s	2
Krylov-Schur	No	4	4749 s	28
BIFPAM	Yes	4	330 s	4
Generalized Davidson	Yes	4	408 s	49
MGBNM	Yes	4	502 s	3
Hybrid	Yes	4	393 s	2

CG-ILU preconditioner solves the SP_1 problem in the fastest way but this implementation does not allow to reduce the computational memory. If the *full matrix-free* type is considered the computational memory is greatly reduced, but the computational time is increased, for instance with $p = 2$, by a factor of 4 respect to BGS-CG-ILU. However, when the degree of the polynomial is higher, the differences between the Setup times increase and the matrix-vector multiplication performance improves. Consequently, the differences are not as higher.

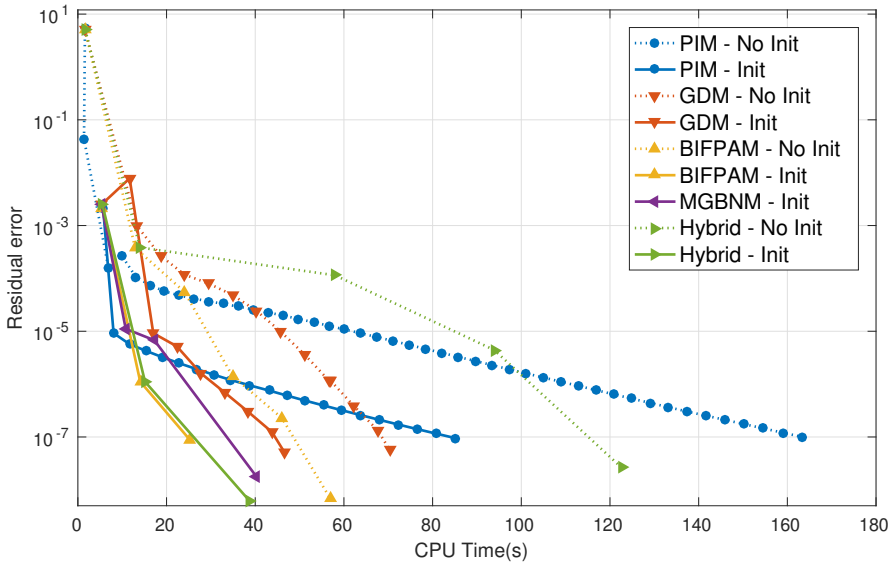


Figure 4.15: Convergence histories for the first eigenvalue for SP_1 problem with $r_r = 1$ and $p = 2$.

Table 4.12: Computational results for the 2D-C5G7 reactor with $r = 1$, $p = 2$ to solve the SP_1.

Precond.	**Allocation**	**Matrix Memory**	**Setup CPU time**	**Total CPU time**
p = 2				
BGS-CG-ILU	*Non-Diagonal*	206 Mb	2 s	184 s
BGS-CG-MLFE	*Non-Diagonal*	206 Mb	1.5 s	544 s
BGS-CG-MLFE	*Full Matrix-Free*	37 Mb	1 s	664 s
BGS-CG-MLFE	*CSR*	2557 Mb	8 s	573 s
p = 3				
BGS-CG-ILU	*Non-Diagonal*	660 Mb	7 s	696 s
BGS-CG-MLFE	*Non-Diagonal*	660 Mb	5 s	1971 s
BGS-CG-MLFE	*Full Matrix-Free*	55 Mb	2 s	1696 s
BGS-CG-MLFE	*CSR*	8522 Mb	23 s	2451 s

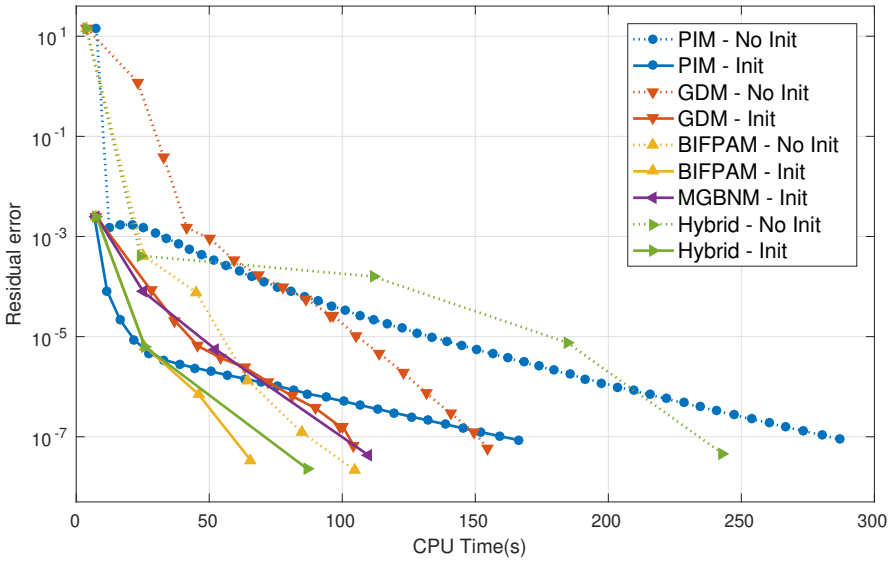


Figure 4.16: Convergence histories for the first eigenvalue for SP_3 problem with $r_r = 1$ and $p = 2$.

4.5.2 Modified 2D C5G7 reactor

To solve a more interesting problem from the computational point of view, the 2D C5G7 problem has been modified to increase the dominance ratio from $\delta \simeq 0.76$ to $\delta \simeq 0.95$. The modification has been achieved by increasing the pin size from 1.26 cm to 2.0 cm while maintaining the fuel radius.

Tables 4.13 and 4.14 show the performance of the different eigenvalue solvers for SP_1 and SP_3 problems with $r_r = 1$ and $p = 2$ for this benchmark. Again, numerical results show that the proposed BIFPAM method with initialization is more efficient than the other methods studied for one eigenvalue calculations. Compared to the previous benchmark, in this case, the CPU times for the hybrid method and MGBNM are very close to the BIFPAM results. Moreover, some more iterations are needed to solve the problem for the Power Iteration, the Krylov-Schur, the Generalized Davidson and BIFPAM methods. This tendency is stronger in the power iteration method where the computational times and the number of iterations are approximately duplicated due to the dominance factor closer to 1.0. Figures 4.17 and 4.18 show the residual norm, for the first eigenvalue in the SP_1 and SP_3 problems with $p = 2$ and $r_r = 1$.

Table 4.13: Performance results for different eigenvalue solvers for SP_1 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

Solver Method	Init	n eigs.	CPU Time	its
Power Iteration	No	1	235 s	119
Power Iteration	Yes	1	156 s	72
Krylov-Schur	No	1	1174 s	24
Generalized Davidson	No	1	57 s	22
Generalized Davidson	Yes	1	40 s	14
BIFPAM	No	1	52 s	8
BIFPAM	Yes	1	36 s	5
MGBNM	Yes	1	37 s	3
Hybrid	No	1	68 s	4
Hybrid	Yes	1	38 s	2
Krylov-Schur	No	4	1805 s	37
Generalized Davidson	Yes	4	156 s	59
BIFPAM	Yes	4	124 s	5
MGBNM	Yes	4	150 s	3
Hybrid	Yes	4	150 s	2

Table 4.14: Performance results for different eigenvalue solvers for SP_3 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

Solver Method	**Init**	**n eigs.**	**CPU Time** (s)	**its**
Power Iteration	No	1	434 s	115
Power Iteration	Yes	1	241 s	59
Krylov-Schur	No	1	2153 s	24
Generalized Davidson	No	1	126 s	25
Generalized Davidson	Yes	1	95 s	18
BIFPAM	No	1	167 s	8
BIFPAM	Yes	1	80 s	5
MGBNM	Yes	1	119 s	3
Hybrid	No	1	163 s	4
Hybrid	Yes	1	124 s	2
Krylov-Schur	No	4	3331 s	37
Generalized Davidson	Yes	4	341 s	63
BIFPAM	Yes	4	310 s	6
MGBNM	Yes	4	436 s	3
Hybrid	Yes	4	359 s	2

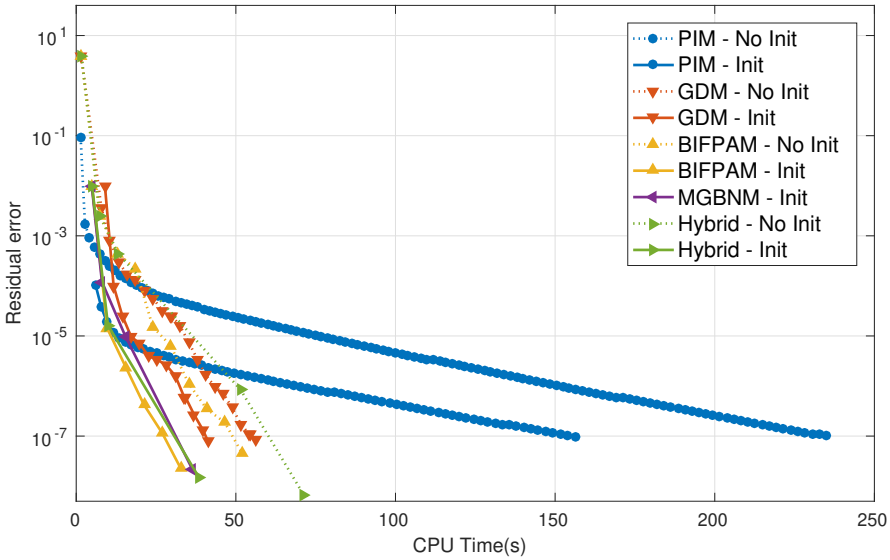


Figure 4.17: Convergence histories for the first eigenvalue for SP_1 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

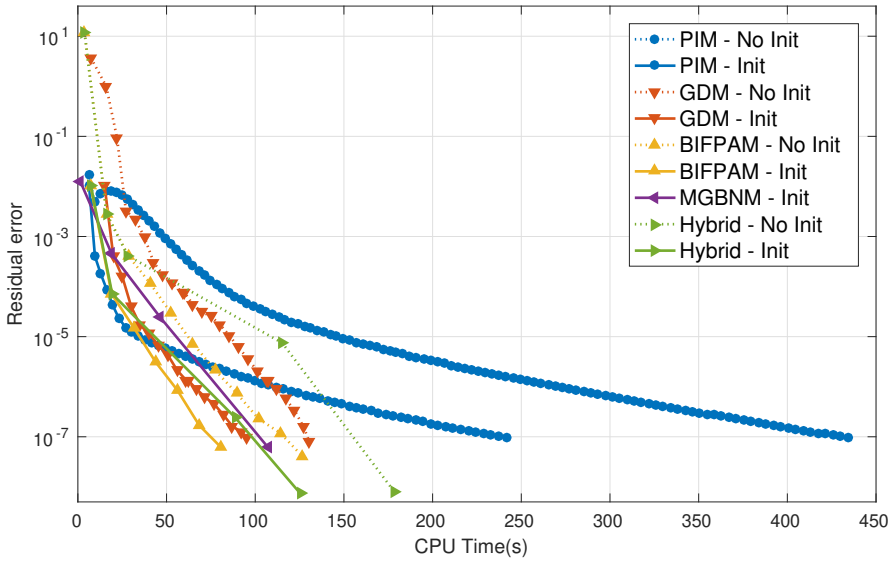


Figure 4.18: Convergence histories for the first eigenvalue for SP_3 problem with $r_r = 1$ and $p = 2$ for the modified 2D-C5G7.

4.5.3 3D-C5G7 reactor

The tridimensional case of the 3D-C5G7 reactor (Lewis et al., 2001) is chosen to test the block methods in a computationally challenging problem. In this case, due to the high computational memory requirements, that the resolution of this problem implies, the performance has been tested in the Rigel Cluster from UPV (*Rigel Cluster Description*). It is composed of 72 nodes where each node includes two Intel Xeon E5-2450 processors and 64 Gb DDR3 RAM. Nodes are linked by two 10 GB Ethernet interfaces. The cluster runs a CentOS 6 operating system.

Tables 4.15 and 4.16 show the performance of the different eigenvalue solvers for SP_1 and SP_3 problems with $r_a = 1$, $r_r = 1$ and $p = 2$ for this benchmark. Thus, the mesh used has 264 992 finite element cells and 16 407 055 degrees of freedom for SP_1 equations and 32 814 110 degrees of freedom for SP_3 equations. First of all, it must be noted that the computational time has been increased from a few minutes for the two-dimensional version to several hours for the three-dimensional case. Again, numerical results show that the proposed BIFPAM method with initialization is more efficient than the other methods studied for one eigenvalue calculations. Figures 4.19 and 4.20 show the historic residual norm, for the first eigenvalue in the SP_1 and SP_3 problems.

Table 4.15: Performance results for different eigenvalue solvers for SP_1 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

Solver Method	Init	n eigs.	CPU Time	its
Power Iteration	No	1	500 min	52
Power Iteration	Yes	1	288 min	27
Generalized Davidson	No	1	203 min	17
Generalized Davidson	Yes	1	143 min	12
BIFPAM	No	1	139 min	5
BIFPAM	Yes	1	90 min	3
MGBNM	Yes	1	119 min	4
Hybrid	No	1	185 min	4
Hybrid	Yes	1	112 min	3

Now, as in the two-dimensional case, the *Full Matrix-free* implementation is tested for two cases for the solution of the SP_1 equations. Similar conclusions are obtained for the SP_3 equations. First, the mesh with $r_r = 0$ and $r_a = 0$ is used for $p = 2$, that gives a number of degrees of freedom equal to 2 998 863. Then, the degree of the polynomial p is increased to $p = 3$, to obtain 9 732 268 degrees of freedom. In this case, the *CSR* strategy has not been computed due to its

Table 4.16: Performance results for different eigenvalue solvers for SP_3 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

Solver Method	Init	n eigs.	CPU Time	its
Power Iteration	No	1	764 min	60
Power Iteration	Yes	1	513 min	34
Generalized Davidson	No	1	602 min	24
Generalized Davidson	Yes	1	486 min	13
BIFPAM	No	1	226 min	5
BIFPAM	Yes	1	184 min	4
MGBNM	Yes	1	257 min	4
Hybrid	No	1	311 min	4
Hybrid	Yes	1	219 min	3

high memory demands. The same pattern of results described for the 2D-C5G7 can be observed in this Table for $p = 2$. However, for the 3D case, the reduction of the memory is considerable and the differences in the CPU times are reduced significantly. One reason is that, the time to allocate the matrices and construct the preconditioner increases. Actually, for $p = 3$, these differences make that the BGS-CG-MLFE preconditioner with the *full matrix-free* allocation gives lower CPU times than the BCG-CG-ILU preconditioner.

Table 4.17: Computational results for the 3D-C5G7 with $r_a = 0$, $r_r = 0$ for the SP_1 problem.

Precond.	Allocation	Mat. Memory	Setup CPU time	Total CPU time
p = 2				
BGS-CG-ILU	*Non-Diagonal*	2474 Mb	17 s	832 s
BGS-CG-MLFE	*Non-Diagonal*	2474 Mb	15 s	852 s
BGS-CG-MLFE	*Full Matrix-Free*	481 Mb	3 s	1014 s
p = 3				
CG-ILU	*Non-Diagonal*	15041 Mb	104 s	3657 s
BGS-CG-MLFE	*Non-Diagonal*	15041 Mb	95 s	7236 s
BGS-CG-MLFE	*Full Matrix-Free*	950 Mb	13 s	3111 s

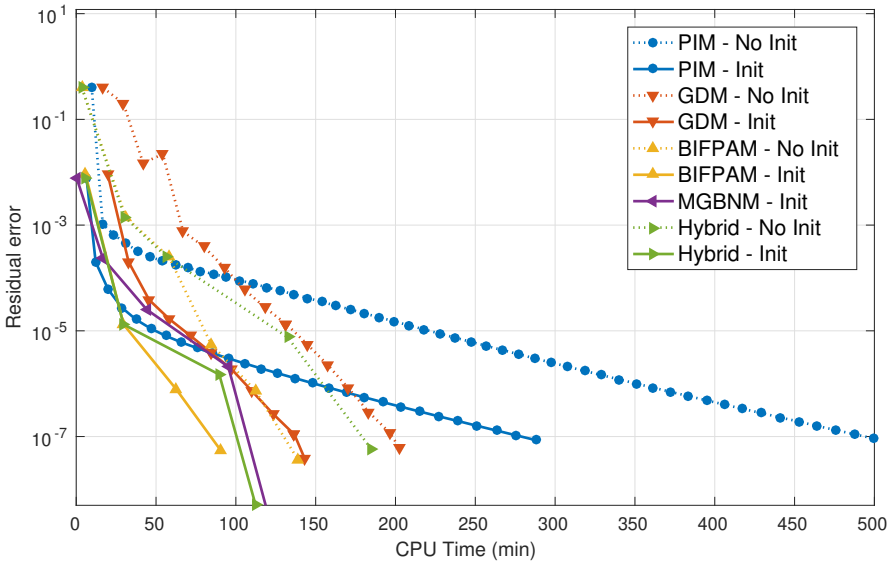


Figure 4.19: Convergence histories for the first eigenvalue for SP_1 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

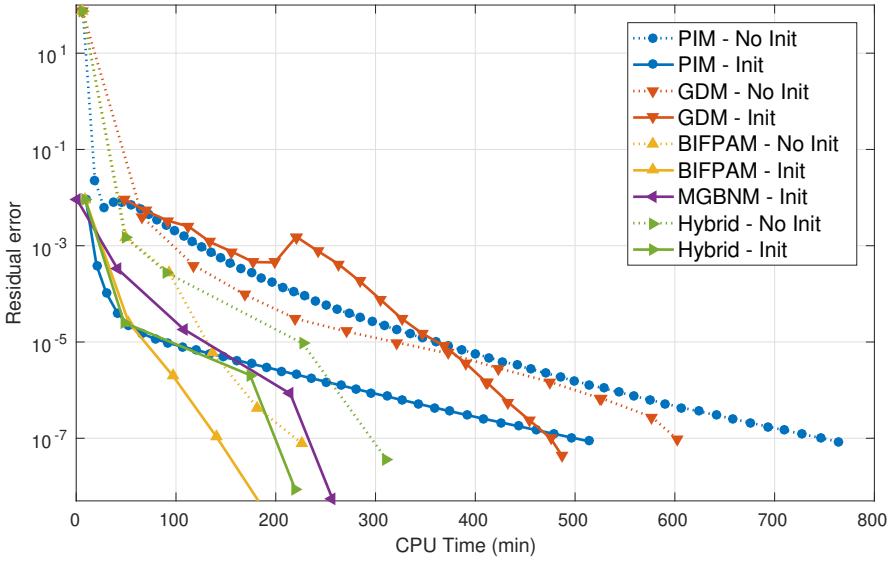


Figure 4.20: Convergence histories for the first eigenvalue for SP_3 problem with $r_a = 1$, $r_r = 1$ and $p = 2$ for the 3D-C5G7.

CHAPTER 5

MODAL METHODS FOR THE TIME DEPENDENT NEUTRON DIFFUSION EQUATION

The solution of time-dependent multigroup neutron diffusion equations approximates the neutron distribution that describes reactor kinetics. This equation depends on the position and time. The finite element discretization leads to a semidiscrete time-dependent ordinary system of differential equations (ODE). Usually, this problem is a so-called stiff problem in time due, among other things, to the presence of both prompt and delayed neutrons that lead to time scales of different orders of magnitude. In consequence, obtaining, efficiently, the solution of this system will depend heavily on the methodology used.

Several approaches have been studied to integrate this time-dependent equation. One group of these methods is based on finite differentiation schemes for the time variable such as the backward differential method or the implicit Euler method (Stacey, 1969). Other methods are based on the factorization of space and time dependence of the neutronic flux (factorization methods). These methodologies express the solution as a product of two functions: one time-dependent function (amplitude factor) and a second one that describes the spatial distribution (shape function) and has a slow variation with time. In the point kinetics reactor approximation, the dominant eigenfunction associated with an auxiliar eigenvalue problem, corresponding to a static configuration of the core, is taken as the shape

function (Akcasuh, 1971). For the quasistatic approximation, this shape function is updated over time (Henry, 1958; Dulla et al., 2008; Yamamoto et al., 2010). However, this expansion has limitations when the spatial distribution of the neutron flux changes along the transient and cannot be described using only one shape function. This occurs, for instance, in the out-of-phase oscillations observed in Boiling Water Reactors (BWR) (March-Leuba and Blakeman, 1991; March-Leuba and Rey, 1993). The generalization of this approach is the modal method (Stacey, 1969; Miró et al., 2002; Lima et al., 2009; Avvakumov et al., 2017a). It expands the time-dependent flux as a sum of several spatial eigenfunctions associated with the initial configuration of the reactor. These spatial modes can be also updated along the transient to reduce the number of shape functions used in the expansion (Miró et al., 2002). However, most of the time-dependent core simulators do not rely on such approximations that come from factorization methods (in spite of being, on some occasions, much more efficient in terms of the CPU time and the computational memory).

Different spectral problems can be associated with the neutron diffusion equation for different purposes, as it has been shown in the previous Chapters. Consequently, different modal strategies to integrate the time-dependent neutron equations can be defined. The nodal modal kinetics associated with the λ-modes was studied in detail in (Verdú et al., 1998; Miró et al., 2002) to solve the time-dependent neutron diffusion equation and to classify instabilities in BWRs. This methodology has been also applied by using a finite volume method for the spatial discretization (Bernal García, 2018). Lange et al. also used the subcritical λ-modes to study the BWR stability states (Lange et al., 2014). The α-modes have been also applied to study reactor instabilities (Verdú and Ginestar, 2014). More recently, the State Change Modal (SCM) method has been proposed based also on the calculation of the dominant α-modes (Avvakumov et al., 2018b). In (Dulla et al., 2018), the authors have developed a fully analytical study of the spectrum of the neutron diffusion operator to analyze some general properties of the neutron evolution. In this thesis, different modal schemes with the λ, γ and α-modes are developed to their comparison.

Classically, the implementations of the integration methods for the time-dependent neutron diffusion equation use small fix time-steps over which there are little changes in the neutron population, to ensure the stability of the solution. For differential methods, high order schemes can be proposed to avoid to use small time-steps (Ginestar et al., 1998). However, obtaining the approximations at each time-step is also computationally very expensive. Other approach used very often is the implementation of adaptive time-step controls where the time-step is computed step by step. Ideally, the control is designed to pick an optimal time

step size adaptively based on the current state of a transient. In practice, this step is selected from the solution in the previous steps to satisfy some approximated error tolerance (Gustafsson et al., 1988; Wanner and Hairer, 1996; Söderlind, 2002). These schemes provide improved computational efficiency because time is not wasted solving the system to a level of accuracy beyond relevance, but also to reduce the step size if they detect a numerical instability. Moreover, adaptive schemes remove the necessity for the user to select time steps before the simulation has been run. There are common approaches to do this, however there are no generalized methods that are routinely applied to reactor physics computations.

Recent works have incorporated an adaptive time-step for the Backward differential method for different orders (Shim et al., 2011; Avvakumov et al., 2018a; Boffie and Pounders, 2018; Cai et al., 2019). The quasi-static methodology is also implemented by selecting an appropriate step size from a given tolerance (Caron et al., 2017). Step size controllers require an error estimation. The time selection for the previous works is based on some approximations of the local truncation error causes by the time discretization since both come from a finite difference discretization. In the modal approach, different error estimations must be defined. In this thesis, several error approximations are studied to implement an adaptive time-step control to update the calculation of the modes.

This Chapter presents the modal methodology to integrate the time-dependent neutron diffusion equation. Section 5.1 exposes the algebraic time-dependent equation obtained from the finite element discretization. Section 5.2 briefly describes the backward differential method to compare with the modal methods. Section 5.3 exposes the modal methodology for the λ, the γ and the α-modes. Section 5.4 presents some numerical results for different transients with different type of perturbations. The modal method can be updated along time to avoid to use a high number of modes in the modal expansion. This strategy to update the modal method is depicted in Section 5.5. Section 5.6 tests the updated modal method in some transients. Section 5.7 describes the adaptive time-step control designed for the updated modal methods. Finally, numerical results for several transient benchmark are shown to study the performance of the adaptive updated modal method (Section 5.8). This Chapter reviews and rewrites some results presented in (Carreño et al., 2019c) together with work that has not yet been published.

