Development and acceleration of a 3D characteristics method including an axial polynomial expansion of cross sections

Andrea Gammicchia

université
PARIS-SACLAY

Development and acceleration of a 3D characteristics method including an axial polynomial expansion of cross sections

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 576, Particules Hadrons Énergie et Noyau : Instrumentation, Image, Cosmos et Simulation (PHENIICS)

Spécialité de doctorat: Énergie nucléaire

Unité de recherche : Université Paris-Saclay, CEA, Service d'Études des Réacteurs et de Mathématiques Appliquées, 91191 Gif-sur-Yvette, France

Référent : Faculté des Sciences d'Orsay

Thèse présentée et soutenue à Saclay, le 29/09/2021, par

Andrea GAMMICCHIA

Composition du jury:

Pierre Desesquelles Professor, Université Paris-Saclay	Président
Benoit Forget Professor, Massachussets Institute of Technology	Rapporteur
Piero Ravetto Adjunct Professor, Politecnico di Torino	Rapporteur
Danny Lathouwers Assistant Professor, TU Delft	Examinateur
Richard Sanchez Former Research Director, CEA Saclay	Examinateur
Sandra Dulla Professor, Politecnico di Torino	Directrice
Simone Santandrea Ph.D., CEA Saclay	Co-encadrant & examinateur

Thèse de doctorat

NNT: 2021UPASP098

Acknowledgments

The Ph.D. represents for me a great accomplishment, which crowns and values the long path taken so far. My personal commitment, however considerable, would never have been enough without the contribution of many people and the mark they have left on me in all these years of study and growth. I also owe to them the achievement of this goal.

Thanks to my family for so many years of tireless support and trust, too often taken for granted and rarely corresponded with adequate words of gratitude. Their guide, their example and their sacrifices have made me the person I am, contributing substantially to my achievements. A special thanks goes to my brother, my first travel companion, for enduring my challenging character for so long.

I would like to thank the jury members. I felt honored to defend my thesis before people of such profound cultural and human depth, including those same professors whose passion led me to choose the nuclear path. I am grateful to my thesis director, for her supervising role and the great interest shown in the work progress. A huge thanks to my tutor, for giving me the chance to work with him, for his great support, especially during lockdowns, and his contagious sympathy: words will never be enough to express my gratitude.

Thanks to all my friends, to those I grew up with and I look forward to meeting every time I am in Turin, and to the ones I have known for less time but whose friendship is meant to last. I owe them the pleasant conversations, the great laughs and the many good times spent together, before and during my Ph.D. years. Thanks to all the people of SERMA and, in particular, of the LLPR and LTSD laboratories, for their warm welcome and the stimulating working atmosphere, which motivated me to stay.

Last but not least, thanks to my girlfriend for her love, for accepting and enduring the distance between us, for her being simply extraordinary. Thanks for all that we are and we have built so far.

Abstract

In the framework of deterministic nuclear reactor calculations, the present work introduces a polynomial expansion for the spatial profiles of macroscopic cross sections, with the aim to apply this higher-order scheme to depleted systems. This thesis represents a further enhancement of the 3D Method Of Characteristics (MOC) implemented in the Two-three Dimensional Transport (TDT) solver of the APOLLO3® code, used to solve the time-independent neutron transport equation and thus determine the effective multiplication factor.

In order to improve the representativeness of TDT results, the solver applicability was first extended from 2D to 3D axially extruded geometries. A drawback of the 3D MOC, though, was the important computational cost associated to sweeping the trajectories over a very large number of regions, which was a direct consequence of the Step-Constant (SC) spatial approximation used for the moments of the angular flux and for the cross sections. Hence, to improve performance, a polynomial development was introduced that described the axial profiles of the flux moments. However, since cross sections were not involved in the polynomial treatment, its application was limited to Beginning Of Cycle (BOC) configurations. If the higher-order MOC is to be applied to depleted systems, which is the scope of this thesis, it is sufficient to extend the polynomial expansion to cross sections, so as to properly represent the axial gradients appearing as the burnup builds. In this way, one can expect to benefit from the advantages of the polynomial method also at higher burnup levels.

The introduction of the higher-order description of cross sections has required the adaptation of the transmission and balance MOC equations. To optimize the flops amount, the derived form of the latter envisages the decomposition of a global system, comprising the equations for all polynomial coefficients of the angular flux moments, into subsystems associated to the spherical harmonics order. The obtained transmission equation contains integrals that cannot be expressed in closed-form in the general case, for which an adaptive Gauss-Legendre quadrature has been developed. When possible, the evaluation of the transport coefficients that appear in this same equation is performed in a vectorized fashion, leading to a significant computing time reduction.

A specific strategy has been developed to allow TDT to interface with the self-shielding module and the depletion solver, since these are currently both incompatible with higher-order spatial description. To this purpose, two kinds of axial meshes are adopted: a coarse mesh, used in TDT for the flux calculation, and a Gaussian mesh with the axial layers corresponding to the Gauss-Legendre abscissas, used for self-shielding and depletion. The Gaussian mesh permits reducing part of the memory load linked to the use of two meshes.

In order to accelerate the polynomial cross sections approach by the DP_N synthetic technique, adapted equations have been derived starting from their MOC counterparts. Once

again, vectorization has been exploited for the DP_N coefficients evaluation. The adaptation has concerned also the non-linear least-square fitting technique, already implemented in TDT, which is useful to lessen the memory load due to coefficients storage.