5.1 Finite element discretization

For a given transient, the flux of neutrons inside of a nuclear reactor core can be described through the time-dependent neutron diffusion equation in the approximation of two energy groups, without up-scattering and K groups of delayed neutron precursors (Stacey, 1969). These equations can be expressed as

$$\begin{aligned} \mathcal{V}^{-1}\frac{\partial \Phi}{\partial t} + (\mathcal{L}+\mathcal{S})\Phi &= (1-\beta)\mathcal{F}\Phi + \sum_{k=1}^{K}\lambda_k^d \mathcal{C}_k\chi, \\ \frac{\mathrm{d}\mathcal{C}_k}{\mathrm{d}t} &= \beta_k\mathcal{F}_1\Phi - \lambda_k^d\mathcal{C}_k, \qquad k=1,\ldots,K, \end{aligned} \tag{5.1}$$

where,

$$\mathcal{L} = \begin{pmatrix} -\vec{\nabla}\cdot(D_1\vec{\nabla}) + \Sigma_{a_1} + \Sigma_{12} & 0 \\ 0 & -\vec{\nabla}\cdot(D_2\vec{\nabla}) + \Sigma_{a_2} \end{pmatrix},$$

$$\mathcal{S} = \begin{pmatrix} 0 & 0 \\ -\Sigma_{12} & 0 \end{pmatrix}, \qquad \mathcal{F} = \begin{pmatrix} \nu\Sigma_{f_1} & \nu\Sigma_{f_2} \\ 0 & 0 \end{pmatrix}, \qquad \mathcal{F}_1 = \begin{pmatrix} \nu\Sigma_{f_1} & \nu\Sigma_{f_2} \end{pmatrix},$$

$$\mathcal{V}^{-1} = \begin{pmatrix} v_1^{-1} & 0 \\ 0 & v_2^{-1} \end{pmatrix}, \qquad \chi = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \Phi = \begin{pmatrix} \Phi_1 \\ \Phi_2 \end{pmatrix}.$$

The operator $\mathcal{L}$ is known as leakage operator, the operator $\mathcal{S}$ is the scattering operator and $\mathcal{F}$ is the fission term. The vectors Φ_1 and Φ_2 represent the fast and thermal fluxes of neutrons, respectively. The term $\mathcal{C}_k$ is the concentration of delayed neutron precursors of the group k. The rest of the coefficients (cross-sections) are also, in general, position and time-dependent functions.

The system (5.1) needs to be discretized to obtain an approximated solution. Applying the finite element method presented for the steady-state neutron diffusion equation in Chapter 3 yields to the semi-discrete time-dependent system

$$\begin{aligned} [V]^{-1}\frac{\mathrm{d}\tilde{\Phi}}{\mathrm{d}t} + (L+S)\tilde{\Phi} &= (1-\beta)F\tilde{\Phi} + \sum_{k=1}^{K}\lambda_k^d X C_k, \\ \frac{\mathrm{d}C_k}{\mathrm{d}t} &= \beta_k F_1\tilde{\Phi} - \lambda_k^d C_k, \qquad k=1,\ldots,K, \end{aligned} \tag{5.2}$$

where L, S, F, F_1, $[V]$ are the matrices obtained from the discretization of operators $\mathcal{L}$, $\mathcal{S}$, $\mathcal{F}$, $\mathcal{F}_1$, $\mathcal{V}$ respectively. Vectors $\tilde{\Phi}$ and C_k are the corresponding

coefficients of Φ and $\mathcal{C}_k$ in terms of the Lagrange polynomials used in the finite element method. Henceforth, as an abuse of notation, the algebraic flux is denoted as Φ by removing the tilde from the original notation. The matrix X corresponds to

$$X = \begin{pmatrix} I \\ 0 \end{pmatrix}, \tag{5.3}$$

where I is the identity matrix.

5.2 Backward differential method

The system of ordinary differential equations (5.2) is, in general, stiff. Then, it is convenient to use an implicit method for the time discretization. In particular, the backward differential method (BKM), that is a classical implicit method, can be used (Ginestar et al., 1998; Vidal-Ferràndiz et al., 2016).

This method starts with the steady-state flux, i.e., from the solution of one spatial modes problem, $\Phi(0) = \phi_1^\lambda$, where the system is in critical state by dividing previously the fission terms, F, by $k_{eff} = \lambda_1$. An approximated solution, Φ^{n+1}, in the $(n+1)$-th step, with a time-step of $h_n = t_{n+1} - t_n$, is obtained by solving the following system of linear equations

$$T^{n+1}\Phi^{n+1} = R^n\Phi^n + \sum_{k=1}^{K} \lambda_k^d e^{-\lambda_k^d h_n} X C_k^n \ , \tag{5.4}$$

where the matrices are defined as

$$T^{n+1} = R^n + L^{n+1} + S^{n+1} - \hat{a}F^{n+1}, \qquad R^n = \frac{1}{h_n}[V]^{-1} \ ,$$

and the coefficient $\hat{a}$ is computed as

$$\hat{a} = 1 - \beta + \sum_{k=1}^{K} \beta_k \left(1 - e^{-\lambda_k^d h_n}\right) \ .$$

The matrices L^{n+1}, S^{n+1} and F^{n+1} correspond to the discretized operators at time t_{n+1}.

The neutron precursors equation is discretized in time by using an explicit scheme as

$$C_k^{n+1} = C_k^n e^{-\lambda_k^d h_n} + \frac{\beta_k}{\lambda_k^d}\left(1 - e^{-\lambda_k^d h_n}\right)\left(F_{11}^{n+1} F_{12}^{n+1}\right)\ \Phi^{n+1} \ . \tag{5.5}$$

This system of equations is large and sparse and has to be solved for each new time step, t_{n+1}. In this thesis, the GMRES method has been used to solve the linear systems preconditioned with the block Gauss-Seidel method (Section 4.3). The inverse of each diagonal block is approximated by a solving linear system with the conjugate gradient method with the ILU preconditioner and $tol < 1e-6$. This strategy permits to treat the matrices in the code by using the *non-diagonal* matrix-free implementation (Section 4.4) and thus, to avoid the assembly of the matrices at each time-step.

5.3 Modal method

Alternative to the backward differential method, one can use the modal method to approximate the time-dependent neutron diffusion equation (5.2). This strategy assumes that $\Phi(\vec{r},t)$ admits the following expansion

$$\Phi(\vec{r},t) = \sum_{m=1}^{\infty} n_m^{\delta}(t)\phi_m^{\delta}(\vec{r}), \tag{5.6}$$

where $\phi_m^{\delta}(\vec{r})$ is the unitary eigenvector associated with the m-th dominant eigenvalue of some static problem

$$(L+S)\phi_m^{\lambda} = \frac{1}{\lambda_m} F\phi_m^{\lambda}, \tag{5.7}$$

$$L\phi_m^{\gamma} = \frac{1}{\gamma_m}(F-S)\phi_m^{\gamma}, \tag{5.8}$$

$$(F-S-L)\phi_m^{\alpha} = \alpha_m [V]^{-1}\phi_m^{\alpha}, \tag{5.9}$$

where the matrices are given by the finite element method discretization (Equation (3.14)). The amplitude coefficients $n_m^{\delta}(t)$ are only time dependent and they will change with the kind of spatial mode used ($\delta = \lambda, \gamma, \alpha$). To simplify the notation, we will write n_m^{δ} and ϕ_m^{δ} instead of $n_m^{\delta}(t)$ and $\phi_m^{\delta}(\vec{r})$.

For each kind of modes problem, we choose the matrices L, S and F, and we denote by L_0, S_0 and F_0, as the matrices related to the configuration of the reactor core at $t = 0$. We start with this reactor in critical state by dividing the matrix related to the fission terms, F_0, by $k_{eff} = \lambda_1$. However, note that if we force the transient to start the computation with the reactor in critical state, the dominant α-mode will be equal to zero and the numerical methods have some convergence problems. Thus, in the case of the α-modes, we divide the fission cross-section by $k_{eff} + 10^{-8}$ obtaining a reactor quasi-critical. In this way, we

express the matrices L, S and F as

$$L(t) = L_0 + \delta L(t), \quad S(t) = S_0 + \delta S(t), \quad F(t) = F_0 + \delta F(t). \tag{5.10}$$

If the expansion (5.6) is truncated until a certain number of modes, q, and substituted into the system (5.2), it is obtained

$$\begin{aligned}
&[V]^{-1} \sum_{m=1}^{q} \frac{\mathrm{d}n_m^\delta}{\mathrm{d}t}\phi_m^\delta + \sum_{m=1}^{q} L_0 n_m^\delta \phi_m^\delta + \sum_{m=1}^{q} \delta L n_m^\delta \phi_m^\delta + \sum_{m=1}^{q} S_0 n_m^\delta \phi_m^\delta + \sum_{m=1}^{q} \delta S n_m^\delta \phi_m^\delta \\
&= (1-\beta) \sum_{m=1}^{q} F_0 n_m^\delta \phi_m^\delta + (1-\beta) \sum_{m=1}^{q} \delta F n_m^\delta \phi_m^\delta + \sum_{k=1}^{K} \lambda_k^d X C_k, \\
&\frac{\mathrm{d}XC_k}{\mathrm{d}t} = \sum_{m=1}^{q} \beta_k F_0 n_m^\delta \phi_m^\delta + \sum_{m=1}^{q} \beta_k \delta F n_m^\delta \phi_m^\delta - \lambda_k^d X C_k, \qquad k = 1, \ldots, K.
\end{aligned} \tag{5.11}$$

Hereafter, we particularize the equations obtained for each kind of spatial mode. Using the dominant λ-modes (Equation (3.12)) into the Equation (5.11), we obtain

$$\begin{aligned}
&[V]^{-1} \sum_{m=1}^{q} \frac{\mathrm{d}n_m^\lambda}{\mathrm{d}t}\phi_m^\lambda + \sum_{m=1}^{q} \frac{1}{\lambda_m} F_0 n_m^\lambda \phi_m^\lambda + \sum_{m=1}^{q} \delta L n_m^\lambda \phi_m^\lambda + \sum_{m=1}^{q} \delta S n_m^\delta \phi_m^\delta \\
&= (1-\beta) \sum_{m=1}^{q} F_0 n_m^\lambda \phi_m^\lambda + (1-\beta) \sum_{m=1}^{q} \delta F n_m^\lambda \phi_m^\lambda + \sum_{k=1}^{K} \lambda_k^d X C_k, \\
&\frac{\mathrm{d}XC_k}{\mathrm{d}t} = \sum_{m=1}^{q} \beta_k F_0 n_m^\lambda \phi_m^\lambda + \sum_{m=1}^{q} \beta_k \delta F n_m^\lambda \phi_m^\lambda - \lambda_k^d X C_k, \qquad k = 1, \ldots, K.
\end{aligned} \tag{5.12}$$

Then, Equations (5.12) are multiplied by the adjoint modes $\phi_l^{\lambda,\dagger}$ with $l = 1, \ldots, q$ and the biorthogonality condition (2.49) is used to obtain the following system

of $q(K+1)$ equations

$$\begin{aligned}
&\sum_{m=1}^{q}\langle\phi_l^{\lambda,\dagger},[V]^{-1}\phi_m^{\lambda}\rangle\frac{\mathrm{d}}{\mathrm{d}t}n_m^{\lambda}+\frac{1}{\lambda_l}n_l^{\lambda}+\sum_{m=1}^{q}\langle\phi_l^{\lambda,\dagger},\delta L\phi_m^{\lambda}\rangle n_m^{\lambda}+\sum_{m=1}^{q}\langle\phi_l^{\lambda,\dagger},\delta S\phi_m^{\lambda}\rangle n_m^{\lambda}\\
&=(1-\beta)n_l+(1-\beta)\sum_{m=1}^{q}\langle\phi_l^{\lambda,\dagger},\delta F\phi_m^{\lambda}\rangle n_m^{\lambda}+\sum_{k=1}^{K}\lambda_k^d\langle\phi_l^{\lambda,\dagger},XC_k\rangle, \qquad l=1,\dots,q,\\
&\frac{\mathrm{d}}{\mathrm{d}t}\langle\phi_l^{\lambda,\dagger},XC_k\rangle=\beta_k n_l^{\lambda}+\beta_k\sum_{m=1}^{q}\langle\phi_l^{\lambda,\dagger},\delta F\phi_m^{\lambda}\rangle n_m^{\lambda}-\lambda_k^d\langle\phi_l^{\lambda,\dagger},XC_k\rangle, \quad k=1,\dots,K.
\end{aligned} \tag{5.13}$$

Introducing the notation

$$\begin{aligned}
&\Lambda_{lm}^{\lambda}=\langle\phi_l^{\lambda,\dagger},[V]^{-1}\phi_m^{\lambda}\rangle, \quad \Delta L_{lm}^{\lambda}=\langle\phi_l^{\lambda,\dagger},\delta L\phi_m^{\lambda}\rangle,\\
&\Delta S_{lm}^{\lambda}=\langle\phi_l^{\lambda,\dagger},\delta S\phi_m^{\lambda}\rangle, \quad \Delta F_{lm}^{\lambda}=\langle\phi_l^{\lambda,\dagger},\delta F\phi_m^{\lambda}\rangle,\\
&c_{lk}^{\lambda}=\langle\phi_l^{\lambda,\dagger},XC_k\rangle,
\end{aligned} \tag{5.14}$$

into the system of equations (5.13) and reordering terms yields to the system

$$\begin{aligned}
\sum_{m=1}^{q}\Lambda_{lm}^{\lambda}\frac{\mathrm{d}}{\mathrm{d}t}n_m^{\lambda}&=(1-\frac{1}{\lambda_l}-\beta)n_l^{\lambda}-\sum_{m=1}^{q}\Delta L_{lm}^{\lambda}n_m^{\lambda}-\sum_{m=1}^{q}\Delta S_{lm}^{\lambda}n_m^{\lambda}\\
&\quad+(1-\beta)\sum_{m=1}^{q}\Delta F_{lm}^{\lambda}n_m^{\lambda}+\sum_{k=1}^{K}\lambda_k^d c_{lk}^{\lambda}, \qquad l=1,\dots,q.
\end{aligned}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}c_{lk}^{\lambda}=\beta_k n_l^{\lambda}+\beta_k\sum_{m=1}^{q}\Delta F_{lm}^{\lambda}n_m^{\lambda}-\lambda_k^d c_{lk}^{\lambda}, \qquad k=1,\dots,K, \tag{5.15}$$

In matrix form, the system (5.15) can be also expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}^{\lambda}=\mathbf{T}^{\lambda}\mathbf{N}^{\lambda}, \tag{5.16}$$

where

$$\mathbf{N}^{\lambda}=\begin{pmatrix}n_1^{\lambda}\cdots n_q^{\lambda} & c_{11}^{\lambda}\cdots c_{q1}^{\lambda} & \cdots & c_{1K}^{\lambda}\cdots c_{qK}^{\lambda}\end{pmatrix}^{\mathsf{T}}, \tag{5.17}$$

and

$$\mathbf{T}^{\lambda} = \left(\begin{array}{c|ccc} \Lambda_{\lambda}^{-1}\left((1-\beta)(I+\Delta F^{\lambda}) - [\lambda]^{-1} - \Delta L^{\lambda} - \Delta S^{\lambda}\right) & \Lambda_{\lambda}^{-1}\lambda_1^d & \cdots & \Lambda_{\lambda}^{-1}\lambda_K^d \\ \hline \beta_1(I+\Delta F^{\lambda}) & -\lambda_1^d I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_K(I+\Delta F^{\lambda}) & 0 & \cdots & -\lambda_K^d I \end{array}\right). \tag{5.18}$$

The block $[\lambda]$ denotes the diagonal matrix whose elements are the dominant λ eigenvalues and I is the identity matrix.

Through an analogous process for the γ and α-modes, we obtain similar differential systems. For the γ-modes, using their corresponding biorthogonality condition yields to the following matrix system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}^{\gamma} = \mathbf{T}^{\gamma}\mathbf{N}^{\gamma}, \tag{5.19}$$

where

$$\mathbf{N}^{\gamma} = \begin{pmatrix} n_1^{\gamma} \cdots n_q^{\gamma} & c_{11}^{\gamma} \cdots c_{q1}^{\gamma} & \cdots & c_{1K} \cdots c_{qK}^{\gamma} \end{pmatrix}^{\mathsf{T}}, \tag{5.20}$$

$$\mathbf{T}^{\gamma} = \left(\begin{array}{c|ccc} \Lambda_{\gamma}^{-1}\left(I - [\gamma]^{-1} - \Delta L^{\gamma} - \Delta S^{\gamma} - \beta A^{F,\gamma} + (1-\beta)\Delta F^{\gamma}\right) & \Lambda_{\gamma}^{-1}\lambda_1^d & \cdots & \Lambda_{\gamma}^{-1}\lambda_K^d \\ \hline \beta_1(A^{F,\gamma} + \Delta F^{\gamma}) & -\lambda_1^d I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_K(A^{F,\gamma} + \Delta F^{\gamma}) & 0 & \cdots & -\lambda_K^d I \end{array}\right), \tag{5.21}$$

and

$$\begin{aligned} A_{lm}^{F,\gamma} &= \langle \phi_l^{\gamma,\dagger}, F_0 \phi_m^{\gamma} \rangle, & \Delta L_{lm}^{\gamma} &= \langle \phi_l^{\gamma,\dagger}, \delta L \phi_m^{\gamma} \rangle, \\ \Delta S_{lm}^{\gamma} &= \langle \phi_l^{\gamma,\dagger}, \delta S \phi_m^{\gamma} \rangle, & \Delta F_{lm}^{\gamma} &= \langle \phi_l^{\gamma,\dagger}, \delta F \phi_m^{\gamma} \rangle, \\ c_{lk}^{\gamma} &= \langle \phi_l^{\gamma,\dagger}, X C_k \rangle, & \Lambda_{lm}^{\gamma} &= \langle \phi_l^{\gamma,\dagger}, [V]^{-1} \phi_m^{\gamma} \rangle, \end{aligned} \tag{5.22}$$

The block $[\gamma]$ denotes the diagonal matrix whose elements are the dominant values of γ.

If now, we use the α-modes problem and its associated adjoint problem, we have the following system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}^{\alpha} = \mathbf{T}^{\alpha}\mathbf{N}^{\alpha}, \tag{5.23}$$

where

$$\mathbf{N}^{\alpha} = \begin{pmatrix} n_1^{\alpha} \cdots n_q^{\alpha} & c_{11}^{\alpha} \cdots c_{q1}^{\alpha} & \cdots & c_{1K}^{\alpha} \cdots c_{qK}^{\alpha} \end{pmatrix}^{\mathsf{T}}, \tag{5.24}$$

$$\mathbf{T}^{\alpha} = \left(\begin{array}{c|ccc} [\alpha] - \beta A^{F,\alpha} - \Delta L^{\alpha} - \Delta S^{\alpha} + (1-\beta)\Delta F^{\alpha} & \lambda_1^d & \cdots & \lambda_K^d \\ \hline \beta_1(A^{F,\alpha} + \Delta F^{\alpha}) & -\lambda_1^d I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_K(A^{F,\alpha} + \Delta F^{\alpha}) & 0 & \cdots & -\lambda_K^d I \end{array}\right) \tag{5.25}$$

and

$$\begin{aligned} A_{lm}^{F,\alpha} &= \langle \phi_l^{\alpha,\dagger}, F_0 \phi_m^{\alpha} \rangle, \quad & \Delta L_{lm}^{\alpha} &= \langle \phi_l^{\alpha,\dagger}, \delta L \phi_m^{\alpha} \rangle, \\ \Delta S_{lm}^{\alpha} &= \langle \phi_l^{\alpha,\dagger}, \delta S \phi_m^{\alpha} \rangle, \quad & \Delta F_{lm}^{\alpha} &= \langle \phi_l^{\alpha,\dagger}, \delta F \phi_m^{\alpha} \rangle, \\ c_{lk}^{\alpha} &= \langle \phi_l^{\alpha,\dagger}, X C_k \rangle. \end{aligned} \tag{5.26}$$

The block $[\alpha]$ denotes the diagonal matrix whose elements are the dominant values of α.

All systems of differential equations need initial conditions to be solved. From the equations in the steady state, these conditions are

$$\begin{aligned} & n_1^{\delta}(0) = 1, && n_m^{\lambda}(0) = 0,\ \forall m = 2, \ldots, q \\ & c_{1k}^{\delta}(0) = \frac{\beta_k}{\lambda_k^d} \langle \phi_1^{\delta,\dagger}, F_0 \phi_1^{\delta} \rangle, && c_{mk}^{\delta}(0) = 0,\ \forall m = 2, \ldots, q, \quad \forall k = 1, \ldots, K, \end{aligned}$$

with ϕ_1^{δ} and $\phi_1^{\delta,\dagger}$ the corresponding direct and adjoint eigenvector of the dominant eigenvalue $\delta = \lambda, \gamma, \alpha$.

The system of differential equations obtained for the different spatial modes is quite smaller than the original system (5.2) when the number of eigenvalues used in the expansion, q, is not too large. These systems are also stiff, so implicit methods are needed to obtain approximate solutions. In this work, we use a

backward differentiation formula implemented in the CVODE solver from the SUNDIALS library (Hindmarsh et al., 2005; Abhyankar et al., 2018). This code has implemented an adaptive time step and it is initialized with time step of $\delta t = 10^{-5}$ s. The absolute tolerance in the CVODE has been set equal to 10^{-10}. Moreover, the modal method has been implemented with a *full matrix-free* technique (Section 4.4) for the matrices L, S, F and $[V]$ to avoid their assembly at each time-step.

5.4 Numerical results for the modal method

In this Section, we present numerical results obtained for several three-dimensional transients to compare the performance of the different modal methods presented above. The first benchmark is a theoretical transient that has been used to validate the code. The second one is a more realistic benchmark, the Langenbuch reactor (Langenbuch et al., 1977), that has been perturbed with an out-of-phase local oscillation in the material cross sections. The third transient is a control rod movement in the Langenbuch reactor. For each transient, the results obtained with the modal methods have been compared with the ones obtained with a code that integrates Equation (5.2) by using the backward difference method (BKM).

All results are computed using a degree of the polynomial equal to 3 in the the finite element method, since it has been shown in the Chapter 3 that this degree is enough to obtain accurate results for usual reactor calculations.

The solution of the eigenvalue problems (direct and adjoint) has been computed with the hybrid method (Section 4.1.7) with a tolerance in the generalized eigenvalue problem of $\text{res}_g < 10^{-9}$. For the direct modes computation, the *multilevel-fem* initialization with degree equal to 1 and the solution λ-modes problem is used as initial guess for the hybrid method. For the adjoint modes computation, the solution of the direct modes is used to initialize the solver.

For reactors without spatial symmetry, the eigenvalues solution of problems are not degenerated and the adjoint eigenvectors computed are biorthogonal, thus we only need to divide each adjoint vector $\phi_l^{\lambda,\dagger}$, $\phi_l^{\gamma,\dagger}$, $\phi_l^{\alpha,\dagger}$ by the product $\langle \phi_l^{\lambda,\dagger}, F\phi_l^{\lambda} \rangle$, $\langle \phi_l^{\gamma,\dagger}, (F-S)\phi_l^{\gamma} \rangle$ and $\langle \phi_l^{\alpha,\dagger}, [V]^{-1}\phi_l^{\alpha} \rangle$, respectively, to obtain a biorthonormal basis. For reactors with radial symmetry, it can be proved (see (Tommasi et al., 2016)) that degenerated eigenvalues (i.e. eigenvalues with multiplicity greater than 1) can appear, and consequently the adjoint modes computed are not directly biorthogonal. This problem is solved by using the biorthogonalization procedure shown in Algorithm 5 (Adrover et al., 2005).

In this section, we have computed the relative errors related to the neutron power that we describe hereafter. Remember that the neutron power (P^δ) associated with the δ-modes is defined as

$$\mathrm{P}^\delta(\vec{r},t) = \Sigma_{f1}\phi_1^\delta(\vec{r},t) + \Sigma_{f2}\phi_2^\delta(\vec{r},t),$$

where $\delta = \lambda, \gamma, \alpha$ is the eigenvalue associated with the eigenvector ϕ^δ.

The Local Error (LE) at time t is given by

$$\mathrm{LE}^\delta(t) = \frac{\|\mathrm{P}^\delta(t) - \mathrm{P}^{\delta,\mathrm{ref}}(t)\|_1}{\|\mathrm{P}^{\delta,\mathrm{ref}}(t)\|_1},$$

where $\mathrm{P}^{\delta,\mathrm{ref}}(t)$ is the reference power at time t.

The Mean Power Error (MPE) in the interval $[t_0, t_N]$ is defined by

$$\mathrm{MPE}^\delta = \frac{1}{(t_N - t_0)} \sum_{n=1}^{N} \mathrm{LE}^\delta(t_n)(t_n - t_{n-1}).$$

The Radial Power Error (RPE) at the middle plane is defined by

$$\mathrm{RPE}^\delta(t) = |\mathrm{P}_z^\delta(t) - \mathrm{P}_z^{\delta,\mathrm{ref}}(t)|,$$

where $\mathrm{P}_z^\delta(t)$ denotes the power in the middle plane at time t.

The modal methodology and the Backward differential method have been implemented in C++ based on the data structures provided by the library Deal.II (Bangerth et al., 2007) and PETSc (Abhyankar et al., 2018) . The computer used for the computations was an Intel® Core™i7-4790 3.60 GHz with 32 Gb of RAM running on Ubuntu GNU/Linux 16.04 LTS.

5.4.1 Cuboid reactor

This transient is based on a non homogeneous prismatic reactor. It is described in Appendix B.2. The transient analyzed has been defined from a linear time-dependent perturbation on the fission cross sections of the material 1 so that the neutron power increases during 2 seconds and then it decreases. The functions that define the time evolution of the cross sections are exposed in Appendix B.2.

First, the matrix-free performance of the Backward differential method (BKM) is studied. The time-step for the BKM has been set to 0.001 s. Table 5.1 shows the memory resources used by the code to solve the transient by using the BKM

with the *CSR* strategy and the *non-diagonal* strategy. Note that to solve the linear systems, both algorithms use the GMRES method, but the preconditioner must be different. For the first case is the ILU preconditioner and for the second strategy is the block Gauss-Seidel preconditioner. Results show that the *non-diagonal* allocation improves the computational efficiency of the code.

Table 5.1: Computational efficiency of the *non-diagonal* allocation versus the *CSR* allocation in the backward differential method for the cuboid reactor.

Allocation	**Mean its. GMRES**	**CPU memory**	**CPU Time**
CSR	6.60	580 Mb	1605 s
Non-Diagonal	7.60	186 Mb	778 s

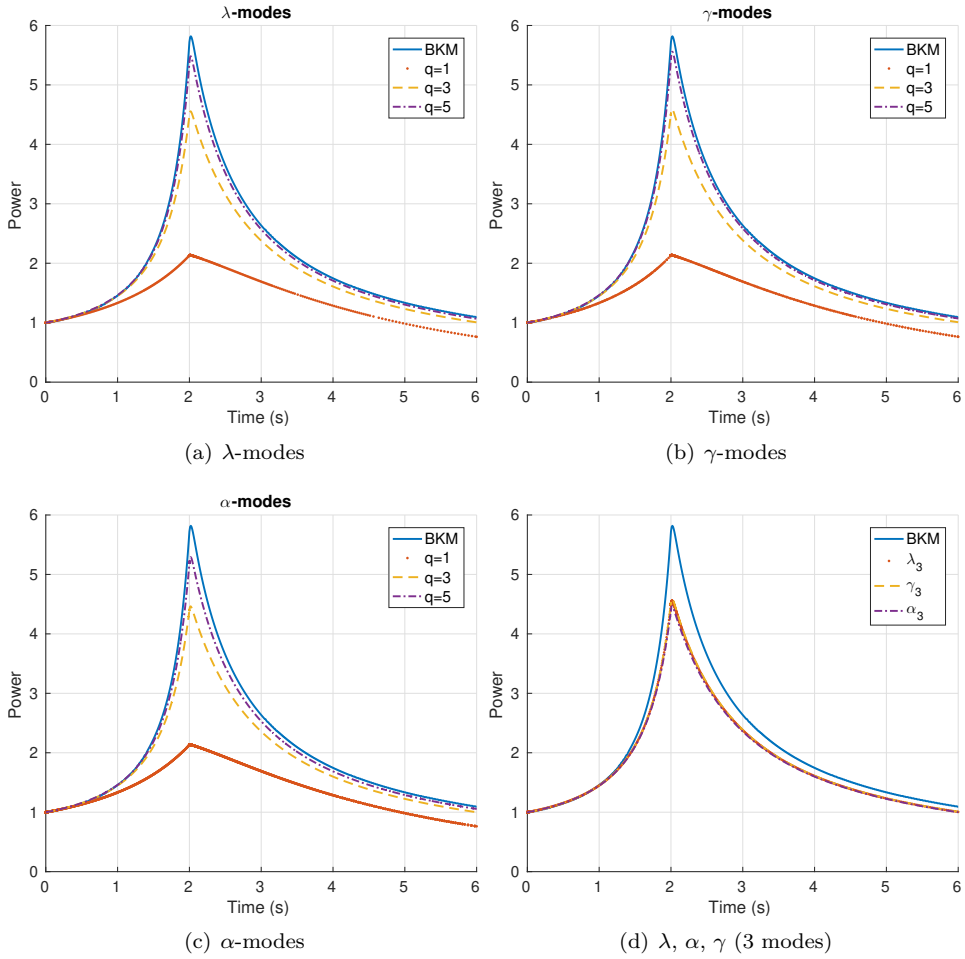
In the following, the modal method is analyzed. The five dominant λ, γ and α-modes are shown in Table 5.2.

Table 5.2: Five dominant modes for the cuboid reactor.

Mode	**λ-modes**	**γ-modes**	**α-modes**
1	1.000000	1.000000	-0.03801
2	0.975493	0.984772	-925.650
3	0.936569	0.960346	-2371.33
4	0.886212	0.928279	-4195.81
5	0.842369	0.899897	-5741.37

To compare the performance of the different modal methods, we have solved this problem using different number of modes, q. In Figure 5.1, the power evolution obtained for the transient using the λ, γ and α modes is represented, together with the power evolution obtained with the BKM, taken as a reference. This Figure shows that the obtained approximations improve when the number of modes used is increased, but this number of modes is not large enough to describe very accurately the transient. This is due to the fact that the perturbation is applied only to the material 1 and the modes have difficulties to catch the very localized character of the response of the system. A high number of modes would be required to obtain better approximations. However, even if the modal methods cannot be the best technique to approximate this transient, this is an interesting challenging problem to test the modal methods. A comparison of

the evolution of the power obtained with the different modal methods has been included (Figure 5.1(d)) by using 3 eigenvalues. In this last graphic, we do not observe big differences between the kind of modes used in the expansion of the flux.



(a) λ-modes (b) γ-modes

(c) α-modes (d) λ, α, γ (3 modes)

Figure 5.1: Evolution of the power of the cuboid transient.

Finally, we compare the computational time (CPU time) to obtain the solution by using the BKM and the modal kinetics methods with the different spatial modes. Table 5.3 displays these results for several number of eigenvalues. Moreover, this

Table includes the Mean Power Error (MPE) (taking the result obtained with the BKM as a reference) to quantify the error made by the modal approximations, the CPU time to solve the eigenvalue problem and the number of steps that uses the CVODE to solve the differential system. First, this Table shows (as it does Figure 5.1) that the MPE decreases when the number of eigenvalues considered in the modal kinetics is increased. Nevertheless, the results show that using a high number of modes is not computationally efficient, since the computational time also increases when the number of eigenvalues is higher. In comparison with the BKM, one can observe that using modal methods is much more efficient than using the BKM (in spite of a high number of eigenvalues is required to obtain accurate approximations). Between the different spatial modes, there are not large differences in the errors, but there are differences in the CPU time. The number of steps in the CVODE to solve the dynamical system associated with the α-modes is lower. In this small reactor, where the differences in the CPU times to compute the modes is not very high, the α modal method is the most efficient option.

Table 5.3: Mean Power Error (MPE) and CPU times of the modal methods to obtain the relative power of the cuboid reactor.

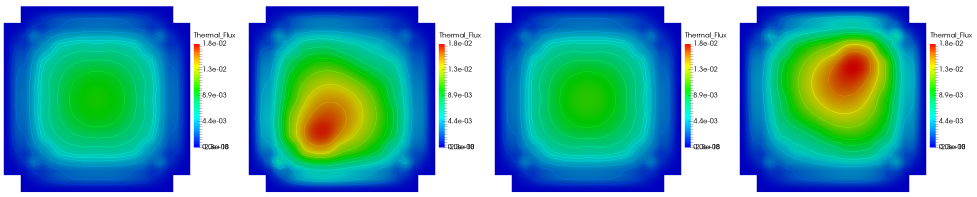
N. eigs q	**Eig. Prob. CPU time**	**n. steps**	**MPE**	**Total CPU time**
BKM	-	-	-	778 s
Modal Kinetics (λ)				
$q = 1$	0.3 s	508	2.799e-01	1 s
$q = 3$	0.7 s	1222	8.042e-02	13 s
$q = 5$	2 s	1939	2.672e-02	58 s
$q = 12$	10 s	1894	1.692e-02	349 s
Modal Kinetics (γ)				
$q = 1$	0.5 s	551	2.775e-01	1 s
$q = 3$	2 s	1048	7.781e-02	11 s
$q = 5$	3 s	1979	2.626e-02	59 s
$q = 12$	12 s	2219	1.793e-02	383 s
Modal Kinetics (α)				
$q = 1$	2 s	566	2.708e-01	3 s
$q = 3$	2 s	829	8.545e-02	10 s
$q = 5$	9 s	1145	3.139e-02	39 s
$q = 12$	19 s	1757	3.090e-02	315 s

5.4.2 Langenbuch-OPP transient. Out-of-phase perturbation

The Langenbuch 3D benchmark reactor (Langenbuch et al., 1977) is chosen to compare the modal schemes in a more realistic case. Two types of transient are defined from this reactor. The first transient is obtained by perturbing the fission cross section by two local sinusoidal perturbations out-of-phase between them. The second one is a classical movement of control rods. The details of these perturbations are described in Appendix B.3. Both transients have been computed without reactivity feedback.

First, the Langenbuch-OPP transient is analyzed. We want to highlight that the reactor is perturbed locally and this induces local changes in the spatial power distribution which makes this transient a challenging problem to be solved using spatial modal methods.

Before to start with the modal method, we have computed the power evolution by using the Backward Difference method (BKM). The time-step for the BKM has been set to 0.001 s. Figure 5.2 represents the radial average power distribution at four relevant times ($t = 0.00$ s, $t = 0.25$ s, $t = 0.50$ s and $t = 0.75$ s). It is observed that at first the maximum power goes from the center to the perturbation 1 zone then, it comes back to the center and then, it goes to the perturbation 2 zone. This behavior is repeated along the transient.

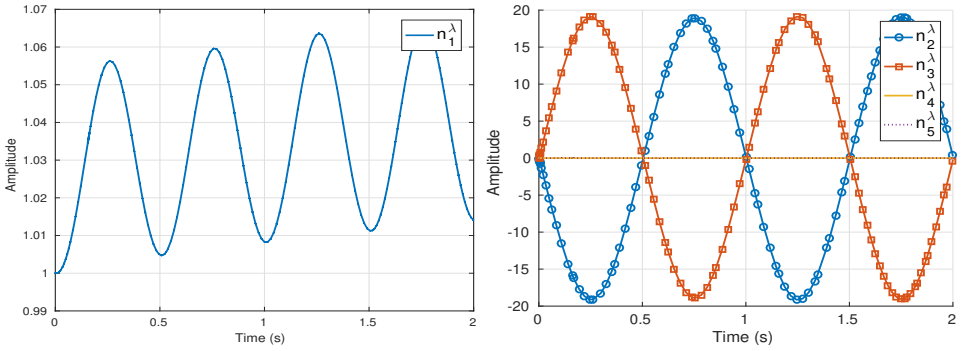


(a) $t = 0.00$ (b) $t = 0.25$ (c) $t = 0.50$ (d) $t = 0.75$

Figure 5.2: Evolution power of the out-of-phase perturbation in the Langenbuch-OPP transient.

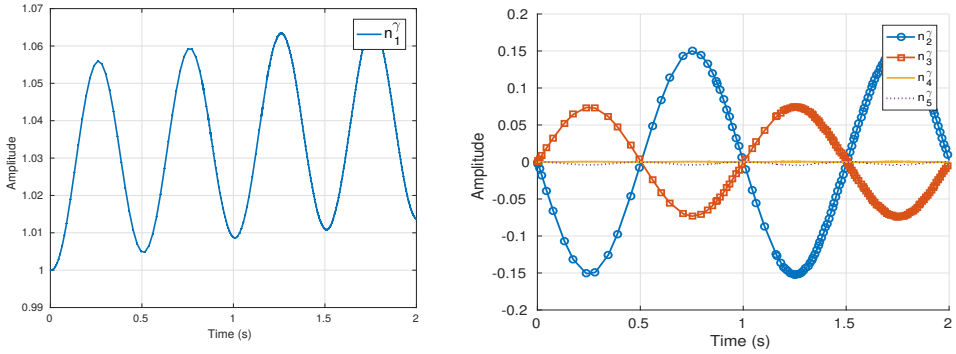
We have computed the time dependent amplitudes $n_m^{\delta}(t)$ to study the importance of the different modes during the modal representation of the neutron flux in the transient. Figures 5.3, 5.4, 5.5 represent the evolution of the amplitudes for the λ, γ and α expansion, respectively. All graphics show that the first coefficient, that corresponds to the first eigenfunction, is the one that contributes in the increasing evolution of the power, since the reactor transient starts from a near critical configuration. This coefficient is equal for all modes since the first eigenfunctions associated to each mode are very similar. Between the subcritical functions,

there are meaningful differences. All figures for subcritical harmonics show that the second and third coefficient are out-of-phase. However, these functions have different amplitude depending on the kind of mode and the number of mode. These differences are due to the different shapes that have the 2nd and 3rd eigenfunctions (Figure 3.5). The n_4^δ and n_5^δ, for all cases, are slightly oscillating, but with values close to zero during all the transient. The next coefficients are not represented since, for all modes, they are close to zero too.



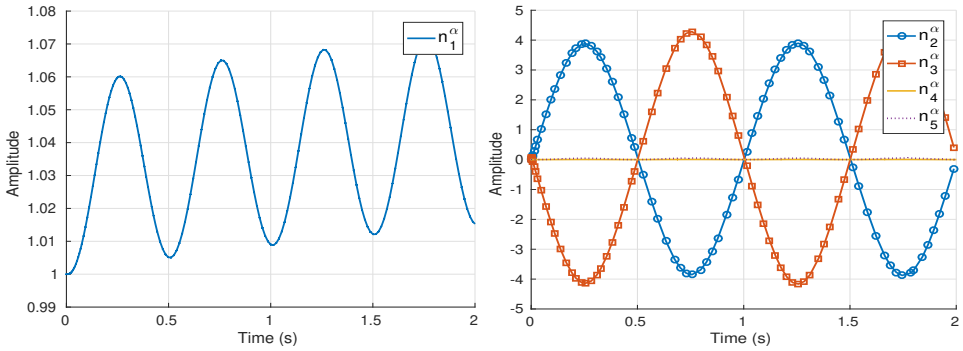
(a) 1st Harmonic (b) 2nd, 3rd, 4th, 5th Harmonics

Figure 5.3: Evolution of the amplitudes in the λ modal expansion of the Langenbuch-OPP transient.



(a) 1st Harmonic (b) 2nd, 3rd, 4th, 5th Harmonics

Figure 5.4: Evolution of the amplitudes in the γ modal expansion of the Langenbuch-OPP transient.



(a) 1st Harmonic

(b) 2nd, 3rd, 4th, 5th Harmonics

Figure 5.5: Evolution of the amplitudes in the α modal expansion of the Langenbuch-OPP transient.

Figure 5.6 shows the relative power computed with the BKM, and the relative powers computed by using the λ modal expansion for $q = 3$, $q = 6$, $q = 10$ and $q = 20$. A similar behaviour is obtained for the γ and α-modes. In this Figure, we can observe that this type of transient needs to be described with a large number of modes because the perturbations are local. This fact is also observed in the evolution of the amplitudes where these values from $q = 3$ are close to zero, and we need a lot of eigenfunctions to obtain accurate approximations for the flux distribution. Figure 5.7 displays a comparison between the relative power computed with six λ, γ and α-modes. In this graphic, we cannot appreciate a large difference between the kind of mode used to compute the total power. Figure 5.8 shows the local error (LE(t)) along the time for 6 eigenvalues. The local errors follow the same distribution as the relative power. The differences between the types modes are higher in the relative maximums of the total power, when the $\delta\Sigma_{fg}$ of the perturbations P_1 and P_2 have their relative maximums. These differences increase when the time is larger.

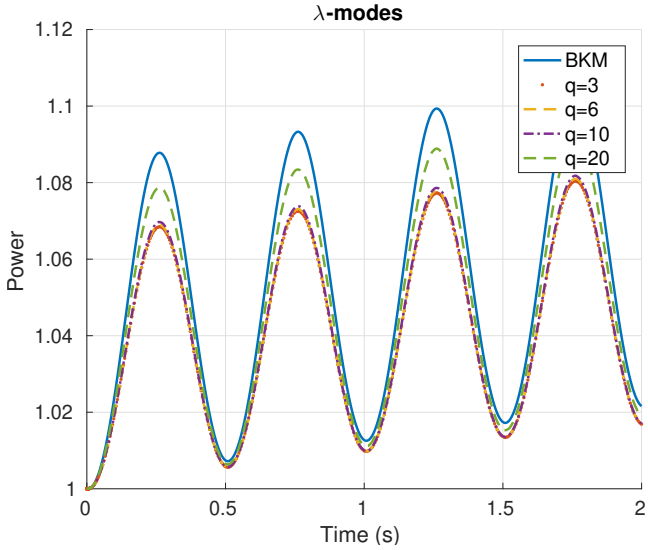


Figure 5.6: Evolution of the relative power computed with the BKM and the λ modal method of the Langenbuch-OPP transient.

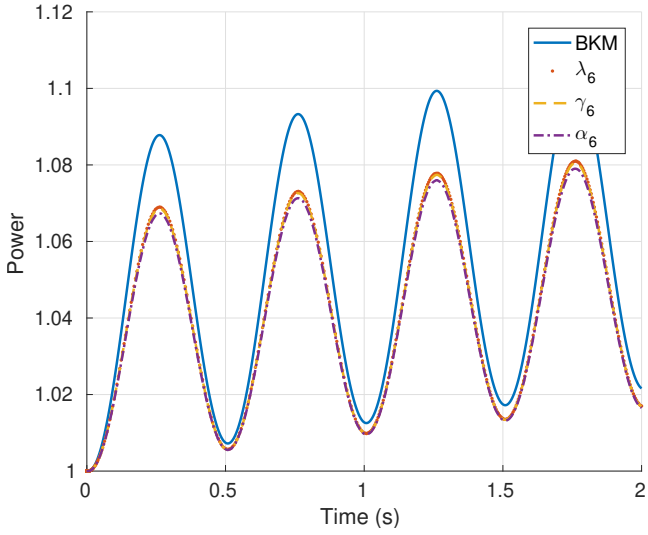


Figure 5.7: Evolution of the relative power computed with the BKM and the modal method with 6 modes for the Langenbuch-OPP transient.

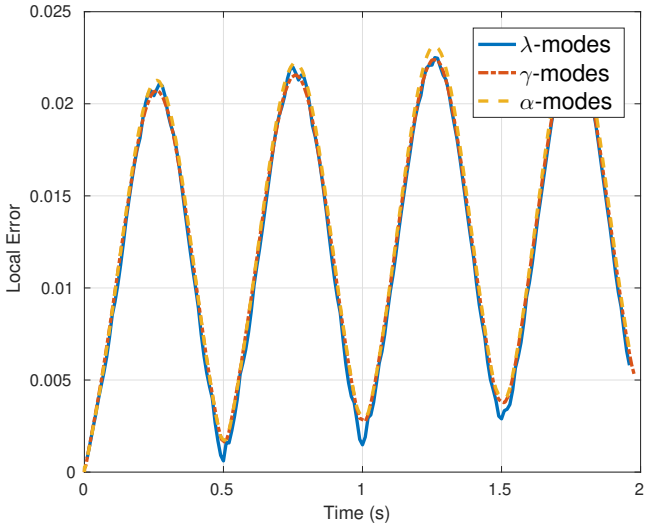
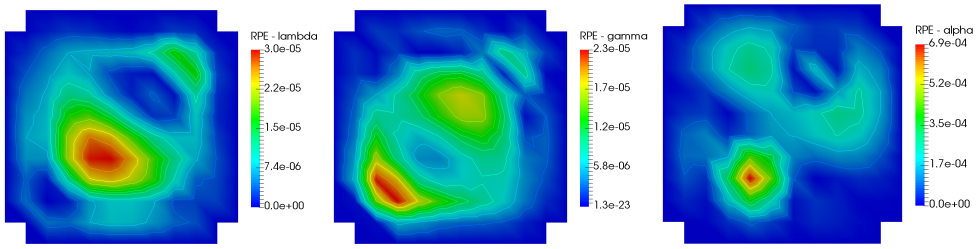


Figure 5.8: Evolution of the local error (LE(t)) along the time in the computation of 6 modes for the Langenbuch-OPP transient.

However, to study with more detail the errors of the modal methods with respect to the solution obtained with the BKM, we have computed the Radial Power Error (RPE), to observe the spatial distribution of the power error and the Mean Power Error (MPE), to quantify the total power errors between the different modal expansions used. The RPEs have been represented in Figure 5.9. These errors have been computed for $t = 0.25$ s because this is a value where the errors are higher. It is observed a different distribution of the errors for the different types of modes. More distributed errors are obtained with the γ-modes. However, in all cases higher errors are placed mainly in the cells that have been perturbed.



(a) 6 λ-modes (b) 6 γ-modes (c) 6 α-modes

Figure 5.9: Radial Power Error (RPE) of the Langenbuch-OPP transient at $t = 0.25$ s with six eigenvalues.

Table 5.4 collects the Mean Power Error (MPE), the number of steps used by the backward method of CVODE, the CPU time to compute the different eigenvalue problems and the total CPU time to integrate the neutron diffusion equation until $t = 2.0$ s. It has the data for $q = 3$, $q = 6$ and $q = 10$ and for each one of spatial modes. Moreover, this Table includes the CPU time to solve the transient problem with the BKM. The results show that less stiff systems are obtained by using the γ-modes, since we need less time steps to reach $t = 2.0$ s. However, the modal expansion that computes the solution in less time is the λ modal expansion because the eigenvalue problem associated with the λ-modes is the cheapest problem to solve. Regarding the CPU time obtained with the BKM, one could observed that this methodology takes much more time to obtain an approximation for the transient than the modal kinetics with the different spatial modes.

Table 5.4: Data of the modal methods to integrate the neutron diffusion equation for the Langenbuch-OPP transient.

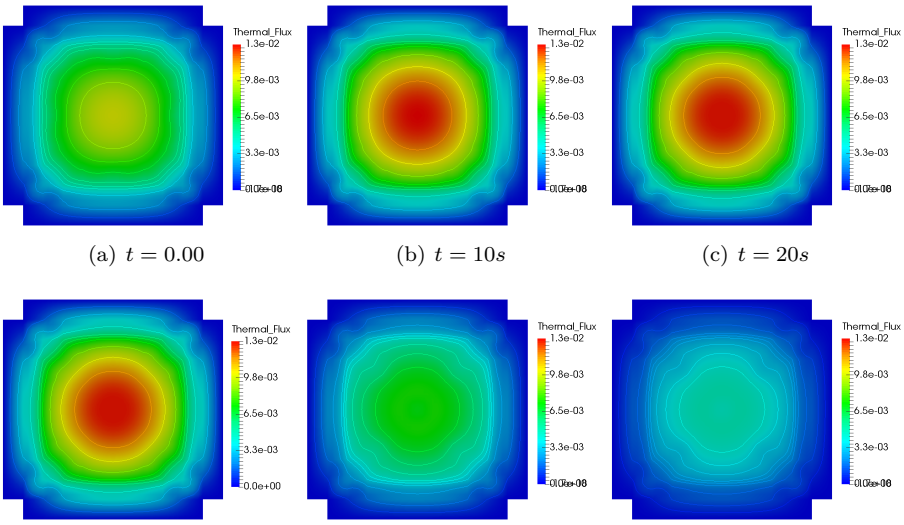
N. eigs q	Eig. Prob. CPU time	n. steps	MPE	Total CPU time
BKM	-	-	-	232 min
λ-modes				
3	0.8 min	541	1.34e-02	1.5 min
6	1.5 min	608	1.29e-02	4.8 min
10	3.3 min	543	1.26e-02	12.7 min
γ-modes				
3	1.5 min	435	1.36e-02	2.1 min
6	4.1 min	501	1.30e-02	7.1 min
10	13.8 min	520	1.27e-02	22.5 min
α-modes				
3	1.1 min	608	1.46e-02	2.8 min
6	7.2 min	530	1.34e-02	14.2 min
10	43.2 min	621	1.28e-02	59.3 min

5.4.3 Langenbuch-CRM transient. Control rods movement

In this part, the Langenbuch-CRM transient for the Langenbuch reactor is analyzed. It is defined from a classical movement of the control rods. Details can found in Appendix B.3.