Applications have involved different Pressurized Water Reactors (PWR) subsystems, comparing the results of the new method with those of the traditional SC approach: in particular, the depletion study of a full PWR assembly has shown the decisive advantage of the polynomial expansion in terms of memory and computing time.

Contents

Chapter 1

Introduction

1.1 The importance of improving nuclear simulations

In 2018, nuclear reactors were responsible for 17.6% of the OECD gross electricity production (10.1% worldwide), a relevant but lower percentage than the 20.8% of 2010, before the Fukushima Daiichi accident [1]. In a global scenario in the name of eco-sustainability, the nuclear field struggles to receive recognition for its advantages and continues to suffer from the oldest of its problems, mistrust, returning to the fore in the current debate on the replacement of old power plants in countries like France and the United States. In this respect, the permanence of nuclear among the main energy sources is closely tied to its capability to guarantee competitiveness through efficiency and safety. At the same time, working to reduce the fears related to nuclear security is essential to attain social acceptability.

What is the role of numerical simulations in this awareness campaign? In the first place, the availability of excellent nuclear codes comes in handy during reactor design, allowing to best reconcile safety and working parameters. But there is a further benefit: better knowledge and predictive ability of real systems leads to greater confidence in our engineering capabilities, allowing for higher-quality and more effective scientific divulgation. Clearly, the latter is essential to restore people's trust. From this perspective, the long-term goal of performing high-fidelity 3D calculations of arbitrary full-core designs represents an actual mission for nuclear engineers.

Starting from a general viewpoint, the next paragraph provides more context to the concept of *high-fidelity* in the numerical simulation framework, also mentioning the related issues, and then focuses on the aspects that concern this thesis more closely.

1.2 Approximating a representation

In the perspective of the description of reality, computational physics can be thought of as a double approximation: the scientific method is applied to find laws that best describe reality and constitute physics (reality $\rightarrow$ physics); these laws, in turn, need be adapted for numerical implementation (physics $\rightarrow$ numerics). There is in fact an additional level of simplification, given by the ancient idea that distinguishes apparent reality from truth (phenomena from *noumena*, to paraphrase I. Kant), yet this distinction can be considered as part of the first

level. Physics is actually a product of man, subjected to the limits of human intelligence; the same is valid for applied physics, since there is no computer smarter than man (nowadays), but with numerical approximations as further limitation.

Both disciplines should aim at filling the gap between the properties of their object of study and the approximated results they obtain. Considering computational physics, as it will be done from now on, the object of study is the physical system, either the best available or the most fit-for-purpose representation of reality; providing a *good* approximation of its behavior is the task of numerical simulations.

This thesis work falls entirely under the branch of physics that studies the behavior and interactions of particles inside a nuclear fission reactor; this field is commonly referred to as (*fission*) *reactor physics*. Traditionally, two types of computational methods are employed in nuclear system simulation: *stochastic* and *deterministic*. The former are only affected by the approximations related to the floating-point representation of numbers and to the imperfect random number generators; they pay for this adherence to physics with conspicuous computing times. On the other hand, deterministic methods introduce errors and biases into the analyzed system as a consequence of the hypotheses made: in reactor physics, the results depend on the numerical discretization of the phase space, and therefore on the spatial mesh and functional representation of quantities, on the maximum anisotropy order and number of angular quadrature directions, on the number of energy groups. Limiting these approximations often clashes with the available resources: the deterministic approach ultimately consists in finding the best compromise between performances and proximity to reference solution, usually provided by computationally expensive stochastic calculations, by acting on the discretization parameters.

In the light of what said, the added value of any new deterministic method, like the one proposed in this manuscript, is given by one or both of the following characteristics: a better adherence to physics (by comparison with the reference solution, when available) and improved performances with respect to pre-existing techniques. As will be shown, the present work has the ambition to pursue both, aiming at taking a further step towards the feasibility of one-step full-core calculations.

1.3 A high-order description of cross sections

Considering 3D axially extruded geometries[1], the basic idea behind this thesis is to substitute a piece-wise constant spatial description of macroscopic cross sections, along the direction of extrusion, with a development in terms of polynomial functions. The so-called *step-constant* approximation, representing physical quantities as constant within every computational region, is instead maintained on the radial plane. The new developments are just the latest in a series of enhancements successively introduced in the Method Of Characteristics (MOC) solver of APOLLO3®, which will be described in detail in the background section. Here I merely evoke the main concepts and the reasons that pushed for the improvements made.

[1]Extruded geometries are obtained by repetition of an original geometry along the direction of extrusion. By (3D) *axially extruded geometries* one refers to the 3D systems composed of one or more portions with a 2D section (on the xy plane) that does not vary along z.

The first step was the extension of the MOC solver to 3D geometries obtained through extrusion of one or more 2D sections. This was realized during a Ph.D. work [2], which also produced several publications [3, 4, 5]. In the same period, other research teams developed 3D MOC solvers [6, 7, 8, 9, 10]: these and other approaches will be discussed later. This fundamental implementation opened the floodgates to new and more realistic neutronics calculations; as a drawback, though, it quickly becomes expensive as the size of the problem grows, due to the cost of sweeping the trajectories over a great number of regions. It was soon realized that the non-negligible, yet quite smooth, axial gradients of the neutron flux could be addressed in a smarter way, that is, by resorting to a higher-order spatial description. A second Ph.D. work [11, 12, 13] therefore introduced a polynomial development for the flux, to make the 3D MOC more efficient by acting directly on the main weakness of the step-constant approach: the number of regions. The polynomial method actually allows the use of a coarser mesh, wherein the axial layers have a greater thickness. The predicted benefits were broadly confirmed by the results: with regards to subsystems of the ASTRID reactor project [14], the time and memory required for eigenvalue calculations diminished by approximately 60% and 70%, respectively, while providing the same level of accuracy.