First, the power evolution by using the Backward Difference method (BKM) is computed. In this transient, the time-step for the BKM has been set to $0.01s$. Figure 5.10 represents the power distribution in the middle plane at several relevant times. It is observed the radial symmetry of the movement as opposed to the previous case.

Figure 5.11 shows the relative power computed with the BKM, and the relative powers computed by using the λ modal expansion for $q = 1$, $q = 3$, $q = 6$ and $q = 15$. A similar behaviour is obtained for the γ and α-modes. It is deduced that there are not differences when a small number of modes are used. Regarding the difference with the BKM, the errors in the global relative power increases as the time increases. This behaviour is also observed in the evolution of the local error (Figure 5.12). Figure 5.13 displays a comparison between the relative power and the local error computed with 6 λ, γ and α-modes. In this graphic,



(a) $t = 0.00$ (b) $t = 10s$ (c) $t = 20s$

(d) $t = 30s$ (e) $t = 40s$ (f) $t = 50s$

Figure 5.10: Evolution power in the Langenbuch-CRM transient.

we cannot appreciate differences between the kind of mode used to compute the total power.

To study with more detail the errors of the modal methods with respect to the solution obtained with the BKM, we have computed the Mean Power Error (MPE), to quantify the total power errors between the different modal expansions used. Table 5.5 collects the MPEs, the CPU time to solve the eigenvalue problem, the number of steps used in the backward method of the CVODE and the total CPU time to integrate the systems until $t = 60.0$s. Data are exposed for $q = 1$, $q = 3$ and $q = 6$ modes. It is observed that similar errors are obtained for the different types of modes and these errors they hardly decrease when the number of eigenvalues is increased. The number of iterations in CVODE is lightly smaller for the α-modes, but their CPU time to obtain these modes is much higher than the rest of the modes. Consequently, the most efficient modal strategy is to use the λ-modes. In comparison with backward differential method (BKM), one could observe that the modal method (with any type of mode) takes much less time to obtain an approximation for the transient.

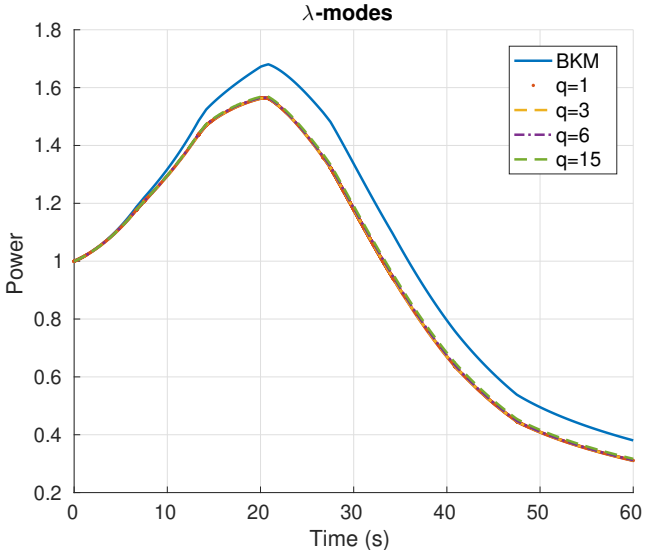


Figure 5.11: Evolution of the relative power computed with the BKM and the λ modal method of the Langenbuch-CRM transient.

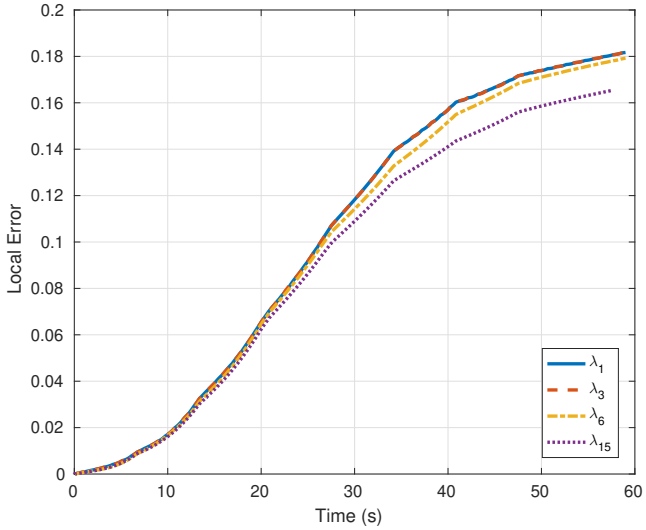
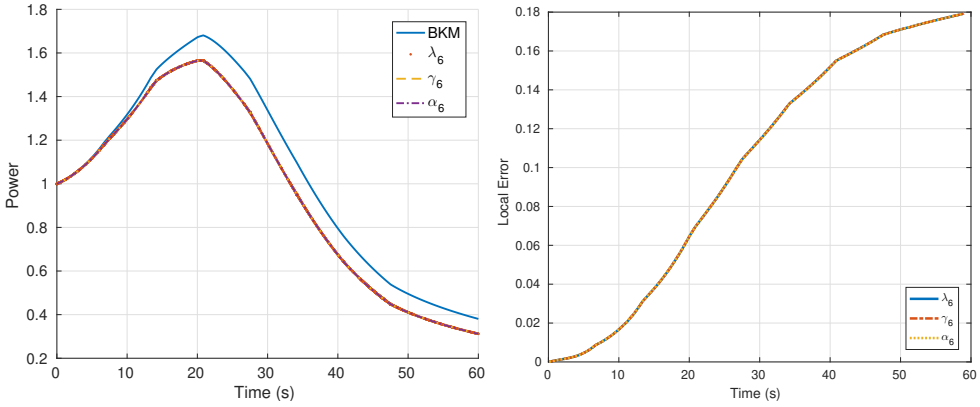


Figure 5.12: Evolution of the local error (LE) obtained with the λ modal method in the Langenbuch-CRM transient.



(a) Power (b) Local error

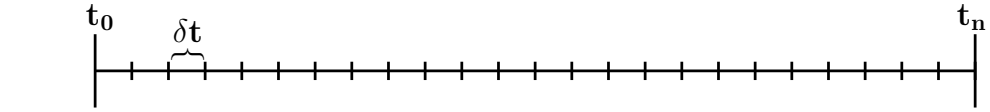
Figure 5.13: Comparison of the evolution of the relative power and local error with the different modal methods for the Langenbuch-CRM transient.

Table 5.5: Number of iterations (n. its) and computational time (s) for the computation of the modal expansion for the Langenbuch-CRM transient.

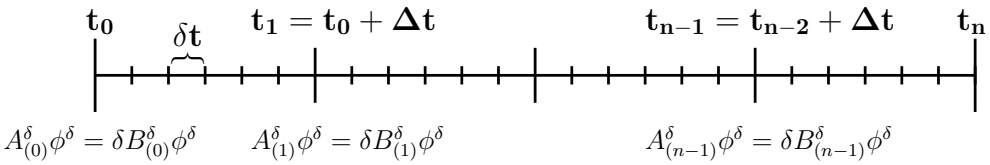
N. eigs q	**Eig. Prob. CPU time**	**n. its**	**MPE**	**Total CPU time**
BKM	-	-	-	44640 s
λ-modes				
1	15 s	2614	0.10287	91 s
3	48 s	1345	0.10286	196 s
6	99 s	3492	0.10024	2279 s
γ-modes				
1	31 s	2379	0.10288	100 s
3	93 s	1468	0.10286	257 s
6	259 s	3317	0.10026	2594 s
α-modes				
1	91 s	1530	0.10286	140 s
3	155 s	1440	0.10286	355 s
6	942 s	3237	0.10021	3286 s

5.5 Updated modal method

In realistic transient computations, the flux of the reactor over time can suffer extremely spatial variations with respect to the initial flux. Thus, in order to obtain good approximations by using the modal method, high number of modes would be needed in the modal expansion. This implies an expensive computational cost in the calculation. A solution for this challenge was proposed in (Miró et al., 2002). In this work, a small number of modes are computed but they are updated from time to time. A diagram to compare the classic modal method with the updated modal method is shown in Figure 5.14. For the classic modal method, only one eigenvalue problem is needed to be solved to obtain the solution at $t = t_n$, whereas for the updated modal method, several eigenvalue problems are defined to update the modes at each time-step equal to Δt. This thesis presents the updated modal method for the λ and extends this approach to the γ and α-modes expansions.



(a) Classic modal method



(b) Updated modal method

Figure 5.14: Modal methodology schemes.

As it is explained before, in the updated modal method, the time domain is divided into several intervals $[t_i, t_i + \Delta t] = [t_i, t_{i+1}]$. For instance, in each interval $[t_i, t_{i+1}]$, the neutron diffusion equation can be integrated by using the λ-modes associated with the problem

$$(L^i + S^i)\phi^{\lambda}_{i,m} = \frac{1}{\lambda_{i,m}} F^i \phi^{\lambda}_{i,m}, \tag{5.27}$$

where L^i, S^i and F^i are the matrices associated with the reactor at time t_i such that

$$L(t) = L^i + \delta L^i(t), \quad S(t) = S^i + \delta S^i(t), \quad F(t) = F^i + \delta F^i(t). \tag{5.28}$$

Thus, the differential equations that are needed to integrate are of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}^{i,\lambda} = \mathbf{T}^{i,\lambda}\mathbf{N}^{i,\lambda}, \tag{5.29}$$

where

$$\mathbf{N}^{i,\lambda} = \left(n_1^{i,\lambda} \cdots n_q^{i,\lambda} \quad c_{11}^{i,\lambda} \cdots c_{q1}^{i,\lambda} \quad \cdots \quad c_{1K}^{i,\lambda} \cdots c_{qK}^{i,\lambda}\right)^{\mathsf{T}}, \tag{5.30}$$

$$\mathbf{T}^{i,\lambda} = \left(\begin{array}{c|ccc} \Lambda_{i,\lambda}^{-1}\left((1-\beta)(I-\Delta F^{i,\lambda}) - [\lambda_i]^{-1} - \Delta L^{i,\lambda} - \Delta S^{i,\lambda}\right) & \Lambda_{i,\lambda}^{-1}\lambda_1^d & \cdots & \Lambda_{i,\lambda}^{-1}\lambda_K^d \\ \hline \beta_1(I+\Delta F^{i,\lambda}) & -\lambda_1^d I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_K(I+\Delta F^{i,\lambda}) & 0 & \cdots & -\lambda_K^d I \end{array}\right), \tag{5.31}$$

and

$$\begin{aligned} &\Lambda_{lm}^{i,\lambda} = \langle \phi_{i,l}^{\lambda,\dagger}, V^{-1}\phi_{i,m}^{\lambda}\rangle, \quad \Delta L_{lm}^{i,\lambda} = \langle \phi_{i,l}^{\lambda,\dagger}, \delta L^i \phi_{i,m}^{\lambda}\rangle, \\ &\Delta S_{lm}^{i,\lambda} = \langle \phi_{i,l}^{\lambda,\dagger}, \delta S^i \phi_{i,m}^{\lambda}\rangle, \quad \Delta F_{lm}^{i,\lambda} = \langle \phi_{i,l}^{\lambda,\dagger}, \delta F^i \phi_{i,m}^{\lambda}\rangle, \\ &c_{lk}^{\lambda} = \langle \phi_{i,l}^{\lambda,\dagger}, XC_k\rangle, \end{aligned} \tag{5.32}$$

where the operators δL^i, δS^i and δF^i also change into the interval $[t_i, t_{i+1}]$. This system is the same as the one used in the modal method without updating where the matrices L^i, S^i, F^i and δL^i, δS^i and δF^i are needed to be updated at each time-step t_i. Nevertheless, the initial conditions (at time t_i) must be reformulated to ensure the continuity of the solution. These initial conditions will depend on the solution in the previous interval $[t_{i-1}, t_i]$, the eigenvectors associated with direct modes ($\phi_{i,m}^{\lambda}$) and the eigenvectors associated with adjoint modes ($\phi_{i,l}^{\lambda,\dagger}$). From the solution obtained in the interval $[t_{i-1}, t_i]$, we compute the initial conditions to obtain the solution in the interval $[t_i, t_{i+1}]$.

The initial conditions for $n^{\lambda}_{i,m}$ at time t_i must be defined to solve the problem (5.29) in the interval $[t_i, t_{i+1}]$. For that purpose, we reconstruct the vector

$$\Phi(t_i) = \sum_{m=1}^{q} n^{\lambda}_{i-1,m}(t_i)\phi^{\lambda}_{i-1,m}, \tag{5.33}$$

from the variables $n^{\lambda}_{i-1,m}(t_i)$, $\phi^{\lambda}_{i-1,m}$ obtained from the integration in the interval $[t_{i-1}, t_i]$.

As the function $\Phi(t)$ must be continuous on all its domain, one could use the expansion

$$\Phi(t_i) = \sum_{m=1}^{q} n^{\lambda}_{i,m}(t_i)\phi^{\lambda}_{i,m},$$

and obtain the value of $n^{i,\lambda}_{m}(t_i)$ as

$$n^{\lambda}_{i,m}(t_i) = \frac{\langle \phi^{\lambda,\dagger}_{i,m}, F^i\Phi(t_i)\rangle}{\langle \phi^{\lambda,\dagger}_{i,m}, F^i\phi^{\lambda}_{i,m}\rangle},$$

where $\Phi(t_i)$ is computed from Equation (5.33).

To compute the initial conditions related to the concentration of the precursor k at time t_i, $c^{i,\lambda}_{l,k}(t_i) = \langle \phi^{\lambda,\dagger}_{i,l}, XC_k\rangle(t_i)$, we use the known $c^{i-1,\lambda}_{m,k}(t_i)$ computed in the previous integration on $[t_{i-1}, t_i]$. We assume that

$$\phi^{\lambda,\dagger}_{i,l} = \sum_{m=1}^{q} a^{\lambda}_{lm}\phi^{\lambda,\dagger}_{i-1,m}.$$

One could collapse the Equation (5.5) by the right along the direction of $F^{i-1}\phi^{\lambda}_{i-1,m}$ to obtain that

$$a^{\lambda}_{lm} = \frac{\langle \phi^{\lambda,\dagger}_{i,l}, F^{i-1}\phi^{\lambda}_{i-1,m}\rangle}{\langle \phi^{\lambda,\dagger}_{i-1,m}, F^{i-1}\phi^{\lambda}_{i-1,m}\rangle}. \tag{5.34}$$

Thus, the concentration of precursors at time t_i can be computed as

$$c^{i,\lambda}_{l,k}(t_i) = \langle \phi^{\lambda,\dagger}_{i,l}, XC_k\rangle(t_i) = \sum_{m=1}^{q} a_{lm}\langle \phi^{\lambda,\dagger}_{i-1,l}, XC_k\rangle(t_i) = \sum_{m=1}^{q} a_{lm}c^{i-1,\lambda}_{m,k}(t_i). \tag{5.35}$$

One can repeat the above process to update the modal expansion in each time interval $[t_i, t_{i+1}]$, but for this case, the neutron diffusion equation is integrated

by using the γ-modes associated with the problem

$$L^i \phi^{\gamma}_{i,m} = \frac{1}{\gamma_{i,m}} (F^i - S^i) \phi^{\gamma}_{i,m}, \tag{5.36}$$

where L^i, S^i and F^i are the matrices associated with the reactor at time t_i. Now, the differential equations that are needed to integrate are of the form

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{N}^{i,\gamma} = \mathbf{T}^{i,\gamma} \mathbf{N}^{i,\gamma}, \tag{5.37}$$

where

$$\mathbf{N}^{i,\gamma} = \left(n_1^{i,\gamma} \cdots n_q^{i,\gamma} \quad c_{11}^{i,\gamma} \cdots c_{q1}^{i,\gamma} \quad \cdots \quad c_{1K}^{i,\gamma} \cdots c_{qK}^{i,\gamma} \right)^{\mathsf{T}}, \tag{5.38}$$

$$\mathbf{T}^{i,\gamma} = \left(\begin{array}{c|ccc} \Lambda^{-1}_{i,\gamma}(I - [\gamma_i]^{-1} - \Delta L^{i,\gamma} - \Delta S^{i,\gamma} - \beta A^{F,\gamma} + (1-\beta)\Delta F^{i,\gamma}) & \Lambda^{-1}_{i,\gamma}\lambda_1^d & \cdots & \Lambda^{-1}_{i,\gamma}\lambda_K^d \\ \hline \beta_1(A^{F,\gamma,i} + \Delta F^{i,\gamma}) & -\lambda_1^d I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_K(A^{F,\gamma,i} + \Delta F^{i,\gamma}) & 0 & \cdots & -\lambda_K^d I \end{array} \right), \tag{5.39}$$

and

$$\begin{aligned} A^{F,\gamma,i}_{lm} &= \langle \phi^{\gamma,\dagger}_{i,l}, F^i_0 \phi^{\gamma}_{i,l} \rangle, & \Delta L^{i,\gamma}_{lm} &= \langle \phi^{\gamma,\dagger}_{i,l}, \delta L^i \phi^{\gamma}_{i,l} \rangle, \\ \Delta S^{i,\gamma}_{lm} &= \langle \phi^{\gamma,\dagger}_{i,l}, \delta S^i \phi^{\gamma}_{i,l} \rangle, & \Delta F^{i,\gamma}_{lm} &= \langle \phi^{\gamma,\dagger}_{i,l}, \delta F^i \phi^{\gamma}_{i,l} \rangle, \\ c^{i,\gamma}_{lk} &= \langle \phi^{\gamma,\dagger}_{i,l}, X C_k \rangle, & \Lambda^{i,\gamma}_{lm} &= \langle \phi^{\gamma,\dagger}_{i,l}, V^{-1} \phi^{\gamma}_{i,l} \rangle. \end{aligned} \tag{5.40}$$

where as before, the operators δL^i, δS^i and δF^i also change into the interval $[t_i, t_{i+1}]$.

To reconstruct the initial conditions for $n^{i,\gamma}_m$ at time t_i, a process equal to the λ-modes updating is applied but in this case the value of $n^{i,\gamma}_m$ is computed as

$$n^{i,\gamma}_m(t_i) = \frac{\langle \phi^{\gamma,\dagger}_{i,m}, (F^i - S^i)\Phi(t_i) \rangle}{\langle \phi^{\gamma,\dagger}_{i,m}, (F^i - S^i)\phi^{\gamma}_{i,m} \rangle},$$

because of the orthogonality between the adjoint γ-modes and the direct γ-modes is satisfied for the inner product associated with the matrix $(F^i - S^i)$.

Likewise, the concentration of precursors at time t_i can be computed as

$$c_{l,k}^{i,\gamma}(t_i) = \sum_{m=1}^{q} a_{lm}^{\gamma} c_{m,k}^{i-1,\gamma}(t_i),$$

where

$$a_{lm}^{\gamma} = \frac{\langle \phi_{i,l}^{\gamma,\dagger}, (F^{i-1} - S^{i-1})\phi_{i-1,m}^{\gamma} \rangle}{\langle \phi_{i-1,m}^{\gamma,\dagger}, (F^{i-1} - S^{i-1})\phi_{i-1,m}^{\gamma} \rangle}.$$

Finally, for the α-modes, it is solved in each time-step $[t_i, t_{i+1}]$ the eigenvalue problem

$$(F^i + S^i - L^i)\phi_{i,m}^{\alpha} = \alpha_{i,m}[V]^{-1}\phi_{i,m}^{\gamma}, \tag{5.41}$$

For this case, the differential equations are of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{N}^{i,\alpha} = \mathbf{T}^{i,\alpha}\mathbf{N}^{i,\alpha}, \tag{5.42}$$

where

$$\mathbf{N}^{i,\alpha} = \begin{pmatrix} n_1^{i,\alpha} \cdots n_q^{i,\alpha} & c_{11}^{i,\alpha} \cdots c_{q1}^{i,\alpha} & \cdots & c_{1K}^{i,\alpha} \cdots c_{qK}^{i,\alpha} \end{pmatrix}^{\mathsf{T}}, \tag{5.43}$$

$$\mathbf{T}^{i,\alpha} = \left(\begin{array}{c|ccc} [\alpha_i] - \Delta L^{i,\alpha} - \Delta S^{i,\alpha} - \beta A^{F,\alpha} + (1-\beta)\Delta F^{i,\alpha} & \lambda_1^d & \cdots & \lambda_K^d \\ \hline \beta_1(A^{F,\alpha,i} + \Delta F^{i,\alpha}) & -\lambda_1^d I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_K(A^{F,\alpha,i} + \Delta F^{i,\alpha}) & 0 & \cdots & -\lambda_K^d I \end{array}\right), \tag{5.44}$$

and

$$\begin{aligned} A_{lm}^{F,\alpha,i} &= \langle \phi_{i,l}^{\alpha,\dagger}, F_0^i \phi_{i,l}^{\alpha} \rangle, & \Delta L_{lm}^{i,\alpha} &= \langle \phi_{i,l}^{\alpha,\dagger}, \delta L^i \phi_{i,l}^{\alpha} \rangle, \\ \Delta S_{lm}^{i,\alpha} &= \langle \phi_{i,l}^{\alpha,\dagger}, \delta S^i \phi_{i,l}^{\alpha} \rangle, & \Delta F_{lm}^{i,\alpha} &= \langle \phi_{i,l}^{\alpha,\dagger}, \delta F^i \phi_{i,l}^{\alpha} \rangle, \\ c_{lk}^{i,\alpha} &= \langle \phi_{i,l}^{\alpha,\dagger}, XC_k \rangle. \end{aligned} \tag{5.45}$$

where as the previous cases, the operators δL^i, δS^i and δF^i also change into the interval $[t_i, t_{i+1}]$.

To reconstruct the initial conditions for $n_m^{i,\alpha}$ at time t_i, the value of $n_m^{i,\alpha}$ is computed as

$$n_m^{i,\alpha}(t_i) = \frac{\langle \phi_{i,m}^{\alpha,\dagger}, [V^{-1}]\Phi(t_i)\rangle}{\langle \phi_{i,m}^{\alpha,\dagger}, [V^{-1}]\phi_{i,m}^{\alpha}\rangle},$$

because of the orthogonality between the adjoint α-modes and the direct α-modes is satisfied for the inner product associated with the matrix $[V^{-1}]$.

The concentration of precursors at time t_i is computed as

$$c_{l,k}^{i,\alpha}(t_i) = \sum_{m=1}^{q} a_{lm}^{\alpha} c_{m,k}^{i-1,\alpha}(t_i),$$

where

$$a_{lm}^{\alpha} = \frac{\langle \phi_{i,l}^{\alpha,\dagger}, [V^{-1}]\phi_{i-1,m}^{\alpha}\rangle}{\langle \phi_{i-1,m}^{\alpha,\dagger}, [V^{-1}]\phi_{i-1,m}^{\alpha}\rangle}.$$

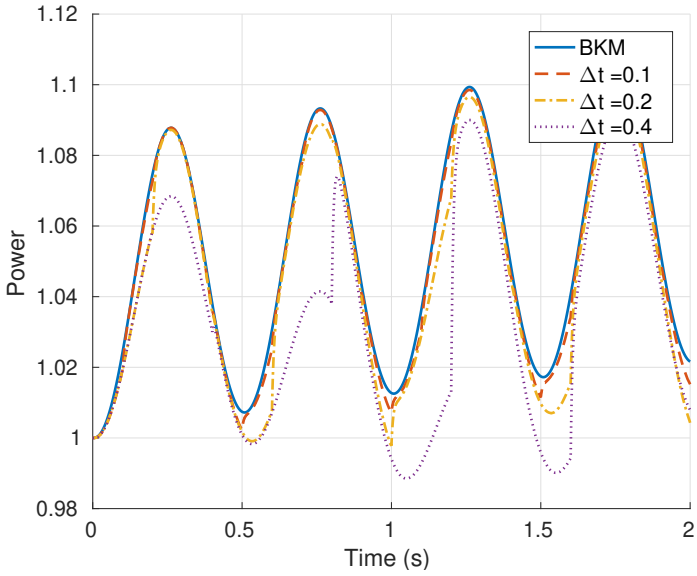
5.6 Numerical results for the updated modal method

Numerical results in Section 5.4 have shown that the modal methodology without updating needs a high number of modes to obtain accurate results. In the following, the updated modal method is studied using the two transients defined for the Langenbuch reactor (Appendix B.3): the out-of-phase sinusoidal perturbation on the fission cross-sections (Langenbuch-OPP transient) and the movement of the control rods (Langenbuch-CRM transient).