The initial implementation of the higher-order method, however, limited its application to Beginning-Of-Cycle (BOC) configurations: in such cases, nuclide concentrations have not yet undergone depletion and therefore macroscopic cross sections can be considered as constant within each region with good approximation. But, as the burnup increases, cross sections acquire axial gradients that cannot be properly represented by a coarse mesh: to avoid being forced to rely on the step-constant approach, the most straightforward strategy is then to extend the polynomial development to cross sections. This permits adopting a less refined mesh also when dealing with depleted systems. Furthermore, a description in terms of continuous functions, at the spatial scales of the reactor subsystems that will be presented, constitutes a better representation of the real physical profiles. Taking also into account that the scope of this work is ultimately to improve the MOC solver performances, by benefiting from the advantages of the higher-order technique in case of non-negligible burnup, it is evident that both of the aforementioned possible requirements for a new method are satisfied.

The manuscript is structured into three major parts, summarized as follows.

Background (chapters 2 - 4) The general context to the thesis is provided, introducing the basics of reactor physics and the common discretizations adopted by deterministic methods. After describing the way APOLLO3® solves the neutron transport problem, the discussion focuses on the use of the MOC in neutronics solvers and, in particular, on its implementation in the code used for this work.

New developments (chapters 5 - 6) The adaptation of MOC equations to the case of axially polynomial cross sections is presented, as well as the numerical tools required by the method, the optimization strategies adopted and its acceleration.

Results and conclusions (chapters 7 - 8) The performances of the new method are compared to the pre-existing technique for different Pressurized Water Reactors (PWR) case studies, before drawing conclusions and outlining potential future developments.

Part I

BACKGROUND

Chapter 2

Principles of reactor physics

The interest for the nuclear source in the energy production field is motivated by the impressive strength of nuclear bonds, compared to the ones at atomic and molecular scales. Whenever a compound is formed (a nucleus, a molecule, etc.), a certain amount of energy is released, corresponding to the mass defect of the resulting entity with respect to the sum of the individual masses of its components. This is referred to as *binding energy* and determines directly the net energy gain of a reaction. Dealing with nuclear reactions, this outcome is typically six orders of magnitude greater than in the case of chemical reactions.

The binding energy of a nuclide does not directly depend on the number of nucleons: looking at the graph of its per-nucleon value (Fig. 2.1), it is possible to notice an increasing trend for the small A's (number of nucleons in nucleus) and decreasing for the heavy nuclides, with the absolute maximum located in proximity of Fe-56. It follows that two strategies are possible to obtain a net positive energy outcome: the *fusion* of light nuclides or the *fission* of the heavy ones. The present work and, more generally, all the current civil applications of the nuclear engineering belong to the framework of the second kind of reactions. The exploitation of nuclear fusion, the process governing the life cycle of stars, is still in an experimental phase.

Reactor physics therefore mainly concerns the control over the induced fission process, which is achieved by knowing the distribution in the phase space (position, direction and energy) of the particles responsible for the process, neutrons. After being discovered by Chadwick in 1932, neutrons were actually used as projectiles by Fermi's team in their experiments aimed at obtaining transuranic nuclides. Due to the absence of electric charge, neutrons are not affected by Coulomb potential and thus had more chances to be absorbed by the target nuclei of uranium. Surprisingly enough, the physicists observed several radioactive decay periods, a fact that could be explained by theorizing the splitting of the original nuclei, which was demonstrated in 1938 by Hahn and Strassmann. Each fission releases about 200 MeV[1] and, most notably, entails the emission of a number $\nu > 1$, on average, of secondary neutrons[2], which makes possible a chain reaction.

Fission is not, however, the only possible effect of the interaction between a neutron and a heavy nuclide: it is therefore important to know the probabilities of each reaction occurring, traditionally expressed in terms of cross sections. The following section will present these

[1] $1\,eV = 1.60218\,10^{-19}\,J$

[2] $\nu = 2.42$ for U-235

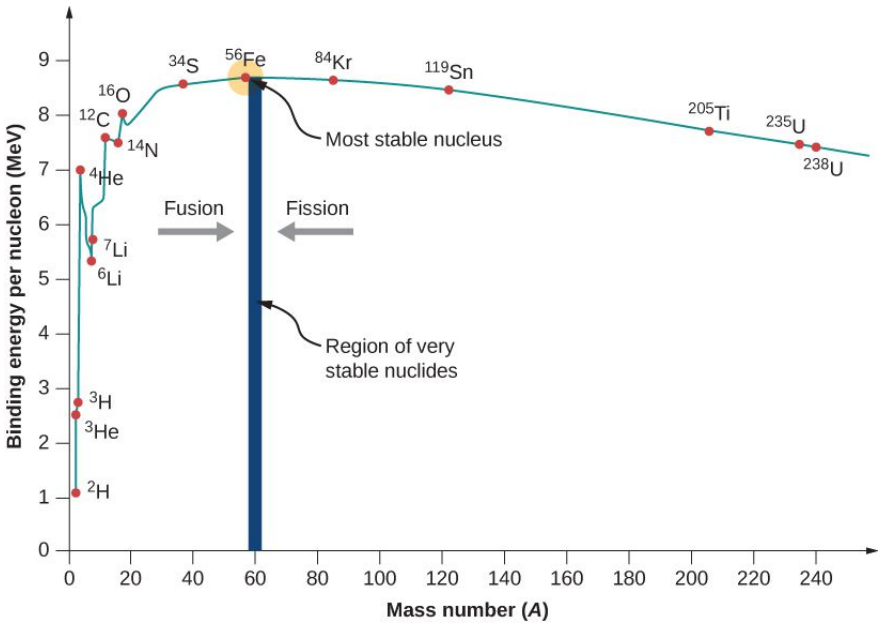


Figure 2.1: Binding energy per nucleon as a function of the mass number. As the graph suggests, nuclei with high binding energy can be obtained either by fusion of lighter elements or by fission of heavier ones. Image source: [15].

fundamental physical quantities, which play a major role in this thesis, before considering the well-known equation ruling the neutrons behavior in the phase space.

2.1 Cross sections

The classical way to introduce the concept of cross section is to consider a homogeneous material layer of nuclide volumic density N $\left(\left[\frac{atoms}{cm^3}\right]\right)$, obtained through extrusion of an arbitrary surface S; the thickness δ is such as to contain only one atom. The interaction probability for a neutron crossing S can be experimentally determined, for different values of energy of the incident neutron and of system temperature. Denoting by p its value, this can be expressed in geometrical terms by associating a fictitious cross-sectional area σ to the nuclides:

$$p = \frac{S\delta N\sigma}{S} = \delta N\sigma. \tag{2.1}$$

σ is the microscopic cross section, so called because it is an intrinsic property of the nuclide and not of its concentration. It is measured in $barns$[3]. On the other hand, the $N\sigma$ quantity is the macroscopic cross section, measured in cm^{-1} and commonly denoted by Σ. Based on Eq. (2.1), it represents the probability of interaction per unit path. Its inverse, $\frac{1}{\Sigma}$, represents the mean free path, which is the distance neutrons travel, on average, before colliding. At a given temperature, in general, macroscopic cross sections will depend on the spatial position (or better, on the local material composition) and on the energy of the colliding neutron. The angular dependence can be neglected, because reactor materials, since they are either

[3] $1\,barn = 10^{-24}\,cm^2$

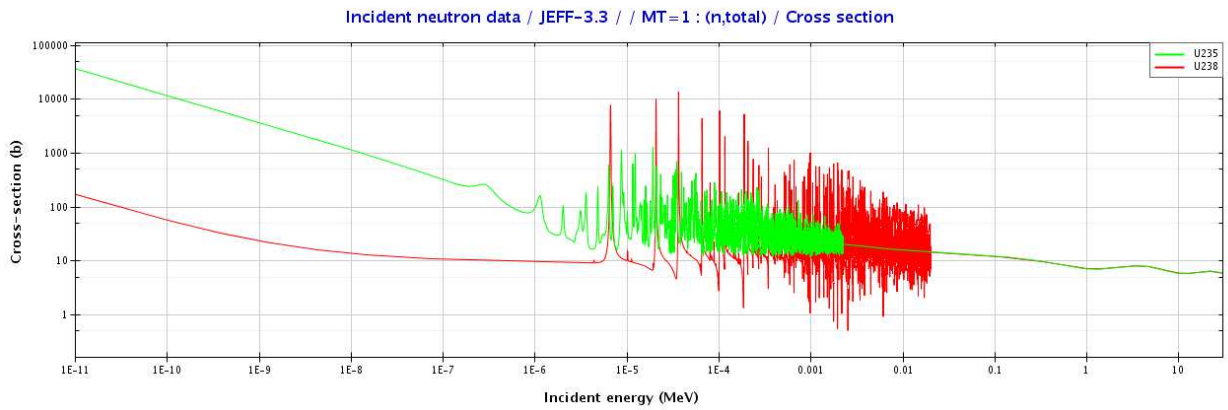


Figure 2.2: Microscopic total cross sections of U-235 and U-238 as a function of the energy of the incident neutron [16].

liquid or gaseous or constituted by very small crystals with respect to the mean free path of neutrons, are considered isotropic. Moreover, while performing an analysis or a simulation at a given burnup level (see section 3.4), the time variable can also be omitted, as the cross section variation is very slow compared to neutron life-time (no more than $10^{-3}\,s$).

It is appropriate to distinguish cross sections according to the kind of reaction. A neutron interacting with a nuclide gives rise to two possible scenarios, absorption or scattering, depending on whether the neutron penetrates the target or not, respectively. The further following classification is possible for both categories:

- Absorption

  Capture the neutron entering the nucleus brings it to an excited state; the consequent de-excitation occurs through radioactive decay and gives rise to the emission of a γ ray, a charged particle or more than one neutron.

  Fission the absorption of the neutron leads to the split of the nuclide and to the emission of secondary neutrons.

- Scattering

  Elastic the neutron is diffused without excitation of the target.

  Inelastic after collision, the nuclide is excited and then returns to the fundamental state through γ emission.

For the purpose of this thesis, a total microscopic cross section σ_T can be expressed as

$$\sigma_T = \sigma_A + \sigma_S = \sigma_C + \sigma_F + \sigma_S, \tag{2.2}$$

σ_A, σ_S, σ_C, σ_F being respectively the absorption, scattering, capture and fission microscopic cross sections. An analogous relation can be written for their macroscopic counterparts.