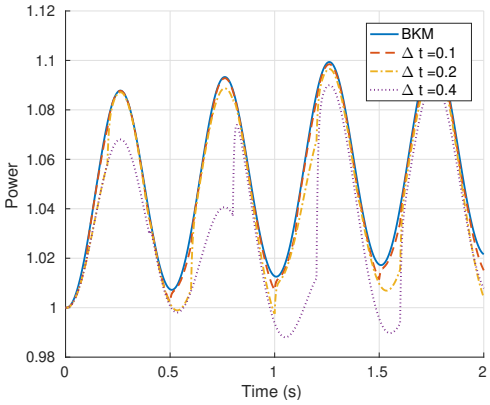
In the case of the updated modal method, the hybrid method, that is used to compute the solution of the eigenvalue problems at each time-step t_i, has been initialized from the solutions computed in the previous time-step t_{i-1}.

5.6.1 Langenbuch-OPP transient. Out-of-phase perturbation

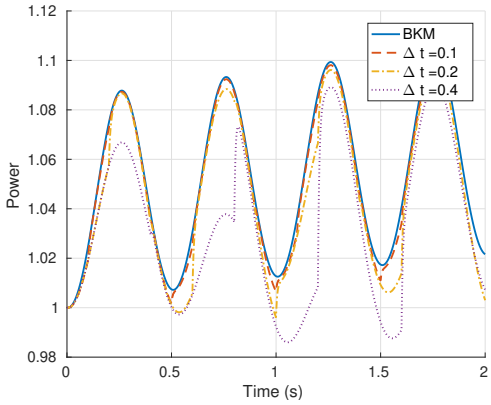
The evolution of the global power computed with the backward differential method (BKM) with $\Delta t = 0.001$ s and with the updated modal method with several fix time-steps Δt is represented in Figure 5.15. The updated modal method for the different types of modes is applied with 3 modes. It is observed that large errors between the BKM and the updated modal method are produced when the perturbations reach their maximums. However, these differences are reduced for small time-steps. The behaviour of the modal methods using the different spatial modes is the same.



(a) λ-modes



(b) γ-modes



(c) α-modes

Figure 5.15: Average relative power computed with the updated modal method with 3 modes for the Langenbuch-OPP transient.

We analyze the local error (difference between the power obtained with the BKM and the modal method) for some settings of the updated modal method. Table 5.6 shows that small time-steps in the updating gives more accurate approximations. In contrast, the CPU times are also higher. Good approximations are obtained

with updating time-steps equal to 0.2 s. However, these parameters can change for other reactor computations. The Table shows more efficient results for the λ modal method. Figure 5.16 represents the local error along the time for the λ modal method for several time-steps. It shows non-uniform errors along the time and high values for the errors near to the extremes of the power (maximums and minimums) and before to the modal updating. The same conclusions for the error are obtained by using the α and γ modal expansions.

Table 5.6: Performance of the Updated modal method with fixed time-step.

N. eigs. (q)	**Δt**	**MPE**	**CPU Time**
λ-modes			
3	0.05 s	1.50e-03	45 min
3	0.10 s	3.29e-03	25 min
3	0.20 s	7.41e-03	13 min
3	0.40 s	2.00e-02	6 min
γ-modes			
3	0.10 s	3.30e-03	30 min
3	0.20 s	7.51e-03	17 min
3	0.40 s	2.03e-02	7 min
α-modes			
3	0.10 s	5.56e-03	41 min
3	0.20 s	7.74e-03	24 min
3	0.40 s	2.11e-02	11 min

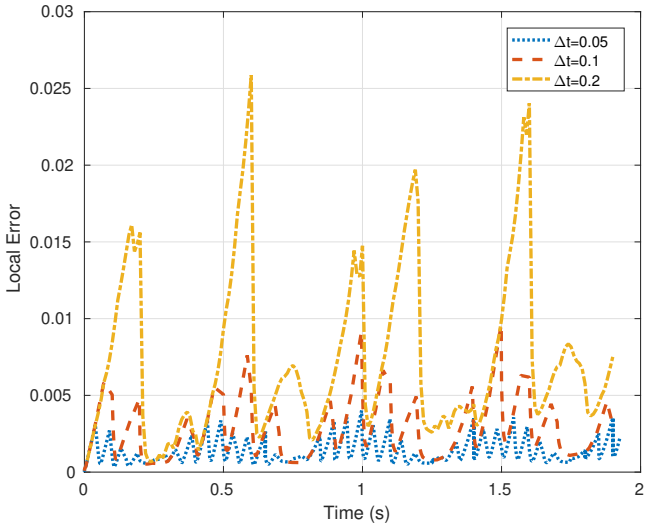


Figure 5.16: Evolution of local error in the Langenbuch-OPP transient in the λ updated modal method with the λ-modes.

5.6.2 Langenbuch-CRM transient. Control rods movement

In this Section, the updated modal methodology is analyzed for the transient defined from the control rods movement in the Langenbuch reactor (Appendix B.3). In this transient, only the modal method for the λ-modes and the γ-modes is presented because when the reactor is supercritical, it is very difficult to obtain a convergence in the eigenvalue solvers for the α-modes. The updated modal method for the λ and γ-modes is applied with 1 mode because it is enough to describe this transient.

Figure 5.17 represents the evolution of the power computed with the backward differential method with $\Delta t = 0.01$ s and with the λ modal method with several fix time-steps. The errors in the global power between the BKM and the updated modal method are mainly produced at the maximum of the relative power. These differences are reduced as the time-steps decrease. Same distribution is obtained with the γ-modes expansion. Note that, in this transient the errors are much smaller than the ones obtained for the Langenbuch-OPP transient.

Table 5.7 displays some relevant data for the updated modal method for the λ and γ-modes to quantify the errors with the BKM and to compare the different spatial modes expansions. As the graphic has showed, the errors are smaller as

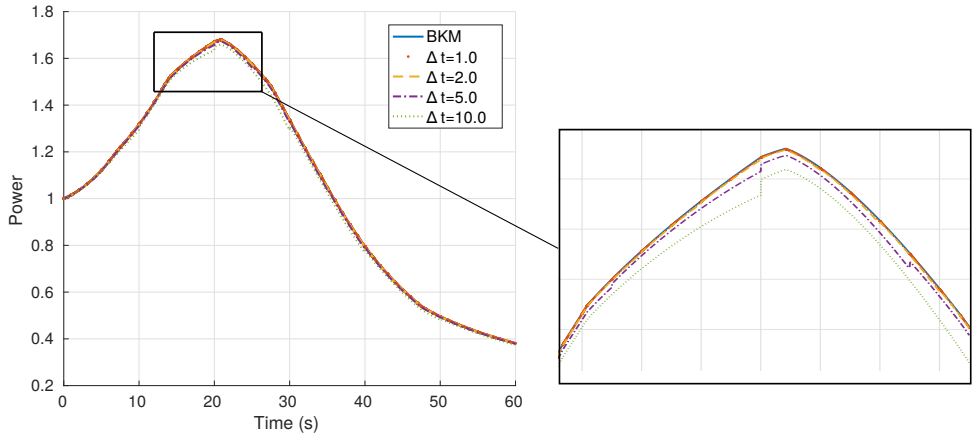


Figure 5.17: Evolution of the relative power computed with the updated modal method with 1 λ-modes for the Langenbuch-CRM transient.

the time-step is reduced. For the γ-modes cases, lightly small errors are obtained, but the CPU times are higher than for the λ modal cases.

Table 5.7: Performance of the updated modal method with fixed time-step in the Langenbuch-CRM transient.

N. eigs. (q)	**Δt**	**MPE**	**CPU Time**
λ-modes			
1	1.0 s	7.59e-04	16 min
1	2.0 s	1.33e-03	10 min
1	5.0 s	5.60e-03	4 min
1	10.0 s	1.73e-02	2.5 min
γ-modes			
1	1.0 s	7.47e-04	23 min
1	2.0 s	1.25e-03	13 min
1	5.0 s	5.55e-03	6 min
1	10.0 s	1.72e-02	3.5 min

The evolution of the local error obtained with the λ updated modal method with several fix time-steps can be observed in Figure 5.18. The local error in the power increases along the time until $t \approx 40$ s and then, it decreases. It can also observe that the errors decrease just at times in which the λ-mode is updated.

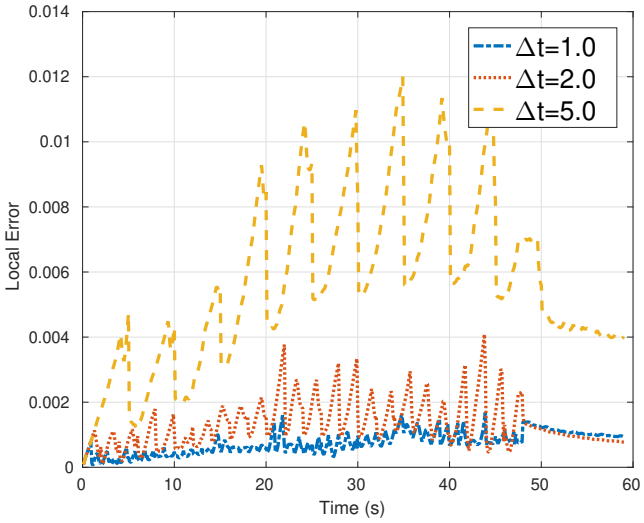


Figure 5.18: Evolution of local error in the Langenbuch-CRM transient with the updated modal method and one λ-mode.

5.7 Adaptive modal method

The updating with a fix time-step (Δt) implies several limitations. First, one needs to select a time-step previously that leads to obtaining results with unpredictable errors. If, one sets the method with a small time-step to ensure good approximations may not be necessary for some stages of the transient. Moreover, the computational cost also increases, since most of the time in the modal computation is spent in the solutions of the eigenvalue problems. On the other hand, if we use large time-steps, we could obtain not very accurate results. For these reasons, there is a big interest to implement an adaptive control of the time-step for the updated modal methods that selects the time-step for the modes updating according to the transient analyzed. To do so, there are two fundamental issues that are needed to study:

1. The error estimation due to the modal expansion assumption.
2. A suitable constraint to select the time-step based on the error estimation.

Several approaches are considered in the following subsections.

5.7.1 Estimate of the local error of shape

After applying the modal methodology, the error that it is obtained comes (essentially) from the assumption that the neutronic flux can be described as a finite linear combination of the spatial modes, since they do not form a complete basis of the function space. It is reasonable to assume that larger variations in the flux will imply larger errors in the modal method.

The first error estimate is based on the difference between eigenfunctions. This approach will be called the *modal difference error*. One can compute the modes in the next time-step to predict how the total flux will change. According to the difference with the previous modes, one can define an error as

$$\varepsilon_{md}^{i} = \max_{m} \frac{\|\phi_{i-1,m}^{\delta} - \phi_{i,m}^{\delta}\|_1}{\|\phi_{i-1,m}^{\delta}\|_1} k_{md},$$

where k_{md} is a constant to adjust the accuracy of the approximation. Its value will depend on the transient analyzed.

The previous estimation requires to compute eigenvalue problems to estimate the error, that it can be computationally very expensive.

The second approach is based on the residual error that appears when the actual modes are substituted on the problem in other time step. Larger residual errors are obtained when the eigenfunctions change more spatially. For that, we define the *modal residual error* as

$$\varepsilon_{mr} = \max_{m} \frac{\|A^{\delta,i}\phi_{i-1,m}^{\delta} - \delta_{m}^{i-1} B^{\delta,i}\phi_{i-1,m}^{\delta}\|_1}{\|\phi_{i-1,m}^{\delta}\|_1} k_{mr},$$

where k_{mr} is the constant for this type of error.

Finally, we assume that the flux along the time will change depending on the variation in the cross-sections. For this reason, we define the *cross-section perturbation error* as

$$\varepsilon_{xs} = \sum_{c} \frac{\|\mathrm{XS}^{i-1}(c) - \mathrm{XS}^{i}(c)\|_1}{\|\mathrm{XS}^{i-1}(c)\|_1} k_{xs},$$

where k_{xs} is the accuracy constant, the value of c denotes the cell c of the reactor and XS, one cross-section type that depends on the perturbation applied to the transient. This estimation is the cheapest estimation and it is used in other neutronic codes. Note that same *cross-section perturbation error* are obtained when a cross-section is increased or decreased with the same value, but the response in the relative power is not the same.

5.7.2 Control algorithm for the shape time steps

Once the error estimation is selected, it is necessary to define the control algorithm to compute the time-step from the error estimation. For that, we have studied two strategies.

The first one depends on the error in the previous step in a fixed way. It is called as *banded* time-step control. If the error is greater than some value max_{le}, the step-time is divided by 2. On the other hand, if the error is lower than other value min_{le}, the next step-time is multiplied by 2. For errors between min_{le} and max_{le}, the step-time remains constant. It can be written as

$$\Delta t_i = \begin{cases} \Delta t_{i-1} * 2, & \varepsilon < min_{le}, \\ \Delta t_{i-1}, & min_{le} < \varepsilon < max_{le}, \\ \Delta t_{i-1} \,/\, 2, & max_{le} < \varepsilon, \end{cases} \tag{5.46}$$

where ε is some error estimation presented in the previous Section. The values of min_{le} and max_{le} are reactor dependent. In the numerical results cases, the value of min_{le} has been fix to 1.0 and the value of $max_{le} = 2.0$.

The second option to select the time-step is based on the control algorithms defined for other differential methods implemented for stiff problems (Wanner and Hairer, 1996). It is called as *dynamic* time-step control. In particular, we obtain the step Δt_i as

$$\Delta t_i = \Delta t_{i-1} \min\{2.0, \max\{0.5, \sqrt{1.0/\varepsilon}\}\}, \tag{5.47}$$

where ε is some error defined in Section 5.7.1.

5.8 Numerical results for the adaptive updated modal method

Numerical results in Section 5.6 have shown that the updated modal methodology with fix time-steps requires to use small time-steps to obtain accurate approximations in some instants of the transients. However, there are also moments where these small time-step are not necessary. In the following, the performance of the adaptive updated modal method is studied for the transients defined for the Langenbuch reactor: the Langenbuch-OPP transient and Langenbuch-CRM transient. As in the updated modal method, the hybrid method, that is used to compute the solution of the eigenvalue problems at each time-step t_i, has been initialized from the solutions computed in the previous time-step t_{i-1}.

5.8.1 Langenbuch-OPP transient. Out-of-phase perturbation

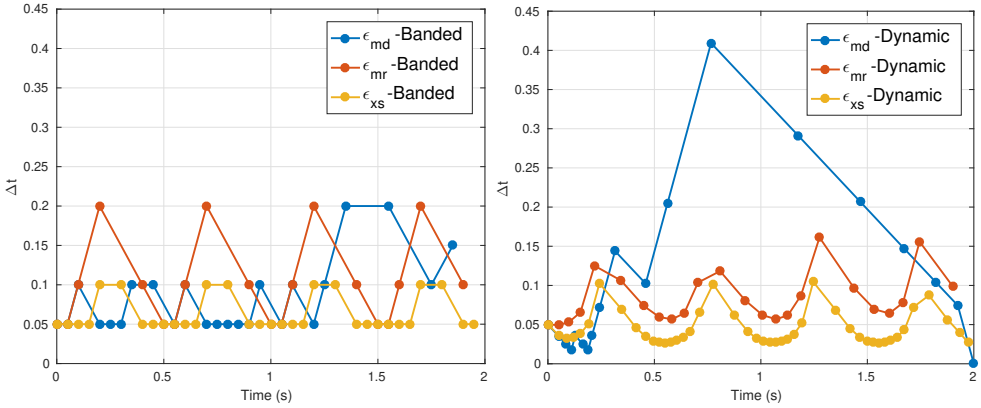
In this Section, the adaptive updated modal methodology with the λ-modes for the out-of-phase perturbation is studied. The number of modes for the modal method has been set to 3.

The initial time-step for all strategies has been set to $\Delta t_0 = 0.05$ s. The accuracy coefficients for the errors has been $k_{md} = 1.0$, $k_{mr} = 100$ and $k_{xs} = 1.0$. The cross-section used for the *cross-section perturbation error* has been the fission cross-section. Table 5.8 shows the Mean Power Errors and CPU times obtained by setting the different error estimations and control errors with three λ-modes. One can deduce that the *modal difference error* (ε_{md}) is not very efficient because it needs to compute the modes to estimate the error and it is very expensive. Regarding the other error estimations, the *cross-section perturbation error* (ε_{mr}) gives lower errors, but in more time than the *modal residual error*. If the type of control error is compared, in general, the dynamic control gives better approximations than the banded control. Figure 5.19 shows the time-steps obtained for each one of the settings. It can be observed a similar behaviour in the computation of the time-step for the *modal residual error* and the *cross-section perturbation error*, but for this last error, the time-steps computed are slightly smaller.

If we compare the local error for the updated modal method with fixed $\Delta t = 0.1$ s and for the adaptive modal method with the *modal residual error* and *dynamic* control time-step, both strategies obtain similar mean power errors (MPE $\approx 3 \cdot 10^{-3}$) but with the adaptive control time-step the results are obtained in less time. Figure 5.20 displays the evolution of the local error in these two cases. More distributed errors are obtained by using an adaptive time-step control.

Table 5.8: Errors and CPU time obtained with the adaptive time-step modal method for the Langenbuch-OPP transient.

	Banded Control		Dynamic Control	
Type of Error	**MPE**	**CPU Time**	**MPE**	**CPU Time**
ε_{md}	4.14e-03	30min	1.06e-02	38 min
ε_{mr}	2.49e-03	21min	2.50e-03	21 min
ε_{xs}	1.82e-03	30min	1.28e-03	43 min



(a) Banded Error Control

(b) Dynamic Error Control

Figure 5.19: Evolution of the Δt in the adaptive time-step control for the Langenbuch-OPP transient.

Finally, all types of modal strategies with the λ-modes and the BKM are compared in Table 5.9. All modal methods are set with 3 eigenvalues. The updated modal method with fixed time-step is set with $\Delta t = 0.1s$. The results of the adaptive updated modal method are obtained with initial time-step $\Delta t_0 = 0.05$ s, *modal residual error* and *dynamic* control time-step. Table shows that the adaptive updated modal methods gives the most accurate results with a CPU time smaller than the obtained with the updated modal method with fixed time-step. Modal methods reduce considerably the CPU time obtained with the BKM.

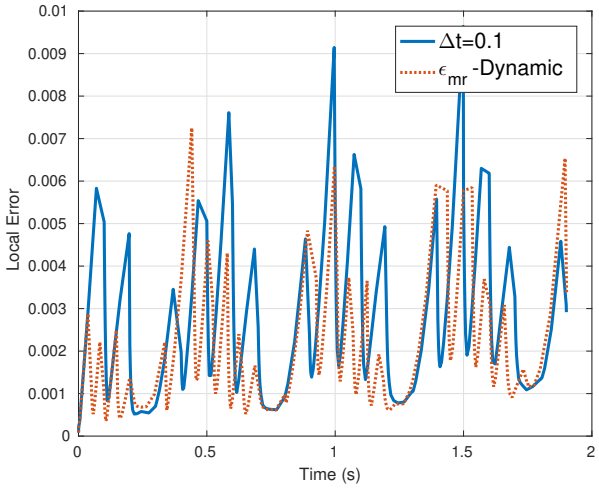


Figure 5.20: Comparison of the evolution of the local error with the fixed updated modal method and the adaptive updated modal method for the Langenbuch-OPP transient.

Table 5.9: Errors and CPU time obtained to integrate the Langenbuch-OPP transient.

	BKM	Modal	Updated Modal	Adaptive Upd. Mod.
MPE		1.34e-02	3.29e-03	2.49e-03
CPU Time	232 min	1.5 min	25 min	21 min

5.8.2 Langenbuch-CRM transient. Control rods movement

The adaptive updated modal methodology is analyzed for the Langenbuch-CRM transient only for the λ-modes. In this reactor, only one mode is used in the modal expansion.

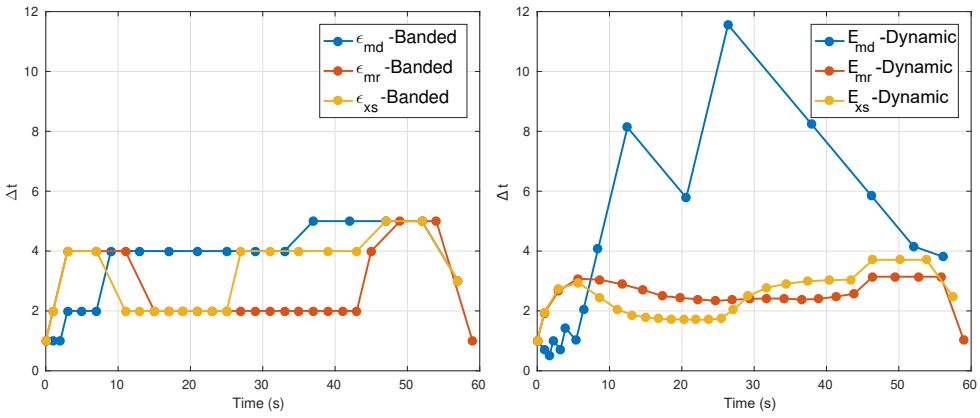
The initial time-step for all strategies has been set to $\Delta t_0 = 1.0$ s. The accuracy coefficients for the errors has been $k_{md} = 1.0$, $k_{mr} = 300$ and $k_{xs} = 1.0$. The cross-section used for the *cross-section perturbation error* has been the absorption cross-section because the insertion of control rods in this benchmark only changes this cross-section.

Table 5.10 shows the Mean Power Errors and CPU times obtained by setting the different error estimations and control time-steps with one λ-mode. As the previous transient, the *modal difference error* (ε_{md}) is not very efficient. Between

the other error estimations, the *cross-section perturbation error* (ε_{xs}) gives better results in less time. If the type of control error is compared, there are not relevant differences between the dynamic control and banded control. Figure 5.21 shows the time-steps obtained for each one of the settings. It is deduced that the *modal residual error* decreases the time-step when the local error is higher, while the *cross-section perturbation error* and *the modal difference error* reduce the time-step when the perturbation in the cross-section is higher.

Table 5.10: Errors and CPU time obtained with the adaptive time-step modal method for the Langenbuch-CRM transient.

	Banded Control		Dynamic Control	
Type of Error	**MPE**	**CPU Time**	**MPE**	**CPU Time**
ε_{md}	3.5e-03	10.7 min	9.6e-03	14.6 min
ε_{mr}	2.0e-03	8.2 min	1.8e-03	8.9 min
ε_{xs}	2.5e-03	6.5 min	1.7e-03	8.3 min



(a) Banded Error Control (b) Dynamic Error Control

Figure 5.21: Evolution of the Δt in the adaptive time-step control for the Langenbuch-CRM transient.

If we compare the results obtained with the fixed updated modal method (Table 5.7) with the adaptive updated modal method, one can deduce that similar errors with less CPU time are obtained with the adaptive modal method. Figure 5.22 displays the local error for the updated modal method with fixed Δt and for the adaptive modal method with the *modal residual error* and *dynamic*

control time-step. More distributed errors are obtained by using an adaptive time-step control in spite of the MPE for the fixed modal method is smaller.

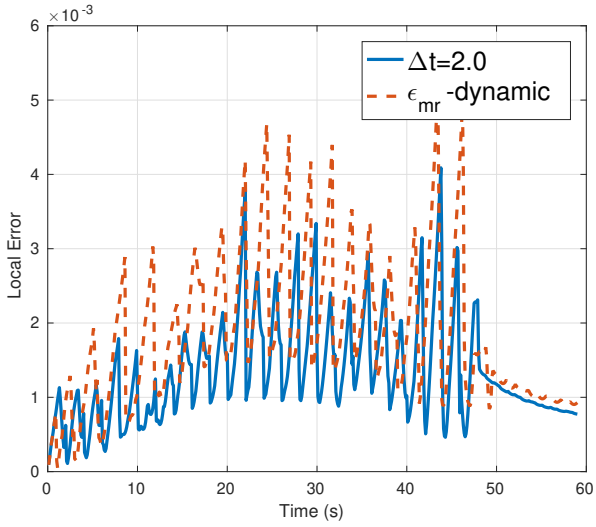


Figure 5.22: Comparison of the evolution of the local error with the fixed updated modal method and the adaptive updated modal method for the Langenbuch-CRM transient.

Finally, all types of modal strategies with the λ-modes and the BKM are compared in Table 5.11. All modal methods are set with 1 eigenvalue. The updated modal method with fixed time-step is set with $\Delta t = 2.0s$. The results of the adaptive updated modal method are obtained with initial time-step $\Delta t_0 = 1.0$ s, *modal residual error* and *dynamic* control time-step. Table shows similar errors between the updated modal method with fixed and adaptive time-step, but in the last case in less CPU Time. Modal methods reduce considerably the CPU time obtained with the BKM.