The energy dependence of microscopic cross sections is notoriously highly non-linear, since it results from the ensemble of excitation levels of the nucleus. For the heavy nuclides,

the profile can be very complex, due to the presence of several resonances, i.e. energy ranges wherein the cross section value varies of many orders of magnitude compared to the neighboring intervals. Figure 2.2 reports the total microscopic cross sections of U-238 and U-235 as a function of the energy E; it is possible to identify at least four domains, which will be useful later while presenting the multi-group representation:

- the *thermal domain* (up to $\sim 1\,eV$), characterized by a smooth profile, approximately proportional to $\frac{1}{E}$.
- the *resolved resonance region*, wherein resonances are identifiable by experimental measurements.
- the *unresolved resonance region*, presenting resonances whose profile cannot be adequately determined, due to insufficient experimental energy resolution.
- the *fast domain* (above $\sim 0.2\,MeV$), wherein the profile returns relatively smooth but without a regular behavior.

This subdivision strongly depends on the nuclide considered. The second and the third intervals are sometimes considered part of a unique range, called *epithermal*; moreover, their distinction is actually due to the present accuracy of measurements, rather than to a different physical behavior. Since the fission cross section of U-235 is about $2 \div 3$ decades higher in the thermal than in the fast energy range, thermal reactors (by far the most widespread nuclear reactor design) require a low-absorbing material with a small mass number to *moderate* neutrons, that is, to slow them down. In fact, neutrons arising from fission have energies of the order of some MeV, and need be thermalized in as few interactions as possible, to avoid being lost by resonance capture. This is possible by resorting to a moderator made of light nuclei, to which a neutron can transfer a significant fraction of its energy (up to 99.99995% in the case of hydrogen in water[4]). The traditionally employed moderators are light and heavy water and graphite.

2.2 The neutron transport equation

The reason for expressing interaction probabilities in terms of per-unit-path quantities is due to the fact that neutrons, and particles in general, need to move to interact. Consequently, reactor physicists prefer describing the neutron population in the phase space (spanned by the position vector $\vec{r}$, the direction vector $\vec{\Omega}$ and the energy - scalar - value E) considering the path they travel per unit time, volume, solid angle and energy: the concentration n is thus substituted by the *flux* ψ, which is generally function of $\vec{r}$, $\vec{\Omega}$, E and time t. The relation between the flux and the concentration reads

$$\psi(\vec{r}, \vec{\Omega}, E, t) = v(E)\, n(\vec{r}, \vec{\Omega}, E, t), \tag{2.3}$$

v being the neutron velocity at energy E. Multiplying the macroscopic cross section for the general R reaction, Σ_R, by the neutron flux one obtains the reaction rate r_R, which indicates

[4]Supposing elastic scattering with a proton having zero initial kinetic energy

the number of R reactions occurring at $(\vec{r}, \vec{\Omega}, E)$ per unit time:

$$r_R(\vec{r}, \vec{\Omega}, E, t) = \Sigma_R(\vec{r}, E)\, \psi(\vec{r}, \vec{\Omega}, E, t). \tag{2.4}$$

The aforementioned quantities, expressed per unit solid angle, are usually referred to as *angular*. When the information on the direction is not needed, the corresponding *scalar* values are considered: for instance, the scalar flux ϕ is obtained as

$$\phi(\vec{r}, E, t) = \oint d\vec{\Omega}\, \psi(\vec{r}, \vec{\Omega}, E, t). \tag{2.5}$$

This occurs in practical applications where the interest is often to determine the total amount of reactions taking place within a certain volume and energy range, which requires taking all angles into account and integrating the scalar reaction rate over the chosen range.

For the typical energies of the neutrons inside a reactor ($\sim 0\,eV \div 20\,MeV$) and under the hypothesis that they do not collide between themselves (reasonable, since there is approximately one neutron every 10^{14} nuclei), the equation governing the neutron flux is the non-relativistic linear Boltzmann equation [17]

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \psi(\vec{r}, \vec{\Omega}, E, t)}{\partial t} &+ \vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}, E, t) + \Sigma_T(\vec{r}, E)\, \psi(\vec{r}, \vec{\Omega}, E, t) \\ &= \int_0^\infty dE' \oint d\vec{\Omega}'\, \Sigma_S(\vec{r}, \vec{\Omega} \cdot \vec{\Omega}', E' \to E)\, \psi(\vec{r}, \vec{\Omega}', E', t) \\ &+ \sum_{i=1}^{N_f} \frac{\chi_i(E)}{4\pi} \int_0^\infty dE'\, \nu\Sigma_{F,i}(\vec{r}, E')\, \phi(\vec{r}, E', t) + S(\vec{r}, \vec{\Omega}, E, t), \end{aligned} \tag{2.6}$$

which is of differential type in time and space and integral in angle and energy. The adopted definitions and conventions are specified in the following.

$\vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}, E, t)$ is the streaming term, representing the net rate at which neutrons leave volume $d\vec{r}$ (balance of outflow and inflow) with direction $\vec{\Omega}$ and energy E.

$\Sigma_T(\vec{r}, E)\, \psi(\vec{r}, \vec{\Omega}, E, t)$ is the removal term, that is, the number of neutrons that disappear from the phase space, due to absorption or scattering, per unit time.

$\int_0^\infty dE' \oint d\vec{\Omega}'\, \Sigma_S(\vec{r}, \vec{\Omega} \cdot \vec{\Omega}', E' \to E)\, \psi(\vec{r}, \vec{\Omega}', E', t)$ is the transfer term, accounting for the neutrons introduced in the phase space by scattering from all energies and directions, per unit time; by an abuse of notation, the differential cross section Σ_S indicates the product of the actual macroscopic scattering cross section by the probability density function f_S, such that $f_S\, d\vec{\Omega}\, dE$ is the probability for a colliding neutron in $(\vec{r}, \vec{\Omega}', E')$ of being emitted in $(\vec{r}, d\vec{\Omega}$ about $\vec{\Omega}, dE$ about $E)$:

$$\Sigma_S(\vec{r}, \vec{\Omega} \cdot \vec{\Omega}', E' \to E) = \Sigma_S(\vec{r}, E')\, f_S(\vec{r}, \vec{\Omega} \cdot \vec{\Omega}', E' \to E). \tag{2.7}$$

As was stated in the previous section, material isotropy allows neglecting the dependence on the incident direction $\vec{\Omega}'$ and to consider just $\vec{\Omega} \cdot \vec{\Omega}'$ (the cosine of the angle in between), instead of $\vec{\Omega}' \to \vec{\Omega}$.

$\sum_{i=1}^{N_f} \frac{\chi_i(E)}{4\pi} \int_0^\infty dE'\, \nu\Sigma_{F,i}(\vec{r}, E')\, \phi(\vec{r}, E', t)$ is the fission term, which represents the number of neutron emerging from the fission of the N_f fissile isotopes per unit time, in the phase space. Since fission can be assumed as an isotropic phenomenon in nuclear applications, neutrons are equally emitted in all directions, according to the emission spectra $\{\chi_i\}$, and the scalar flux appears inside the integral. $\nu\Sigma_{F,i}(\vec{r}, E')$ is a compact way to express

$$\nu\Sigma_{F,i}(\vec{r}, E') = \nu_i(E')\, \Sigma_{F,i}(\vec{r}, E'), \tag{2.8}$$

where the number of secondary neutrons and the fission cross section both refer to isotope i and energy E'.

$S(\vec{r}, \vec{\Omega}, E, t)$ is the external source.

2.2.1 Eigenvalue formulation

In nominal working conditions, the neutron flux is expected to remain constant in time (within the range of statistical oscillations), so that the time derivative can be elided. In addition, one wants the chain reaction to self-sustain without the need for external sources. The resulting homogeneous stationary equation can then be reformulated as an eigenvalue problem by introducing the k_{eff} parameter, called *effective multiplication factor*, which can be seen as the number of neutrons produced by fission and inducing new fission events or, analogously, the ratio between the numbers of neutrons emerging from fission of two successive *generations*, where the birth and the death of a generation of neutrons are associated to their emission by fission and their absorption, respectively. Based on these considerations, the eigenvalue formulation of Eq. (2.6) reads

$$\begin{aligned}\left[\vec{\Omega} \cdot \nabla + \Sigma_T(\vec{r}, E)\right] \psi(\vec{r}, \vec{\Omega}, E) = &\int_0^\infty dE' \oint d\vec{\Omega}'\, \Sigma_S(\vec{r}, \vec{\Omega} \cdot \vec{\Omega}', E' \to E)\, \psi(\vec{r}, \vec{\Omega}', E') \\ &+ \frac{1}{k_{\text{eff}}} \sum_{i=1}^{N_f} \frac{\chi_i(E)}{4\pi} \int_0^\infty dE'\, \nu\Sigma_{F,i}(\vec{r}, E')\, \phi(\vec{r}, E').\end{aligned} \tag{2.9}$$

Its expression can be further simplified by omitting variable dependencies and introducing an operator notation:

$$\mathcal{L}\, \psi = \mathcal{H}\, \psi + \frac{1}{k_{\text{eff}}} \mathcal{F}\, \phi, \tag{2.10}$$

$\mathcal{L}* = (\vec{\Omega} \cdot \nabla + \Sigma_T)*$ being the transport operator (streaming + removal) and $\mathcal{H}*$ and $\mathcal{F}*$ the transfer and fission operators, respectively.

2.2.2 Boundary conditions

A complete presentation of the eigenvalue problem (2.10) requires the definition of spatial boundary conditions. Denoting by $\mathcal{D}$ the domain considered, the entering flux through the boundary $\partial\mathcal{D} = \sum_j \alpha_j$ can be set in different ways, depending on the kind of each α interface.

- If α is a convex border, like the actual boundary of the reactor core, one can suppose that the number of incoming neutrons be negligible with respect to the population within $\mathcal{D}$; hence, *vacuum* conditions are imposed:

$$\psi(\vec{r}_b, \vec{\Omega}, E) = 0 \qquad \vec{r}_b \in \alpha,\ \vec{\Omega} \cdot \hat{n}_\alpha < 0, \tag{2.11}$$

  $\hat{n}_\alpha$ being the unit vector normal to α and oriented towards the outside of $\mathcal{D}$.

- When symmetry allows considering a smaller portion of the initial domain, called *basic* domain, a G mapping of the $(\vec{r}, \vec{\Omega})$ subspace can be used to express the entering flux at $G[(\vec{r}_b, \vec{\Omega})] = \Big(G(\vec{r}_b), G(\vec{\Omega})\Big)$ in terms of the outgoing flux at $(\vec{r}_b, \vec{\Omega})$. For instance, in case of *reflectional* symmetry at interface α, one can write

$$G[(\vec{r}_b, \vec{\Omega})] = \Big(\vec{r}_b, \vec{\Omega} - 2(\vec{\Omega} \cdot \hat{n}_\alpha)\hat{n}_\alpha\Big). \tag{2.12}$$

  Other examples of symmetry relations include *translational* and *rotational* conditions. The total domain is then obtained by repeatedly applying the proper geometrical movement, depending on the symmetry kind, to the basic domain.