Table 5.11: Errors and CPU time obtained to integrate the Langenbuch-CRM transient.

	BKM	**Modal**	**Updated Modal**	**Adaptive Upd. Mod.**
MPE		1.0e-01	1.3e-03	1.7e-03
CPU Time	744 min	1.5 min	10 min	8.3 min

CHAPTER 6

CONCLUSIONS

The neutron transport equations describes the distribution of the neutron population in a nuclear reactor. A fast and accurate solution is essential for the design and safe operation of a nuclear reactor and other nuclear systems. As the solution of this equation is not straightforward, numerical approximations must be considered. In this thesis, different methodologies have been studied and implemented to integrate efficiently the multigroup neutron diffusion equation and the simplified spherical harmonics equation.

Chapter 3 has presented the high order finite element method to spatially discretize the steady-state approximations. First, the algebraic expressions have been obtained for the spatial modes associated with the neutron diffusion equation. Small numerical errors have been obtained using polynomial degrees larger or equal to 2 in the finite element method, for each kind of spatial mode. If the type of modes is compared, several remarks can be deduced. For critical configurations of reactors, the spatial distribution of the first eigenfunction corresponding to the dominant eigenvalue is equal, but the spatial distributions for the next eigenfunctions are very different. In subcritical configurations, the first eigenfunctions are not equal. However, while for the λ and γ-modes, the spatial distribution does not change very much between them, for the α-modes case, this difference is more evident. Furthermore, the differences between the spatial distribution in critical state and the spatial distribution in subcritical states are higher in α-modes than for the rest of the spatial modes. With respect to the spectrum of the modes, this is more clustered for the γ-modes case.

For the SP_N equations, a set of λ-modes have been computed with different type of meshes. The SP_N equations have been proven to be a useful approximation to the neutron transport equation especially for full core nuclear reactor calculations. Accurate results are obtained with polynomial degrees larger or equal to 2 in the finite element discretization. The conclusions for two-dimensional and three-dimensional reactors are the same, but for the three-dimensional case the size of the problem rises enormously.

Chapter 4 presents and compares different methodologies to solve the generalized eigenvalue problems obtained from the finite element discretization. First, the computation of the spatial modes are compared. The α-modes, in critical configurations, gives an ill-conditioned matrix, since the eigenvalues are close to zero. Thus, the CPU time needed to solve any lineal systems with the matrix associated with these modes is higher than the time for better conditioned matrices that appear in the other modes. Consequently, computing α-modes with eigensolvers, that need to solve many lineal systems with this matrix, as the Krylov-Schur method, is not reasonable. For subcritical configurations, the matrices become better conditioned, but the computational times remain larger than for the λ-modes. The matrices associated with the γ-modes are well-conditioned. Nevertheless, for the same configuration their eigenvalue spectrum is more clustered than the one obtained for λ and α-modes. Thus, in general, the convergence of the eigensolvers is slower than the convergence of the λ and the α-modes. Therefore, λ-modes can be computed faster than the rest of the spatial modes. Near of reactor criticality, the eigenfunctions associated with the different spatial modes are similar. Thus, the λ-modes can be used as initial guess for the eigensolvers to compute the γ and α-modes. This initialization strategy, together with one of the two generalizations of the modified block Newton method proposed in this thesis, has been shown to be more efficient than computing the γ and α-modes directly.

Second, in this Chapter, two block methods to solve the λ-modes problem obtained from the discretization of the neutron diffusion equation have been studied. Both methods have been defined for generalized eigenvalue problems. Moreover, a block multilevel technique has been proposed to initialize both methods that improves the convergence rate of the methods with hardly any overrun. The first method is the block inverse free preconditioned Arnoldi method (BIFPAM), where the efficiency of using a preconditioner has been studied. In particular, the ILU(0) preconditioner and a geometrical multigrid preconditioner (GMG) have been used obtaining similar computional times for dimensions in the Krylov subspace of 8 and 4, respectively. The second block method has been the modified generalized block Newton method. This method

does not require to solve many linear systems, but these systems need to be previously preconditioned. Different block preconditioners have been studied as an alternative to assemble the full matrix and construct a preconditioner in each iteration. The preconditioners proposed in this work break down the setup cost at the price of a slight increase of the number of iterations. The result is a significant reduction of the total CPU time needed to reach the convergence and the memory occupancy. Moreover, a hybrid scheme is proposed combining these previous methods that improves the robustness and the computational times. Numerical tests indicate too that the block hybrid method is more efficient than other competitive methods as the Krylov-Schur method and the Generalized Davidson method. Moreover, it is more efficient in the computation of one eigenvalue that the power iteration method, very used in nuclear computations.

The block structure of the matrices permits different strategies of the allocations in the matrices as well as to introduce a matrix-free implementation in the code. A good option for two energy groups diffusion computations is the *non-diagonal* strategy that only saves the diagonal blocks of the matrices reducing enormously the CPU time, removing the time to assemble the full matrices and improving the velocity of matrix-vector computations for degrees in the FEM higher than 2. The differences increase when the size of the problem is larger. For this type of implementation the Block Gauss-Seidel is used (instead the ILU(0)). A *full matrix-free* allocation for diffusion computations is not efficient because, at the moment, the matrix-free preconditioners are not competitive for medium size problems.

To end this Chapter, the eigenvalue solvers are tested to compute the λ-modes associated with the SP_N equations for the C5G7 benchmark in the two-dimensional and three-dimensional version. First, an initialization based on using the solution of the SP_N problem with $N = 1$ and linear shape functions in FEM (*Multilevel-fem-spn*) is proposed. It only takes a small percentage of the total CPU time to solve the problem and improves considerably the convergence of the block methods. The number of energy groups in the numerical problems is seven. Thus, the *non-diagonal* matrix allocation is used to compare the eigenvalue solvers. The preconditioner used in the numerical computations has been the Block Gauss-Seidel. The block inverse-free preconditioner Arnoldi method (BIFPAM) has been the most efficient option compared with the standard power iteration method, the Krylov-Schur method, the generalized Davidson method, the modified generalized block Newton method (MGBNM) and the hybrid method. In this case, the block Gauss-Seidel preconditioner separates the spectrum of matrices better than for diffusion calculations. Thus, the rate of convergence of the BIFPAM is higher than for the MGBNM. The computational efficiency

improves as the problem is higher. Furthermore, the *full matrix-free* implementation is used to compute the λ-modes associated with the SP_N equations with the BIFPAM method and the Block Gauss-Seidel preconditioner with the conjugate gradient and the multilevel-fem-spn preconditioner in their diagonal blocks. Numerical results show for all cases a huge reduction of the computational memory in the running. The multilevel-fem-spn preconditioner is not as efficient as the ILU(0) preconditioner. However, this strategy reduces the CPU times for three-dimensional problems with cubic polynomials in the FEM.

Chapter 5 presents different modal kinetic methods to integrate the time dependent neutron diffusion equation. In particular, the λ, the γ and the α-modes problem to develop this methodology. From the obtained results, we can highlight the following conclusions. A modal kinetics with more than one eigenvalue is necessary to describe some types of transient such as the out-of-phase oscillations or the local perturbations. For symmetric control bars movements, only one eigenvalue is required in the expansions. However, these modal kinetics give more accurate results when the number of eigenvalues considered is larger. The largest differences in the power evolution between the backward differential method BKM solution and the modal approximations solutions are mainly when the spatial distribution of the power is more different from the power distribution of the reactor in steady state. The CPU times obtained with the BKM and the CPU times obtained with the different modal expansions show that the modal methodology is a faster strategy to obtain the solution in the time-dependent problems analyzed. Regarding the different spatial modes used in the modal expansions, we have not observed meaningful differences between the results in terms of the total power evolution of the transients studied. However, there are some differences in the stiffness of the resulting dynamical systems associated with the modal kinetics for each kind of mode. The differential systems associated with the α-modes and the γ-modes are not as stiff as the systems obtained using the λ-modes. Nevertheless, as the λ-modes are cheaper to be computed than the rest of the modes, the λ modal kinetics gives the approximations in less time.

To obtain better approximations without using a high number of modes the updated modal method is developed where the eigenfunctions have been updated along the transient. Small steps of time are necessary to ensure good accuracy in the approximations. There are not high differences between the type of mode used. The smallest CPU times are also obtained with the λ-modes. To avoid the previous selection of the fixed time-step, an adaptive control time-step has been proposed. This control is adapted depending on the error in the modal functions. Different error estimations and type of controls are defined for the modal methodology. Better results are obtained with an error estimation based

on the residual error on the generalized eigenvalue problem. With respect to the type of control, the dynamic control error is adapted more to the local errors. The results show that the adaptive updated modal method decreases the errors with the same CPU times than the updated modal method with fixed time-step, but also it obtains more distributed local errors over the time.

6.1 Future work

As future work of this thesis, the following ideas can be developed.

- In spite of the computation of the α and γ-modes is more expensive than for the λ-modes, they are shown very useful in many neutron applications. In future, the definition and study of the α and γ-modes for the SP_N equations will be proposed.
- For the SP_N equations, the block Gauss-Seidel preconditioner to solve the linear systems in the MGBNM is not as efficient as for diffusion computations. Thus, future works are being devoted to design alternatives block preconditioners.
- The matrix-free implementation reduces the CPU memory and for high degrees of the polynomial in FEM reduces also the CPU times. Other works have proved that the efficiency of this technique improves with more than one processor. Thus, the parallelization of the code with GPUs is being developed. This technique, mainly with the full matrix-free implementation, requires efficient preconditioners that are not based on the factorization of the matrices. The implementation of the *multilevel-fem-spn* preconditioner will be improved.
- The backward differential method needs to solve many linear systems. For this reason, the CPU time to obtain an approximate solution for transients is very high. A future implementation of the code will introduce a methodology based on low-rank updates of preconditioners for sequences of linear systems.
- The modal methodology has been shown to be an efficient methodology to integrate the neutron diffusion equation. Future works, we will extend this study to integrate the SP_N equations (or other approximations of the neutron transport equations) in order to test benchmark problems such as the time-dependent C5G7-TD.

- In realistic transient reactors, the neutronic computations are highly dependent on thermal-hydraulic variables. Future works will be devoted for the coupling of the code developed in this thesis with a thermal-hydraulic feedback.

BIBLIOGRAPHY

Abhyankar, S., J. Brown, E. M. Constantinescu, D. Ghosh, B.F. Smith, and H. Zhang (2018). "PETSc/TS: A Modern Scalable ODE/DAE Solver Library". In: *arXiv preprint arXiv:1806.01437* (cit. on pp. 133, 134).

Adams, M., M. Brezina, J. Hu, and R. Tuminaro (2003). "Parallel multigrid smoothing: polynomial versus Gauss–Seidel". In: *Journal of Computational Physics* 188.2, pp. 593–610 (cit. on p. 94).

Adrover, A., F. Creta, M. Giona, and M. Valorani (2005). "Biorthogonalization, geometric invariant properties and rate-based estimate of Lyapunov spectra". In: *Physics Letters A* 342.5, pp. 421–429 (cit. on pp. 79, 133).

Akcasuh, Z. (1971). *Mathematical methods in nuclear reactor dynamics*. Elsevier (cit. on p. 124).

Allen, E.J. and R.M. Berry (2002). "The inverse power method for calculation of multiplication factors". In: *Annals of Nuclear Energy* 29.8, pp. 929–935 (cit. on p. 63).

Avvakumov, A. V., V. F. Strizhov, P. N. Vabishchevich, and AO Vasilev (2017a). "Modelling dynamic processes in a nuclear reactor by state change modal method". In: *Journal of Physics: Conference Series*. Vol. 937. IOP Publishing, p. 012003 (cit. on p. 124).

Avvakumov, A. V, V. F Strizhov, P. N. Vabishchevich, and A. O. Vasilev (2017b). "Spectral properties of dynamic processes in a nuclear reactor". In: *Annals of Nuclear Energy* 99, pp. 68–79 (cit. on p. 17).

Avvakumov, A. V, V. F Strizhov, P. N. Vabishchevich, and A. O. Vasilev (2018a). "Automatic Time Step Selection for Numerical Solution of Neutron Diffusion Problems". In: *International Conference on Finite Difference Methods*. Springer, pp. 145–152 (cit. on p. 125).

Avvakumov, A.V., V.F. Strizhov, P. Vabishchevich, and A. Vasilev (2018b). "State change modal method for numerical simulation of dynamic processes in a nuclear reactor". In: *Progress in Nuclear Energy* 106, pp. 240–261 (cit. on pp. 17, 124).

Baker, C. G., U.L. Hetmaniuk, R.B. Lehoucq, and H. K. Thornquist (2009). "Anasazi software for the numerical solution of large-scale eigenvalue problems". In: *ACM Transactions on Mathematical Software (TOMS)* 36.3, p. 13 (cit. on p. 72).

Bangerth, W., R. Hartmann, and G. Kanschat (2007). "deal.II – a General Purpose Object Oriented Finite Element Library". In: *ACM Trans. Math. Softw.* 33.4, pp. 24/1–24/27 (cit. on pp. 38, 134).

Bastian, P., M. Blatt, and R. Scheichl (2012). "Algebraic multigrid for discontinuous Galerkin discretizations of heterogeneous elliptic problems". In: *Numerical Linear Algebra with Applications* 19.2, pp. 367–388 (cit. on p. 95).

Bastian, P., E.H. Müller, S. Müthing, and M. Piatkowski (2019). "Matrix-free multigrid block-preconditioners for higher order Discontinuous Galerkin discretisations". In: *Journal of Computational Physics* (cit. on p. 94).

Bell, G.I. and S. Glasstone (1970). *Nuclear Reactor Theory*. New York: Litton Educational Publishing, INC (cit. on pp. 17, 18).

Bernal A.and Hébert, A., J.E. Roman, R. Miró, and G. Verdú (2017). "A Krylov–Schur solution of the eigenvalue problem for the neutron diffusion equation discretized with the Raviart–Thomas method". In: *Journal of Nuclear Science and Technology* 54.10, pp. 1085–1094 (cit. on pp. 64, 69).

Bernal García, A. (2018). "Development of a 3D Modal Neutron Code with the Finite Volume Method for the Diffusion and Discrete Ordinates Transport Equations. Application to Nuclear Safety Analyses". PhD thesis (cit. on pp. 38, 124).

Boffie, J. and J. M. Pounders (2018). "An adaptive time step control scheme for the transient diffusion equation". In: *Annals of Nuclear Energy* 116, pp. 280–289 (cit. on p. 125).

Boudjemai, A., R. Amri, A. Mankour, H. Salem, M.H. Bouanane, and D. Boutchicha (2012). "Modal analysis and testing of hexagonal honeycomb plates used for satellite structural design". In: *Materials & Design* 35, pp. 266–275 (cit. on p. 2).

Brantley, P.S. and E.W. Larsen (2000). "The simplified P3 approximation". In: *Nuclear Science and Engineering* 134.1, pp. 1–21 (cit. on p. 33).

Brownjohn, J.M.W., F. Magalhaes, E. Caetano, and A. Cunha (2010). "Ambient vibration re-testing and operational modal analysis of the Humber Bridge". In: *Engineering Structures* 32.8, pp. 2003–2018 (cit. on p. 2).

Cai, Y., X. Peng, Q. Li, K. Wang, X. Qin, and R. Guo (2019). "The numerical solution of space-dependent neutron kinetics equations in hexagonal-z geometry using backward differentiation formula with adaptive step size". In: *Annals of Nuclear Energy* 128, pp. 203–208 (cit. on p. 125).

Capilla, M., C. Talavera, D. Ginestar, and G. Verdú (2005). "A nodal collocation method for the calculation of the lambda modes of the PL equations". In: *Annals of Nuclear Energy* 32.17, pp. 1825–1853 (cit. on p. 27).

Capilla, M., C.F. Talavera, D. Ginestar, and G. Verdú (2008). "A nodal collocation approximation for the multi-dimensional PL equations–2D applications". In: *Annals of Nuclear Energy* 35.10, pp. 1820–1830 (cit. on p. 37).

Carney, S., F. Brown, B. Kiedrowski, and W. Martin (2014). "Theory and applications of the fission matrix method for continuous-energy Monte Carlo". In: *Annals of Nuclear Energy* 73, pp. 423–431 (cit. on p. 66).

Caron, D., S. Dulla, and P. Ravetto (2017). "Adaptive time step selection in the quasi-static methods of nuclear reactor dynamics". In: *Annals of Nuclear Energy* 105, pp. 266–281 (cit. on p. 125).

Carreño, A., A. Vidal-Ferrándiz, D. Ginestar, and G. Verdú (2017a). "Multilevel method to compute the lambda modes of the neutron diffusion equation". In: *Applied Mathematics and Nonlinear Sciences* 2.1, pp. 225–236 (cit. on pp. 65, 89).

Carreño, A., A. Vidal-Ferràndiz, D. Ginestar, and G. Verdú (2017b). "Spatial modes for the neutron diffusion equation and their computation". In: *Annals of Nuclear Energy* 110, pp. 1010–1022 (cit. on pp. 18, 39, 65, 76).

Carreño, A., A. Vidal-Ferràndiz, D. Ginestar, and G. Verdú (2018a). "Block hybrid multilevel method to compute the dominant λ-modes of the neutron

diffusion equation". In: *Annals of Nuclear Energy* 121, pp. 513–524 (cit. on pp. 65, 73, 81, 96).

Carreño, A., A. Vidal-Ferràndiz, D. Ginestar, and G. Verdú (2018b). "The Solution of the Lambda Modes Problem Using Block Iterative Eigensolvers". In: *International Conference on Computational Science*. Springer, pp. 846–855 (cit. on pp. 65, 73).

Carreño, A., A. Vidal-Ferràndiz, D. Ginestar, and G. Verdú (2019a). "A Matrix-Free Eigenvalue Solver for the Multigroup Neutron Diffusion Equation". In: *International Conference on Computational Science*. Springer, pp. 702–709 (cit. on pp. 65, 73, 104).

Carreño, A., L. Bergamaschi, A. Martinez, A. Vidal-Ferrándiz, D. Ginestar, and G. Verdú (2019b). "Block Preconditioning Matrices for the Newton Method to Compute the Dominant λ-Modes Associated with the Neutron Diffusion Equation". In: *Mathematical and Computational Applications* 24.1, p. 9 (cit. on pp. 65, 97).

Carreño, A., A. Vidal-Ferràndiz, D. Ginestar, and G. Verdú (2019c). "Modal methods for the neutron diffusion equation using different spatial modes". In: *Progress in Nuclear Energy* 115, pp. 181–193 (cit. on pp. 18, 65, 125).

Chaigne, A. and J. Kergomard (2016). *Acoustics of musical instruments*. Springer (cit. on p. 2).

Cornejo, L.R., D. Y Anistratov, and K. Smith (2019). "Iteration Methods with Multigrid in Energy for Eigenvalue Neutron Diffusion Problems". In: *Nuclear Science and Engineering*, pp. 1–25 (cit. on p. 94).

Davidson, E.R. (1975). "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices". In: *Journal of Computational Physics* 17.1, pp. 87–94 (cit. on p. 72).

Demazière, C. (2019). *Modelling of Nuclear Reactor Multi-physics: From Local Balance Equations to Macroscopic Models in Neutronics and Thermal-Hydraulics*. Academic Press (cit. on p. 9).

Dulla, S., E. H. Mund, and P. Ravetto (2008). "The quasi-static method revisited". In: *Progress in Nuclear Energy* 50.8, pp. 908–920 (cit. on p. 124).

Dulla, S., P. Ravetto, and P. Saracco (2018). "The time eigenvalue spectrum for nuclear reactors in multi-group diffusion theory". In: *The European Physical Journal Plus* 133.9, p. 390 (cit. on p. 124).

Finnemann, H. (1975). *A consistent nodal method for the analysis of space-time effects in large LWR's*. Tech. rep. Technische Univ. Muenchen, Garching (F.R. Germany). Lab. fuer Reaktorregelung und Anlagensicherung (cit. on p. 37).

Finnemann, H. and A. Galati (1991). "NEACRP 3-D LWR core transient benchmark, final specification". In: *NEACRPL-335 (Revision 1)* (cit. on pp. 53, 201).

Freeman, T. and K. Weilnau (2017). *Golf Cart Noise and Vibration Troubleshooting*. Tech. rep. SAE Technical Paper (cit. on p. 2).

Fu, Z.F. and J. He (2001). *Modal analysis*. Elsevier (cit. on p. 1).

Gelbard, E.M. (1960). "Application of spherical harmonics method to reactor problems". In: *Bettis Atomic Power Laboratory, West Mifflin, PA, Technical Report No. WAPD-BT-20* (cit. on pp. 8, 26).

Gill, D.F. and Y.Y. Azmy (2009). "A Jacobian-Free Newton-Krylov iterative scheme for criticality calculations based on the neutron diffusion equation". In: *International Conference on Mathematics, Computational Methods, and Reactor Physics, Saratoga Springs, NY, United States* (cit. on p. 65).

Gill, D.F. and Y.Y. Azmy (2011). "Newton's method for solving k-eigenvalue problems in neutron diffusion theory". In: *Nuclear Science and Engineering* 167.2, pp. 141–153 (cit. on p. 65).

Ginestar, D., G. Verdú, V. Vidal, R. Bru, J. Marín, and J. L. Munoz-Cobo (1998). "High order backward discretization of the neutron diffusion equation". In: *Annals of Nuclear Energy* 25.1-3, pp. 47–64 (cit. on pp. 124, 127).

Golub, G. and Q. Ye (2002). "An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems". In: *SIAM Journal on Scientific Computing* 24.1, pp. 312–334 (cit. on pp. 64, 73, 75).

Golub, G.H. and J.H. Welsch (1969). "Calculation of Gauss quadrature rules". In: *Mathematics of computation* 23.106, pp. 221–230 (cit. on p. 41).

González-Pintor, S., D. Ginestar, and G. Verdú (2009). "High order finite element method for the lambda modes problem on hexagonal geometry". In: *Annals of Nuclear Energy* 36.9, pp. 1450–1462 (cit. on p. 38).

González-Pintor, S., D. Ginestar, and G. Verdú (2011). "Updating the Lambda Modes of a nuclear power reactor". In: *Mathematical and Computer Modelling* 54.7, pp. 1796–1801 (cit. on p. 64).

Gustafsson, K., M. Lundh, and G. Söderlind (1988). "API stepsize control for the numerical solution of ordinary differential equations". In: *BIT Numerical Mathematics* 28.2, pp. 270–287 (cit. on p. 125).

Hackbusch, W. (2013). *Multi-grid methods and applications*. Vol. 4. Springer Science & Business Media (cit. on p. 89).

Hageman, L.A. and D.M. Young (2012). *Applied iterative methods*. Courier Corporation (cit. on p. 63).

Hamilton, S.P. and T.M. Evans (2015). "Efficient solution of the simplified PN equations". In: *Journal of Computational Physics* 284, pp. 155–170 (cit. on pp. 27, 38, 64).

Hauck, C. and R. McClarren (2010). "Positive Pn closures". In: *SIAM Journal on Scientific Computing* 32.5, pp. 2603–2626 (cit. on p. 27).

Hébert, A. (1987). "Development of the nodal collocation method for solving the neutron diffusion equation". In: *Annals of Nuclear Energy* 14.10, pp. 527–541 (cit. on p. 37).

Hébert, A. (2006). "The search for superconvergence in spherical harmonics approximations". In: *Nuclear science and engineering* 154.2, pp. 134–173 (cit. on p. 26).

Hebert, A. (2008). "A Raviart–Thomas–Schneider solution of the diffusion equation in hexagonal geometry". In: *Annals of Nuclear Energy* 35.3, pp. 363–376 (cit. on p. 38).

Hébert, A. (2009). *Applied reactor physics*. Presses inter Polytechnique (cit. on pp. 11, 37).

Henry, A. F. (1958). "The application of reactor kinetics to the analysis of experiments". In: *Nuclear Science and Engineering* 3.1, pp. 52–70 (cit. on p. 124).

Henry, A.F. (1975). *Nuclear-reactor analysis*. Vol. 4. MIT press Cambridge, Massachusetts (cit. on pp. 11, 17, 24).

Hernandez, V., J.E. Roman, and Vidal V. (2005). "SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems". In: *ACM Trans. Math. Software* 31.3, pp. 351–362 (cit. on pp. 66, 72).