2.3 Numerical discretization of the phase space

In the absence of an analytical solution in the general case, the transport equation requires numerical methods to determine the k_{eff} eigenvalue and the corresponding flux eigenfunction. In order to fit in deterministic solver, the phase space continuous dependence is thus substituted by discrete approximations of the space, angle and energy variables.

2.3.1 Spatial approximation

A preliminary and unavoidable step to perform a numerical simulation is the translation of the physical system into its numerical *equivalent*. This entails the subdivision of the domain into a set of meshes, identifying the computational regions $\{r\}$, which allows for different spatial descriptions within the region considered. Depending on the mesh refinement and the chosen descriptions, in fact, the obtained numerical domain will be more or less faithful to the original one. A general quantity q can therefore be written, within region r, as

$$q(\vec{r}) = \sum_{i=1}^{N_e(r)} q_{r,i}\, f_{r,i}(\vec{r}) = \vec{q}_r \cdot \vec{f}_r(\vec{r}), \tag{2.13}$$

with $\vec{f}_r = \{f_{r,i}\}$ being the set of N_e functions adopted as basis for the spatial expansion. As the notation suggests, $\vec{f}$ and N_e may in general depend on r. The simplest possible choice of expansion functions constitutes the so-called *step-constant* approximation, according to which quantities take piece-wise constant values within each region:

$$\vec{f}_r(\vec{r}) = f_r(\vec{r}) = \begin{cases} 1 & \vec{r} \in r \\ 0 & \text{otherwise} \end{cases} \qquad \forall r. \tag{2.14}$$

As the spatial gradients become more severe, this strategy requires a more and more refined mesh, with a direct impact on performances. For this reason higher-order methods are proposed, like the one discussed in this thesis, which limit the number of regions by using more complex definitions of the expansion functions.

2.3.2 Angular expansion

In order to convert the integral over directions of Eq.(2.9) into a form suitable for calculation, the transfer term $\mathcal{H}\psi$ needs be reformulated by introducing the expansion over real spherical harmonics $\{A_l^m\}$. To do that, the dependence on $\mu = \vec{\Omega} \cdot \vec{\Omega}'$ of the differential scattering cross section Σ_S is firstly expressed in terms of Legendre polynomials $\{\mathcal{L}_l\}$, defined as

$$\mathcal{L}_l(\mu) = \frac{1}{2^l\, l!} \frac{d^l(\mu^2-1)^l}{d\mu^l} \qquad l = 0, 1, 2, \ldots \tag{2.15}$$

Adopting the following definition for the cross section moments:

$$\Sigma_{S,l}(\vec{r}, E' \to E) = 2\pi\,(2l+1) \int_{-1}^{1} d\mu\, \Sigma_S(\vec{r}, \mu, E' \to E)\, \mathcal{L}_l(\mu) \tag{2.16}$$

and truncating the expansion to the $l = K$ anisotropy order, the approximation reads

$$\Sigma_S(\vec{r}, \mu, E' \to E) \approx \frac{1}{4\pi} \sum_{l=0}^{K} \Sigma_{S,l}(\vec{r}, E' \to E)\, \mathcal{L}_l(\mu). \tag{2.17}$$

The expression of the A_l^m real spherical harmonics is

$$A_l^m(\vec{\Omega}) = \begin{cases} \sqrt{(2-\delta_{m0})\frac{(l-m)!}{(l+m)!}}\, \mathcal{L}_l^m(\cos\theta)\, \cos(m\varphi) & m \geq 0 \\ \sqrt{2\,\frac{(l-|m|)!}{(l+|m|)!}}\, \mathcal{L}_l^{|m|}(\cos\theta)\, \sin(|m|\varphi) & m < 0, \end{cases} \tag{2.18}$$

with θ and φ being the polar and azimuthal components of $\vec{\Omega}$, respectively, and $\mathcal{L}_l^m(\mu) = (1-\mu^2)^{m/2}\, \frac{d^m \mathcal{L}_l(\mu)}{d\mu^m}$ the associated Legendre function. Since $\mathcal{L}_l(\vec{\Omega}\cdot\vec{\Omega}') = \sum_{m=-l}^{l} A_l^m(\vec{\Omega}) A_l^m(\vec{\Omega}')$, the transfer term becomes

$$\begin{aligned} \mathcal{H}\psi &\approx \int_0^\infty dE' \oint \frac{d\vec{\Omega}'}{4\pi} \sum_{l=0}^{K} \Sigma_{S,l}(\vec{r}, E' \to E) \sum_{m=-l}^{l} A_l^m(\vec{\Omega})\, A_l^m(\vec{\Omega}')\, \psi(\vec{r}, \vec{\Omega}', E') \\ &= \int_0^\infty dE' \sum_{l=0}^{K} \Sigma_{S,l}(\vec{r}, E' \to E) \sum_{m=-l}^{l} A_l^m(\vec{\Omega})\, \Phi_l^m(\vec{r}, E') \\ &= \sum_{n=1}^{N_m} A_n(\vec{\Omega}) \int_0^\infty dE'\, \Sigma_{S,l(n)}(\vec{r}, E' \to E)\, \Phi_n(\vec{r}, E'), \end{aligned} \tag{2.19}$$