Hindmarsh, A.C., P.N. Brown, K.E. Grant, S. L. Lee, R. Serban, D.E. Shumaker, and C.S. Woodward (2005). "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers". In: *ACM Transactions on Mathematical Software (TOMS)* 31.3, pp. 363–396 (cit. on p. 133).

Kerschen, G., M. Peeters, J.C. Golinval, and C. Stéphan (2013). "Nonlinear modal analysis of a full-scale aircraft". In: *Journal of Aircraft* 50.5, pp. 1409–1419 (cit. on p. 2).

Klose, A.D. and E.W. Larsen (2006). "Light transport in biological tissue based on the simplified spherical harmonics equations". In: *Journal of Computational Physics* 220.1, pp. 441–470 (cit. on p. 8).

Knoll, D.A., H. Park, and C. Newman (2011). "Acceleration of k-eigenvalue/criticality calculations using the Jacobian-free Newton-Krylov method". In: *Nuclear Science and Engineering* 167.2, pp. 133–140 (cit. on p. 65).

Knyazev, A.V. and K. Neymeyr (2003). "A geometric theory for preconditioned inverse iteration III: A short and sharp convergence estimate for generalized eigenvalue problems". In: *Linear Algebra and its Applications* 358.1-3, pp. 95–114 (cit. on p. 96).

Komatitsch, D., G. Erlebacher, D. Göddeke, and D. Michéa (2010). "High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster". In: *Journal of computational physics* 229.20, pp. 7692–7714 (cit. on p. 103).

Kópházi, J. and D. Lathouwers (2012). "Three-dimensional transport calculation of multiple alpha modes in subcritical systems". In: *Annals of Nuclear Energy* 50, pp. 167–174 (cit. on p. 18).

Kronbichler, M. and K. Kormann (2012). "A generic interface for parallel cell-based finite element operator application". In: *Computers & Fluids* 63, pp. 135–147 (cit. on p. 104).

Kronbichler, M. and W.A. Wall (2018). "A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers". In: *SIAM Journal on Scientific Computing* 40.5, A3423–A3448 (cit. on p. 94).

Lamarsh, J.R. and A. J. Baratta (2001). *Introduction to nuclear engineering*. Vol. 3. Prentice hall Upper Saddle River, NJ (cit. on pp. 8, 9).

Lange, C., D. Hennig, M. Schulze, and A. Hurtado (2014). "Complex BWR dynamics from the bifurcation theory point of view". In: *Annals of Nuclear Energy* 67, pp. 91–108 (cit. on p. 124).

Langenbuch, S., W. Maurer, and W. Werner (1977). "Coarse-mesh flux-expansion method for the analysis of space-time effects in large light water reactor cores". In: *Nuclear Science and Engineering* 63.4, pp. 437–456 (cit. on pp. 50, 133, 138, 198).

Larsen, E.W., J.E. Morel, and J.M. McGhee (1996). "Asymptotic derivation of the multigroup P 1 and simplified PN equations with anisotropic scattering". In: *Nuclear science and engineering* 123.3, pp. 328–342 (cit. on p. 8).

Lefvert, T. (1996). "Ringhals 1 stability benchmark". In: *NEA Nuclear Science Committee Report EA/NSC/DOC (96)* 22 (cit. on p. 107).

Lehoucq, R.B. (2001). "Implicitly restarted Arnoldi methods and subspace iteration". In: *SIAM Journal on Matrix Analysis and Applications* 23.2, pp. 551–562 (cit. on p. 71).

Lewins, J. (2013). *Nuclear reactor kinetics and control*. Elsevier (cit. on p. 18).

Lewis, E.E. and W.F. Miller (1984). *Computational methods of neutron transport* (cit. on pp. 12, 14, 18, 20).

Lewis, E.E., M.A. Smith, N. Tsoulfanidis, G. Palmiotti, T.A. Taiwo, and R.N. Blomquist (2001). "Benchmark specification for Deterministic 2-D/3-D MOX fuel assembly transport calculations without spatial homogenization (C5G7 MOX)". In: *NEA/NSC* 280 (cit. on pp. 57, 120, 203).

Li, P., C. Liu, Q. Tian, H. Hu, and Y. Song (2016). "Dynamics of a deployable mesh reflector of satellite antenna: form-finding and modal analysis". In: *Journal of Computational and Nonlinear Dynamics* 11.4, p. 041017 (cit. on p. 2).

Lima, Z. de, F. O. da Silva, and A. Alvim (2009). "A modal multidimensional kinetics method using pseudo-harmonics". In: *Annals of Nuclear Energy* 36.6, pp. 752–759 (cit. on p. 124).

Lösche, R., H. Schwetlick, and G. Timmermann (1998). "A modified block Newton iteration for approximating an invariant subspace of a symmetric matrix". In: *Linear algebra and its applications* 275, pp. 381–400 (cit. on pp. 64, 76).

Mahadevan, V. and J. Ragusa (2008). "Novel hybrid scheme to compute several dominant eigenmodes for reactor analysis problems". In: *International Conference on the Physics of Reactors, Interlaken, Switzerland, September 14-19* (cit. on p. 65).

March-Leuba, J. and E.D. Blakeman (1991). "A mechanism for out-of-phase power instabilities in boiling water reactors". In: *Nuclear Science and Engineering* 107.2, pp. 173–179 (cit. on pp. 17, 124).

March-Leuba, J. and J. Rey (1993). "Coupled thermohydraulic-neutronic instabilities in boiling water nuclear reactors: a review of the state of the art". In: *Nuclear Engineering and Design* 145.1-2, pp. 97–111 (cit. on p. 124).

McCormick, S.F. (1987). *Multigrid methods*. SIAM (cit. on p. 89).

Miró, R., D. Ginestar, D. Hennig, and G. Verdu (2000). "On the regional oscillation phenomenon in BWR's". In: *Progress in Nuclear Energy* 36.2, pp. 189–229 (cit. on pp. 3, 17, 63).

Miró, R., D. Ginestar, G. Verdú, and D. Hennig (2002). "A nodal modal method for the neutron diffusion equation. Application to BWR instabilities analy-

sis". In: *Annals of Nuclear Energy* 29.10, pp. 1171–1194 (cit. on pp. 3, 17, 124, 148).

Morgan, R.B. and D.S. Scott (1986). "Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices". In: *SIAM Journal on Scientific and Statistical Computing* 7.3, pp. 817–825 (cit. on p. 73).

Olsen, J., P. Jørgensen, and J. Simons (1990). "Passing the one-billion limit in full configuration-interaction (FCI) calculations". In: *Chemical Physics Letters* 169.6, pp. 463–472 (cit. on p. 73).

Ortega-Moñux, A., J.G. Wanguemert-Perez, I. Molina-Fernandez, E. Silvestre, and P. Andrés (2006). "Enhanced accuracy in fast-Fourier-based methods for full-vector modal analysis of dielectric waveguides". In: *IEEE photonics technology letters* 18.10, pp. 1128–1130 (cit. on p. 2).

Otuka, N., E. Dupont, V. Semkova, B. Pritychenko, A.I. Blokhin, M. Aikawa, S. Babykina, M. Bossant, G. Chen, S. Dunaeva, et al. (2014). "Towards a more complete and accurate experimental nuclear reaction data library (EXFOR): international collaboration between nuclear reaction data centres (NRDC)". In: *Nuclear Data Sheets* 120, pp. 272–276 (cit. on p. 11).

Panza, M.A. (2015). "A review of experimental techniques for NVH analysis on a commercial vehicle". In: *Energy Procedia* 82, pp. 1017–1023 (cit. on p. 2).

Pioldi, F., J. Salvi, and E. Rizzi (2017). "Refined FDD modal dynamic identification from earthquake responses with Soil-Structure Interaction". In: *International Journal of Mechanical Sciences* 127, pp. 47–61 (cit. on p. 2).

Quillen, P. and Q. Ye (2010). "A block inverse-free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems". In: *Journal of Computational and Applied Mathematics* 233.5, pp. 1298–1313 (cit. on pp. 64, 74).

Ronen, Y., D. Shvarts, and JJ. Wagschal (1976). "A Comparison of Some Eigenvalues in Reactor Theory". In: *Nuclear Science and Engineering* 60.1, pp. 97–101 (cit. on pp. 17, 18, 54).

Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM (cit. on pp. 68, 69, 71, 93, 94, 104).

Saad, Y. and M.H. Schultz (1986). "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM Journal on scientific and statistical computing* 7.3, pp. 856–869 (cit. on p. 93).

Sampath, R.S. and G. Biros (2010). "A parallel geometric multigrid method for finite elements on octree meshes". In: *SIAM Journal on Scientific Computing* 32.3, pp. 1361–1392 (cit. on p. 89).

Sanchez, R. (2012). "On SN-PN Equivalence". In: *Transport Theory and Statistical Physics* 41.5-6, pp. 418–447 (cit. on p. 8).

Shim, C., Y. Jung, J. Yoon, and H. Joo (2011). "Application of backward differentiation formula to spatial reactor kinetics calculation with adaptive time step control". In: *Nuclear Engineering and Technology* 43.6, pp. 531–546 (cit. on p. 125).

Silvestre, E., T. Pinheiro-Ortega, P. Andrés, J.J. Miret, and A. Ortigosa-Blanch (2005). "Analytical evaluation of chromatic dispersion in photonic crystal fibers". In: *Optics letters* 30.5, pp. 453–455 (cit. on p. 2).

Singh, T., T. Mazumdar, and P. Pandey (2014). "NEMSQR: A 3-D multi group diffusion theory code based on nodal expansion method for square geometry". In: *Annals of Nuclear Energy* 64, pp. 230 –243. ISSN: 0306-4549 (cit. on p. 37).

Sleijpen, G.L., A.G. Booten, D.R. Fokkema, and H.A. Van der Vorst (1996). "Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems". In: *BIT Numerical Mathematics* 36.3, pp. 595–633 (cit. on p. 73).

Smith, K. (1979). "An analytic nodal method for solving the two-group, multidimensional, static and transient neutron diffusion equations". PhD thesis. Dept. of Nuclear Engineering, Massachusetts Institute of Technology (cit. on p. 37).

Smith, M.A., E.E. Lewis, and B.C. Na (2003). *Benchmark on deterministic transport calculations without spatial homogenisation – A 2-D/3-D MOX Fuel Assembly Benchmark (C5G7 MOX Benchmark).* Tech. rep. NEA/NSC/DOC(2003)16. OECD/NEA (cit. on pp. 57, 60).

Smith, M.A., E. E. Lewis, and B. Na (2006). "Benchmark on deterministic 3-D MOX fuel assembly transport calculations without spatial homogenization". In: *Progress in Nuclear Energy* 48.5, pp. 383–393 (cit. on p. 203).

Söderlind, G. (2002). "Automatic control and adaptive time-stepping". In: *Numerical Algorithms* 31.1-4, pp. 281–310 (cit. on p. 125).

Stacey, W. M. (1969). *Space-time nuclear reactor kinetics*. Vol. 5. Academic Press (cit. on pp. 21, 123, 124, 126).

Stacey, W.M. (2018). *Nuclear reactor physics*. John Wiley & Sons (cit. on pp. 10, 22, 30, 69).

Stewart, G.W. (2002). "A Krylov–Schur algorithm for large eigenproblems". In: *SIAM Journal on Matrix Analysis and Applications* 23.3, pp. 601–614 (cit. on p. 71).

Sutton, T.M. (1988). "Wielandt iteration as applied to the nodal expansion method". In: *Nuclear Science and Engineering* 98.2, pp. 169–173 (cit. on p. 63).

Theler, G. (2013). "Unstructured grids and the multigroup neutron diffusion equation". In: *Science and Technology of Nuclear Installations* 2013 (cit. on p. 38).

Tommasi, J., M. Maillot, and Gérald Rimpault (2016). "Calculation of Higher-Order Fluxes in Symmetric Cores-I: Theory". In: *Nuclear Science and Engineering* 184.2, pp. 174–189 (cit. on pp. 50, 133).

Turcksin, B., J.C. Ragusa, and W. Bangerth (2010). "Goal-oriented h-adaptivity for the multigroup SPN equations". In: *Nuclear Science and Engineering* 165.3, pp. 305–319 (cit. on p. 38).

Universitat Politècnica de València. *Rigel Cluster Description*. URL: `https://wiki.upv.es/confluence/pages/viewpage.action?pageId=264044546` (cit. on p. 120).

Uyttenhove, W., D. Lathouwers, J.L. Kloosterman, T. van der Hagen, G. Van den Eynde, and P. Baeten (2014). "Methodology for modal analysis at pulsed

neutron source experiments in accelerator-driven systems". In: *Annals of Nuclear Energy* 72, pp. 286–297 (cit. on p. 18).

Velarde, G., C. Ahnert, and J.M. Aragones (1978). "Analysis of the eigenvalue equations in k, lambda, gamma, and alpha applied to some fast and thermal neutron systems". In: *Nucl. Sci. Eng.* 66:3 (cit. on p. 17).

Verdú, G. and D. Ginestar (2014). "Modal decomposition method for BWR stability analysis using Alpha-modes". In: *Annals of Nuclear Energy* 67, pp. 31–40 (cit. on p. 124).

Verdú, G., D. Ginestar, V. Vidal, and J.L. Muñoz-Cobo (1994). "3D λ-modes of the neutron-diffusion equation". In: *Annals of Nuclear Energy* 21.7, pp. 405–421 (cit. on pp. 17, 37, 63).

Verdú, G., D. Ginestar, V. Vidal, and R. Miró (1998). "Modal decomposition method for BWR stability analysis". In: *Journal of nuclear science and technology* 35.8, pp. 538–546 (cit. on pp. 17, 124).

Verdú, G., R. Miró, D. Ginestar, and V. Vidal (1999). "The implicit restarted Arnoldi method, an efficient alternative to solve the neutron diffusion equation". In: *Annals of nuclear energy* 26.7, pp. 579–593 (cit. on p. 63).

Verdú, G., D. Ginestar, R. Miró, and V. Vidal (2005). "Using the Jacobi–Davidson method to obtain the dominant Lambda modes of a nuclear power reactor". In: *Annals of nuclear energy* 32.11, pp. 1274–1296 (cit. on p. 64).

Verdú, G., D. Ginestar, J.E. Roman, and V. Vidal (2010). "3D alpha modes of a nuclear power reactor". In: *Journal of nuclear science and technology* 47.5, pp. 501–514 (cit. on pp. 18, 24, 25, 37).

Vidal, V., G. Verdú, D. Ginestar, and J.L. Muñoz-Cobo (1998). "Variational acceleration for subspace iteration method. Application to nuclear power reactors". In: *International journal for numerical methods in engineering* 41.3, pp. 391–407 (cit. on p. 63).

Vidal Ferràndiz, A. (2018). "Development of a finite element method for neutron transport equation approximations". PhD thesis (cit. on pp. 11, 38).

Vidal-Ferrandiz, A., R. Fayez, D. Ginestar, and G. Verdú (2014). "Solution of the Lambda modes problem of a nuclear power reactor using an h–p finite element method". In: *Annals of Nuclear Energy* 72, pp. 338–349 (cit. on pp. 38, 63).

Vidal-Ferràndiz, A., R. Fayez, D. Ginestar, and G. Verdú (2016). "Moving meshes to solve the time-dependent neutron diffusion equation in hexagonal geometry". In: *Journal of Computational and Applied Mathematics* 291, pp. 197–208 (cit. on p. 127).

Vidal-Ferràndiz, A., A. Carreño, D. Ginestar, and G. Verdú (2019). "A block Arnoldi method for the SPN equations". In: *International Journal of Computer Mathematics*, pp. 1–17 (cit. on pp. 39, 65, 73, 104).

Vidal-Ferràndiz, A. and Ginestar, D. and Verdú, G. *`FEMFFUSION`: A finite element method code for the neutron diffusion equation.* URL: `https://bitbucket.org/Zonni/femffusion` (cit. on p. 38).

Wanner, G. and E. Hairer (1996). *Solving ordinary differential equations II*. Springer Berlin Heidelberg (cit. on pp. 125, 160).

Warsa, J.S., T.A. Wareing, J.E. Morel, J.M. McGhee, and R.B. Lehoucq (2004). "Krylov subspace iterations for deterministic k-eigenvalue calculations". In: *Nuclear Science and Engineering* 147.1, pp. 26–42 (cit. on p. 63).

Weinberg, A.M. and E.P. Wigner (1958). "The physical theory of neutron chain reactors". In: (cit. on p. 10).

Wexler, A. (1967). "Solution of waveguide discontinuities by modal analysis". In: *IEEE Transactions on Microwave Theory and Techniques* 15.9, pp. 508–517 (cit. on p. 2).

Yamamoto, T. and H. Sakamoto (2019). "Decomposition of neutron noise in a reactor into higher-order mode components and investigation of the space and frequency dependence". In: *Progress in Nuclear Energy* 117, p. 103098 (cit. on pp. 3, 18).

Yamamoto, T., H. Endo, T. Yokoyama, and M. Kawashima (2010). "Implementation of Transient Neutron Transport Solver in ASTERIA-FBR". In: *Proceedings of SNA+ MC*, pp. 17–21 (cit. on p. 124).

Zienkiewicz, O.C., R.L. Taylor, P. Nithiarasu, and J.Z. Zhu (1977). *The finite element method*. Vol. 3. McGraw-hill London (cit. on pp. 41, 44).

APPENDIX A

ANALYTICAL SOLUTIONS FOR 3D HOMOGENEOUS REACTORS

A.1 Analytical solution for λ-modes problem

A 3D prismatic homogeneous reactor is considered. The λ-modes problem associated to the two energy groups neutron diffusion equation for a tridimensional domain, $V = [0, L_x] \times [0, L_y] \times [0, L_z]$, is defined as

$$-\vec{\nabla} D_1 \vec{\nabla} \psi_1 + (\Sigma_{a1} + \Sigma_{s12})\psi_1 = \frac{1}{\lambda}(\nu\Sigma_{f1}\psi_1 + \nu\Sigma_{f2}\psi_2), \tag{A.1}$$

$$-\Sigma_{s12}\psi_1 - \vec{\nabla} D_2 \vec{\nabla} \psi_2 + \Sigma_{a2}\psi_2 = 0, \tag{A.2}$$

with the boundary conditions

$$\begin{aligned} &\psi_g(0, y, z) = \psi_g(L_x, y, z) = 0, \\ &\psi_g(x, 0, z) = \psi_g(x, L_y, z) = 0, \qquad g = 1, 2. \\ &\psi_g(x, y, 0) = \psi_g(x, y, L_z) = 0, \end{aligned}$$

Using the variables separation method leads to decompose the solution as

$$\psi_g(x, y, z) = X_g(x)Y_g(y)Z_g(z), \qquad g = 1, 2, \tag{A.3}$$

then, the solutions of thermal group can be obtained imposing that

$$\frac{\mathrm{d}^2X_2}{\mathrm{d}x^2}(x) = -B_{x,m}^2 X_2(x), \quad \frac{\mathrm{d}^2Y_2}{\mathrm{d}y^2}(y) = -B_{y,n}^2 Y_2(y), \quad \frac{\mathrm{d}^2Z_2}{\mathrm{d}z^2}(z) = -B_{z,p}^2 Z_2(z), \tag{A.4}$$

with

$$X_2(0) = X_2(L_1) = Y_2(0) = Y_2(L_2) = Z_2(0) = Z_2(L_3) = 0. \tag{A.5}$$

These functions have the general form,

$$\begin{aligned} X_{2,m}(x) &= C_x \sin\left(B_{x,m}x\right), \qquad m = 1, 2, \ldots \\ Y_{2,n}(y) &= C_y \sin\left(B_{y,n}y\right), \qquad n = 1, 2, \ldots \\ Z_{2,p}(z) &= C_z \sin\left(B_{z,p}z\right), \qquad p = 1, 2, \ldots \end{aligned}$$

where

$$B_{x,m} = \frac{m\pi}{L_x}, \quad B_{y,n} = \frac{n\pi}{L_y}, \quad B_{z,p} = \frac{p\pi}{L_z}. \tag{A.6}$$

Different values of the integer numbers m, n and p correspond to the different eigenvalues and the corresponding eigenfunctions of the reactor. The thermal group eigenfunctions are,

$$\psi_2(x, y, z) = k \sin\left(\frac{m\pi}{L_1}x\right) \sin\left(\frac{n\pi}{L_2}y\right) \sin\left(\frac{p\pi}{L_3}z\right), \tag{A.7}$$

so that

$$\nabla^2\psi_2(x, y, z) = -B_{m,n,p}^2 \psi_2(x, y, z), \tag{A.8}$$

where

$$B_{m,n,p}^2 = B_{x,m}^2 + B_{y,n}^2 + B_{z,p}^2. \tag{A.9}$$

From (A.2), it is obtained

$$\psi_1(x, y, z) = \frac{D_2 B_{m,n,p}^2 + \Sigma_{a2}}{\Sigma_{s12}} \psi_2(x, y, z). \tag{A.10}$$

If the Equation (A.10) is replaced in (A.1) and it is simplified, it is derived that the eigenvalues λ are of the form,

$$\lambda = \frac{\nu\Sigma_{f1}(D_2 B_{m,n,p}^2 + \Sigma_{a2}) + \nu\Sigma_{f2}\Sigma_{s12}}{(D_2 B_{m,n,p}^2 + \Sigma_{a2})(\Sigma_{a1} + \Sigma_{s12} + D_1 B_{m,n,p}^2)}.$$

To obtain the value of the constant k of the fluxes (Equation (A.7)), the following normalization condition is imposed,

$$1 = \frac{1}{V_t}\int_V (\Sigma_{f1}|\psi_1| + \Sigma_{f2}|\psi_2|)\ \mathrm{d}V. \tag{A.11}$$

In this case,

$$1 = \frac{1}{L_x L_y L_z}\left(\Sigma_{f1}\left(\frac{D_2 B^2_{m,n,p} + \Sigma_{a2}}{\Sigma_{s12}}\right) + \Sigma_{f2}\right)\int_V |\psi_2(x,y,z)|\ \mathrm{d}V.$$

It can be proved, that

$$\int_\Omega |\psi_2(x,y,z)|\ \mathrm{d}V = k\frac{8L_xL_yL_z}{\pi^3} \qquad \forall m,n,p = 1,2,\ldots, \tag{A.12}$$

to obtain,

$$k = \frac{\pi^3}{8}\left(\frac{\Sigma_{s12}}{\Sigma_{f1}(D_2 B^2_{m,n,p} + \Sigma_{a2}) + \Sigma_{s12}\Sigma_{f2}}\right).$$

A.2 Analytical solution for γ-modes problem

The γ-modes problem for a tridimensional domain, $V = [0, L_x] \times [0, L_y] \times [0, L_z]$, is defined as,

$$-\vec{\nabla} D_1 \vec{\nabla}\phi_1 + (\Sigma_{a1} + \Sigma_{s12})\phi_1 = \frac{1}{\gamma}(\nu\Sigma_{f1}\phi_1 + \nu\Sigma_{f2}\phi_2), \tag{A.13}$$

$$-\vec{\nabla} D_2 \vec{\nabla}\phi_2 + \Sigma_{a2}\phi_2 = \frac{1}{\gamma}\Sigma_{s12}\phi_1, \tag{A.14}$$

with the boundary conditions

$$\begin{aligned} &\phi_g(0,y,z) = \phi_g(L_x,y,z) = 0, \\ &\phi_g(x,0,z) = \phi_g(x,L_y,z) = 0, \qquad g = 1,2. \\ &\phi_g(x,y,0) = \phi_g(x,y,L_z) = 0, \end{aligned}$$

The solution of γ-modes problem is obtained by using the variables separation method following an analogous process to the one used for the λ-modes problem.

For this problem we get,

$$\phi_2(x,y,z) = k \sin\left(\frac{m\pi}{L_x}x\right)\sin\left(\frac{n\pi}{L_y}y\right)\sin\left(\frac{p\pi}{L_z}z\right), \tag{A.15}$$

$$\phi_1(x,y,z) = \frac{\gamma(B^2_{m,n,p}D_2 + \Sigma_{a2})}{\Sigma_{s12}}\phi_2(x,y,z), \tag{A.16}$$

with $m, n, p \in \mathbb{N}$.

The eigenvalue γ is a solution of the second degree equation

$$\begin{aligned}&\left[D_1 B^2_{m,n,p}(D_2 B^2_{m,n,p} + \Sigma_{a2}) + (\Sigma_{a1} + \Sigma_{s12})(D_2 B^2_{m,n,p} + \Sigma_{a2}\right]\gamma^2 \\ &- \left[\nu\Sigma_{f1}(D_2 B^2_{m,n,p} + \Sigma_{a2})\right]\gamma - \Sigma_{s12}\nu\Sigma_{f2} = 0,\end{aligned}$$

and $B^2_{m,n,p}$ is defined in Subsection A.1. In typical reactors, the two solutions of this equation are real and they are sorted by largest magnitude. Different values of m, n, p correspond to different modes and eigenfunctions of the reactor.

To obtain the value of the constant k of the fluxes, the condition of that the mean power production has to be equal to 1 (Eq. A.11) is imposed to get

$$k = \frac{\pi^3}{8}\left(\frac{\Sigma_{s12}}{\Sigma_{f1}\gamma(D_2 B^2_{m,n,p} + \Sigma_{a2}) + \Sigma_{s12}\Sigma_{f2}}\right). \tag{A.17}$$

A.3 Analytical solution for α-modes problem

The α-modes problem for a three-dimensional domain, $V = [0, L_x] \times [0, L_y] \times [0, L_z]$, is defined as,

$$v_1\big(\vec{\nabla}D_1\vec{\nabla}\varphi_1 - (\Sigma_{a1} + \Sigma_{s12})\varphi_1 + \nu\Sigma_{f1}\varphi_1 + \nu\Sigma_{f2}\varphi_2\big) = \alpha\varphi_1, \tag{A.18}$$

$$v_2\big(\Sigma_{s12}\varphi_1 + \vec{\nabla}D_2\vec{\nabla}\varphi_2 - \Sigma_{a2}\varphi_2\big) = \alpha\varphi_2, \tag{A.19}$$

with the boundary conditions

$$\begin{aligned}&\varphi_g(0,y,z) = \varphi_g(L_x,y,z) = 0, \\ &\varphi_g(x,0,z) = \varphi_g(x,L_y,z) = 0, \qquad g = 1,2. \\ &\varphi_g(x,y,0) = \varphi_g(x,y,L_z) = 0,\end{aligned}$$

Using a similar procedure to the one followed for the other modes yields to the analytical solution of the α-modes problem

$$\varphi_2(x,y,z) = k \sin\left(\frac{m\pi}{L_1}x\right)\sin\left(\frac{n\pi}{L_2}y\right)\sin\left(\frac{p\pi}{L_3}z\right), \tag{A.20}$$

$$\varphi_1(x,y,z) = \frac{B^2_{m,n,p}v_2D_2 + v_2\Sigma_{a2} + \alpha}{v_2\Sigma_{s12}}\varphi_2(x,y,z), \tag{A.21}$$

where $m, n, p \in \mathbb{N}$.

The eigenvalues α are solutions of

$$\begin{aligned}
&\alpha^2 + \big[B^2_{m,n,p}v_2D_2 + v_2\Sigma_{a2} - v_1\nu\Sigma_{f1} && \text{(A.22)}\\
&\quad + v_1(\Sigma_{a1} + \Sigma_{s12}) + v_1D_1B^2_{m,n,p}\big]\alpha && \text{(A.23)}\\
&\quad + v_1D_1B^2_{m,n,p}(B^2_{m,n,p}v_2D_2 + v_2\Sigma_{a2}) && \text{(A.24)}\\
&\quad + v_1(\Sigma_{a1} + \Sigma_{s12})(B^2_{m,n,p}v_2D_2 + v_2\Sigma_{a2}) && \text{(A.25)}\\
&\quad - (v_1\nu\Sigma_{f1}(B^2_{m,n,p}v_2D_2 + v_2\Sigma_{a2}) + v_2\Sigma_{s12}v_1\nu\Sigma_{f2} = 0, && \text{(A.26)}
\end{aligned}$$

and $B^2_{m,n,p}$ is defined in Subsection A.1. If typical mrosacroscopic cross sections are used, the two solutions of this equation are real numbers. Different eigenvalues are obtained changing the value of m, n and p.

The value of the constant k for these modes (Equation (A.20)) is,

$$k = \frac{\pi^3}{8}\left(\frac{v_2\Sigma_{s12}}{\Sigma_{f1}(B^2_{m,n,p}v_2D_2 + v_2\Sigma_{a2} + \alpha) + v_2\Sigma_{s12}\Sigma_{f2}}\right).$$

APPENDIX B

BENCHMARKS DEFINITIONS

Reactor benchmarks, based on well defined problems with a complete set of input data and a unique solution, are widely used and accepted means of verifying the reliability of numerical simulations, i.e. to validate the accuracy, stability and efficiency of numerical nuclear codes. Problems are often very testing, but tend to be somewhat simplified, in order to make the analysis manageable to compare different models. Several realistic benchmarks have been defined in the literature. For completion purposes the definitions of the benchmarks used in this thesis are reproduced in here.

B.1 3D Homogeneous reactor

This is a three-dimensional prismatic reactor with a homogeneous material. The dimensions are 300 cm × 300 cm × 450 cm. The material cross sections are displayed in Table B.1. The velocities are $v_1 = 2.8{\cdot}10^7$cm/s and $v_2 = 4.4{\cdot}10^5$cm/s. The number of neutron produced by fission (ν) has been considered constant in the reactor core and equal to 2.5. The boundary conditions are zero flux at the boundary.

Table B.1: Macroscopic cross section values for the homogeneous reactor.

Material	g	D_g(cm)	Σ_{ag}(cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	$\Sigma_{sg,g+1}$(cm^{-1})
Fuel	1	1.5015	9.4003e-03	1.6850e-02	1.0100e-01
	2	4.3290e-01	8.2108e-02	6.0600e-03	-

B.2 3D Cuboid reactor (Benchmark E)

This transient is based on a non homogeneous prismatic reactor. It is composed of 72 equal nodes ($3 \times 3 \times 8$) of dimension $30 \times 30 \times 30$ cm^3 whose distribution is represented in Figure B.1. Table B.2 collects the material cross sections. The constants associated with the the six groups of precursors are displayed in Table B.3. The neutron velocities are $v_1 = 10^7$cm/s and $v_2 = 10^5$cm/s. The boundary conditions are zero flux.

The transient analyzed has been defined from a time-dependent perturbation to the fission cross sections of the material 1 so that the neutron power increases during 2 seconds and then it decreases. The functions that define the time evolution of the cross sections are

$$\nu\Sigma_{f1}(t) = \begin{cases} 0.01340976\big(1 + \dfrac{0.0122}{0.8}t\big), & 0 \le t \le 2, \\ 0.01381876\big(1 - \dfrac{0.0122}{0.8}(t-2)\big), & 2 \le t \le 4, \end{cases}$$

$$\nu\Sigma_{f2}(t) = \begin{cases} 0.34239791\big(1 + \dfrac{0.0122}{0.8}t\big), & 0 \le t \le 2, \\ 0.35284104\big(1 - \dfrac{0.0122}{0.8}(t-2)\big), & 2 \le t \le 4\,, \end{cases}$$

where $\nu\Sigma_{f1}$ and $\nu\Sigma_{f1}$ are measured in cm^{-1} and t in seconds.

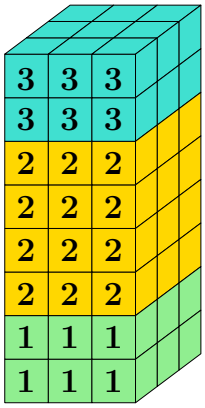


Figure B.1: Distribution of the materials for the cuboid reactor.

Table B.2: Material cross section for the cuboid reactor.

Mat.	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{fg} (cm^{-1})	Σ_{12} (cm^{-1})
1	1	1.695310	0.0139530	0.01340976	0.01340976	0.0164444
	2	0.409718	0.2614097	0.34239791	0.34239791	-
2	1	1.695310	0.0139954	0.01340976	0.01340976	0.0164444
	2	0.409718	0.2614200	0.34239791	0.34239791	-
3	1	1.695310	0.0139523	0.01340976	0.01340976	0.0164444
	2	0.409718	0.2614095	0.34239791	0.34239791	-

Table B.3: Constants for the neutron precursors for the cuboid reactor.

Group	1	2	3	4	5	6
β_g	0.000247	0.0013845	0.001222	0.0026455	0.000832	0.000169
$\lambda_g(s^{-1})$	0.0127	0.0317	0.115	0.311	1.4	3.87

B.3 Langenbuch benchmark

The Langenbuch transient (Langenbuch et al., 1977) is defined from a three-dimensional small LWR reactor composed of 77 fuel assemblies and 40 modelling the reflector as it is shown in Figure B.2. The fast resolution of the reactor makes it an attractive test. Table B.2 exposes the materials cross-sections. Table B.5 displays the neutron precursors data. The velocities are $v_1 = 1.25 \cdot 10^7$ cm/s and $v_2 = 2.5 \cdot 10^5$ cm/s. The value of ν is assumed constant for all material and energy group and equal to 2.5. Zero flux boundary values are applied at the boundary. The assemblies in the spatial discretization have a size of $20 \times 20 \times 20$ cm^3. Two types of transient are defined for this type of reactor.

Transient 1. It has been defined by perturbing the fission cross sections of material 1 represented in the Figure B.3 with striped pattern. We have defined two types of local sinusoidal perturbations that are out of phase between them. They are expressed as

$$\Sigma_{f,g}(t) = \Sigma_{f_0,g}(t) + \delta\Sigma_{f,g}(t) \qquad g = 1, 2. \tag{B.1}$$

The perturbation 1, represented in the Figure B.3 as P_1, is given by

$$\delta\Sigma_{f,g}(t) = 5 \cdot 10^{-4} \sin(\omega 2\pi t) \qquad g = 1, 2,$$

and the perturbation 2, denoted by P_2, is given by

$$\delta\Sigma_{f,g}(t) = 5 \cdot 10^{-4} \sin(\omega 2\pi t + \pi) \qquad g = 1, 2,$$

where $\omega = 1.0$ s^{-1}.

Transient 2. It has been initiated by the withdrawal of a bank of four partially inserted control rods (C1 in Figure B.2) at a rate of 3 cm/s over $0 < t < 26.7$ s. A second bank of control rods (C2 in Figure B.2) is inserted at the same rate over $7.5 < t < 47.5$ s. The transient is followed during 60 s. Figure B.4 represents the profile at initial state, $t = 0.0$ s, and the profile at $t = 60$ s.

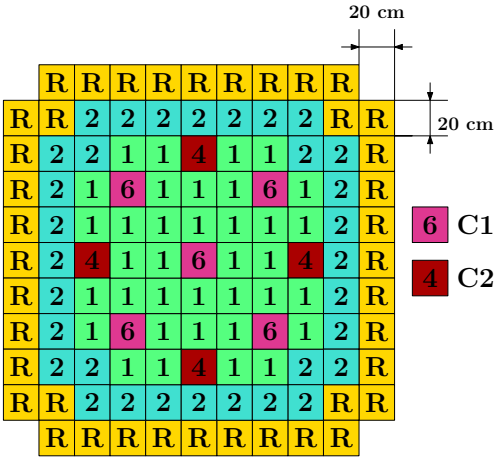


Figure B.2: Distribution of the materials for the Langenbuch reactor.

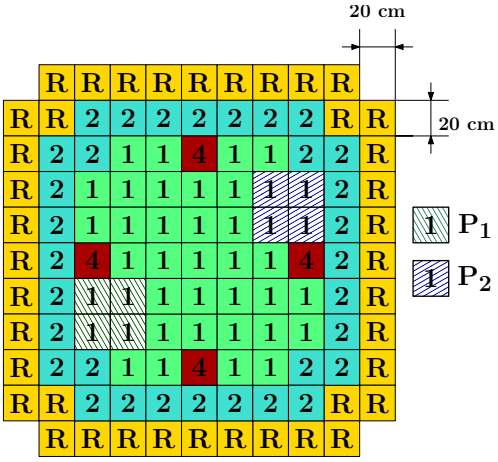


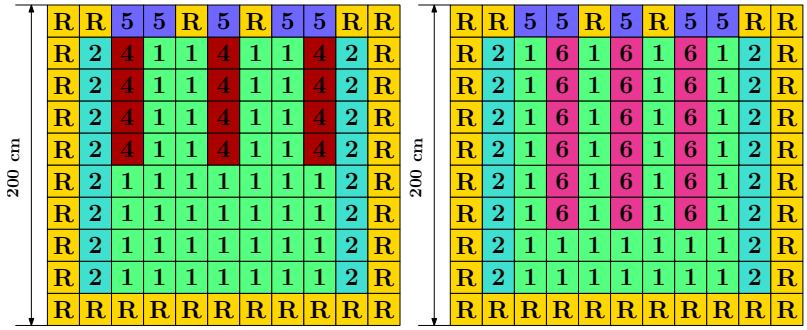
Figure B.3: Location of the perturbation areas for the transient 2.

Table B.4: Cross sections data of the Langenbuch reactor.

Material	**Group**	$\mathbf{D_g}$ (cm)	$\boldsymbol{\Sigma_{ag}}$ (cm^{-1})	$\boldsymbol{\nu\Sigma_{fg}}$ (cm^{-1})	$\boldsymbol{\Sigma_{s12}}$ (cm^{-1})
1 – Fuel	1	1.423913	0.01040206	0.00647769	0.01755550
	2	0.356306	0.08766217	0.11273280	
2 – Fuel	1	1.425611	0.01099263	0.00750328	0.13780040
	2	0.350574	0.09925634	0.01717768	
R –Reflector	1	1.634227	0.00266057	0.00000000	0.02759693
	2	0.264002	0.049363510	0.00000000	
4,6 – Absorbent	1	1.423913	0.01095206	0.00647769	0.11273228
	2	0.356306	0.09146217	0.01755550	
5 – Reflector + Absorbent	1	1.634227	0.00321050	0.00000000	0.02759693
	2	0.264002	0.05316351	0.00000000	

Table B.5: Neutron precursors data of the Langenbuch reactor.

Group (k)	1	2	3	4	5	6
β_k	0.000247	0.0013845	0.001222	0.0026455	0.000832	0.000169
λ_k^d(s^{-1})	0.0127	0.0317	0.115	0.311	1.4	3.87

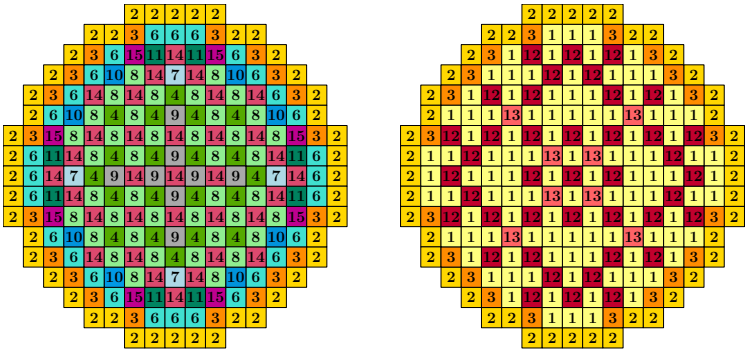


(a) $t = 0.0$ (b) $t = 60.0$

Figure B.4: Langenbuch profiles during the transient 1.

B.4 NEACRP benchmark

A critical configuration of the NEACRP (case A1) benchmark is defined (Finnemann and Galati, 1991). The core is composed of 3978 assemblies where each one measures radially 21.606 cm × 21.606 cm. Axially, the reactor has total height of 427.3 cm divided into 18 layers, from bottom to top, 30.0 cm, 7.7 cm, 11.0 cm, 15.0 cm, 30.0 cm (10 layers), 12.8 cm. 12.8 cm, 8.0 cm and 30 cm. The definition of the radial geometry is shown in Figure B.5 and the axial profile definition is presented in Figure B.6. The velocities are $v_1 = 2.8 \cdot 10^7$ cm/s and $v_2 = 4.4 \cdot 10^5$ cm/s. Zero boundary conditions are imposed.



(a) 3-17th plane (b) 18th plane

Figure B.5: Radial definition of the NEACRP Reactor.

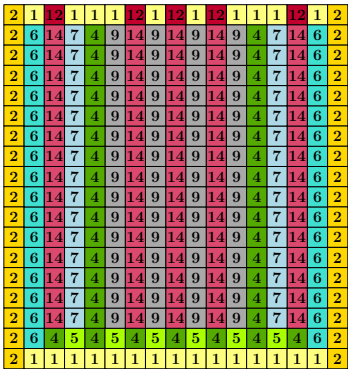


Figure B.6: Axial definition of the NEACRP reactor.

Table B.6: Cross sections data of the NEACRP reactor.

Mat.	**g**	$\mathbf{D_g}$ (cm)	$\boldsymbol{\Sigma_{ag}}$ (cm^{-1})	$\boldsymbol{\nu\Sigma_{fg}}$ (cm^{-1})	$\boldsymbol{\Sigma_{fg}}$ (cm^{-1})	$\boldsymbol{\Sigma_{sg\,g+1}}$ (cm^{-1})
1	1	5.9263e+00	2.6191e-04	0.0000e+00	0.0000e+00	2.7988e-02
	2	8.2277e-01	1.9865e-01	0.0000e+00	0.0000e+00	-
2	1	1.1276e+00	1.1878e-03	0.0000e+00	0.0000e+00	2.3161e-02
	2	1.6978e-01	1.9865e-01	0.0000e+00	0.0000e+00	-
3	1	1.1276e+00	1.1878e-03	0.0000e+00	0.0000e+00	2.0081e-02
	2	1.6978e-01	1.9865e-01	0.0000e+00	0.0000e+00	-
4	1	1.4624e+00	8.4782e-03	5.0150e-03	6.1479e-14	1.9684e-02
	2	3.9057e-01	6.2649e-02	8.7684e-02	1.1515e-12	-
5	1	1.4637e+00	8.8239e-03	5.6085e-03	6.9275e-14	1.9435e-02
	2	3.9489e-01	7.0055e-02	1.0421e-01	1.3685e-12	-
6	1	1.4650e+00	9.1498e-03	6.1819e-03	7.6811e-14	1.9195e-02
	2	3.9855e-01	7.6924e-02	1.1951e-01	1.5694e-12	-
7	1	1.4641e+00	9.0882e-03	5.5830e-03	6.8996e-14	1.8525e-02
	2	4.0582e-01	7.7758e-02	1.0286e-01	1.3509e-12	-
8	1	1.4641e+00	9.1752e-03	5.5741e-03	6.8913e-14	1.8221e-02
	2	4.0950e-01	8.0371e-02	1.0229e-01	1.3433e-12	-
9	1	1.4642e+00	9.2609e-03	5.5649e-03	6.8817e-14	1.7919e-02
	2	4.1317e-01	8.2990e-02	1.0166e-01	1.3351e-12	-
10	1	1.4653e+00	9.4110e-03	6.1564e-03	7.6530e-14	1.8287e-02
	2	4.0923e-01	8.4530e-02	1.1804e-01	1.5501e-12	-
11	1	1.4655e+00	9.4969e-03	6.1474e-03	7.6449e-14	1.7985e-02
	2	4.1280e-01	8.7097e-02	1.1741e-01	1.5419e-12	-
12	1	5.5576e+00	2.7396e-03	0.0000e+00	0.0000e+00	2.4796e-02
	2	8.7000e-01	3.7064e-02	0.0000e+00	0.0000e+00	-
13	1	5.6027e+00	2.4190e-03	0.0000e+00	0.0000e+00	2.5209e-02
	2	8.6358e-01	3.3753e-02	0.0000e+00	0.0000e+00	-
14	1	1.4389e+00	1.0956e-02	4.9121e-03	6.0265e-14	1.6492e-02
	2	4.0090e-01	8.8237e-02	8.4861e-02	1.1145e-12	-
15	1	1.4413e+00	1.1579e-02	6.0592e-03	7.5335e-14	1.6053e-02
	2	4.0669e-01	1.0257e-01	1.1622e-01	1.5263e-12	-

B.5 C5G7 benchmark

The configuration of the C5G7 benchmark (Lewis et al., 2001) consists of a nuclear reactor core with MOX and UO_2 square fuel assemblies surrounded by a moderator region. Each fuel assembly is made up of a 17×17 square pin level cells. The side length of each pin cell is 1.26 cm and all fuel pins and guide tubes have a 0.54 cm of radius. A single moderator composition (water) is given for use in all of the pin cells and for use in the moderator (reflector) surrounding the assemblies. For the two-dimensional domain, vacuum boundary conditions are applied to the right and to the bottom of the geometry while reflected boundary conditions are applied to the top and left of the geometry.

For the three-dimensional configuration, the *Unrodded* problem presented in (Smith et al., 2006) is tested. It is composed of three fuel planes where each fuel assembly has a height of 14.28 cm in the z direction and an additional water reflector of 21.42 cm is added axially (Figure B.7). The z boundary conditions are reflected below and vacuum above.

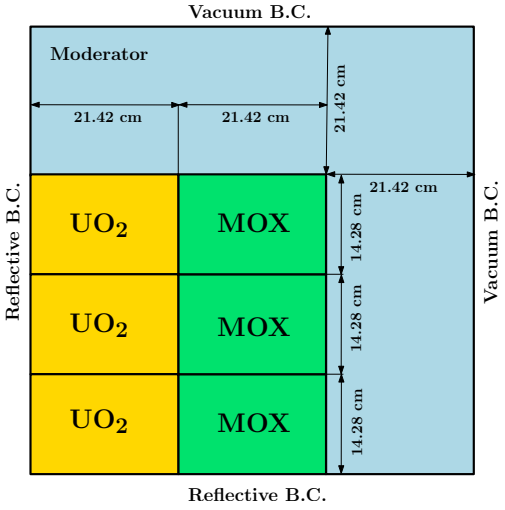


Figure B.7: Axial distribution for the 3D-C5G7 reactor.

The cross sections for each material are provided in (Lewis et al., 2001). A complete description of the MOX and UO_2 fuel assemblies, on the radial planes, is shown in Figure B.8.

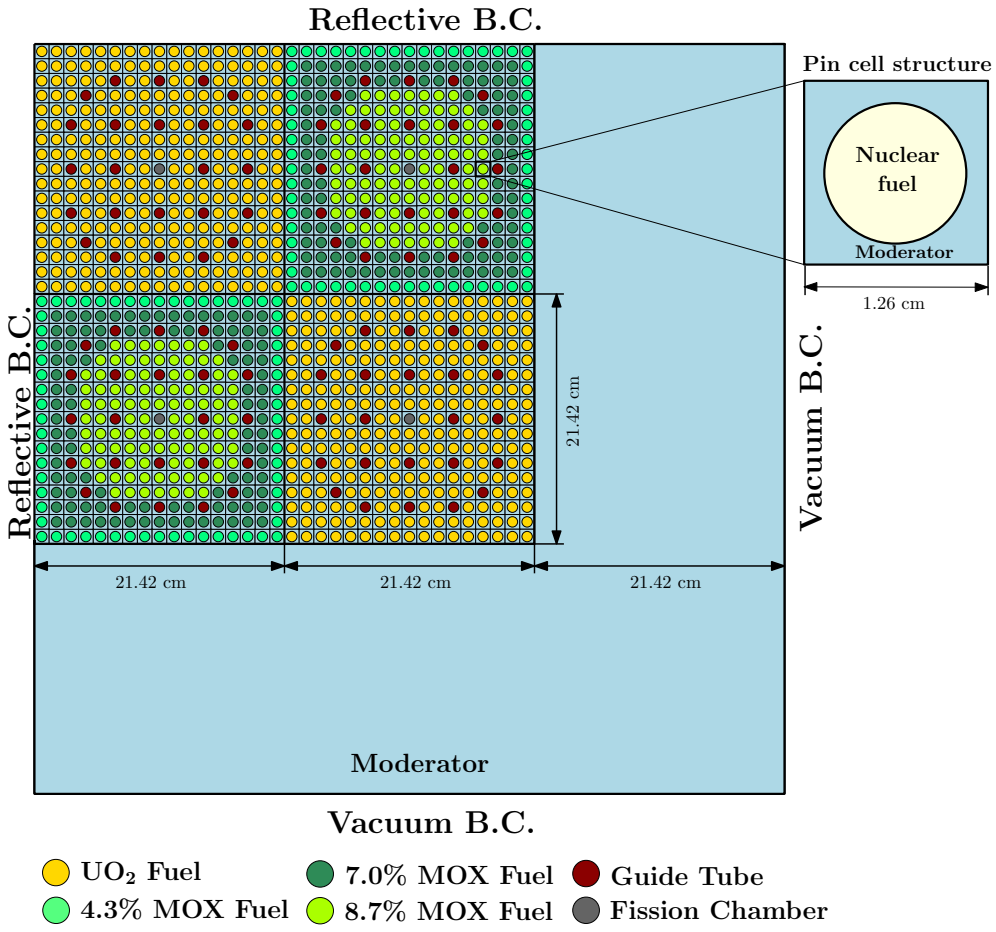


Figure B.8: Material distribution of the C5G7 reactor.