where two different numberings are used for the harmonic indices:

$$\sum_{l=0}^{K} \sum_{m=-l}^{l} A_l^m(\vec{\Omega}) = \sum_{n=1}^{N_m} A_n(\vec{\Omega}), \tag{2.20}$$

with $N_m = (K+1)^2$. In the following, either the extended or the compact version will be used, based on convenience. For the record, the latter is the one used in the code, to assign a single array dimension to the angular moments. In Eq. (2.19), $l(n)$ represents the anisotropy order corresponding to the n^{th} index and the moments $\{\Phi_l^m\}$ of the angular flux are introduced, defined as

$$\Phi_l^m(\vec{r}, E) = \oint \frac{d\vec{\Omega}}{4\pi} A_l^m(\vec{\Omega})\, \psi(\vec{r}, \vec{\Omega}, E). \tag{2.21}$$

Since $A_0^0 = 1$, the scalar flux can be expressed as $\phi = 4\pi\Phi_0^0 = 4\pi\Phi_1$; hence, the fission term is slightly modified to take into account the flux angular expansion:

$$\mathcal{F}\phi = \sum_{i=1}^{N_f} \chi_i(E) \int_0^\infty dE'\, \nu\Sigma_{F,i}(\vec{r}, E')\, \Phi_0^0(\vec{r}, E'). \tag{2.22}$$

A further approximation relating to the angular variable and adopted in the present work concerns the way integrals over directions are calculated: as it is customary in the MOC solvers (see chapter 4), the S_N approximation is used to discretize the angular continuum into a set of N directions, which are then treated as quadrature points. For example, the flux moments (2.21) can be evaluated, according to this approach, as

$$\Phi_l^m(\vec{r}, E) \approx \sum_{k=1}^{N} w(\vec{\Omega}_k)\, A_l^m(\vec{\Omega}_k)\, \psi(\vec{r}, \vec{\Omega}_k, E), \tag{2.23}$$

$w(\vec{\Omega}_k)$ being the weight associated to the k^{th} direction for the chosen quadrature formula. In the following it is assumed that all angular integrals be computed by quadrature rule, even if the analytic notation will still be used.

2.3.3 Multi-group approximation

The remaining variable to discuss is energy. Deterministic methods traditionally subdivide the energy domain into discrete subintervals and, within each of them, consider a *constant* multi-group expression of the angular flux and its moments:

$$\psi^g(\vec{r}, \vec{\Omega}) = \int_{E_g}^{E_{g-1}} dE\, \psi(\vec{r}, \vec{\Omega}, E), \qquad \Phi_n^g(\vec{r}) = \int_{E_g}^{E_{g-1}} dE\, \Phi_n(\vec{r}, E), \tag{2.24}$$

where g denotes the group index and E_g and E_{g-1} the group boundaries, following the convention that counts the groups starting from the highest energy. In order to obtain a multi-group formulation of the transport problem, a condensation step is required to express the multi-group macroscopic cross sections. Imposing the conservation of in-group reaction

rates, these are defined as

$$\begin{aligned}
\Sigma_T^g(\vec{r},\vec{\Omega}) &= \frac{\displaystyle\int_{E_g}^{E_{g-1}} dE\,\Sigma_T(\vec{r},E)\,\psi(\vec{r},\vec{\Omega},E)}{\psi^g(\vec{r},\vec{\Omega})} \\
\Sigma_{S,l(n),n}^{g'\to g}(\vec{r}) &= \frac{\displaystyle\int_{E_g}^{E_{g-1}} dE \int_{E_{g'}}^{E_{g'-1}} dE'\,\Sigma_{S,l(n)}(\vec{r},E'\to E)\,\Phi_n(\vec{r},E')}{\Phi_n^{g'}(\vec{r})} \\
\nu\Sigma_{F,i}^{g'}(\vec{r}) &= \frac{\displaystyle\int_{E_{g'}}^{E_{g'-1}} dE'\,\nu\Sigma_{F,i}(\vec{r},E')\,\Phi_1(\vec{r},E')}{\Phi_1^{g'}(\vec{r})}.
\end{aligned} \tag{2.25}$$

In this phase the flux acts as a weight function for cross sections, causing the appearance of an angular dependence in the total and scattering cross sections (in the latter, this is due to the weighting role of the n^{th} angular flux moment). However, this dependence is neglected in the applications, because the weight function, which is the actual unknown of the problem, is normally substituted by a lower-order estimate (typically the scalar flux, like in APOLLO3®). The details relating to the evaluation of the condensed cross sections, resorting to the self-shielding formalism for the resonance domain, will be discussed in section 3.1.

Integrating Eq. (2.9) over the energy range of the generic g group, taking into account the expansion over spherical harmonics of the transfer and fission terms and introducing the multi-group cross sections just defined, one obtains

$$\begin{aligned}
\left(\vec{\Omega}\cdot\nabla+\Sigma_T^g(\vec{r})\right)\psi^g(\vec{r},\vec{\Omega}) = &\sum_{n=1}^{N_m} A_n(\vec{\Omega}) \sum_{g'=1}^{N_g} \Sigma_{S,l(n)}^{g'\to g}(\vec{r})\,\Phi_n^{g'}(\vec{r}) + \\
&+ \frac{1}{k_{\text{eff}}} \sum_{i=1}^{N_f} \chi_i^g \sum_{g'=1}^{N_g} \nu\Sigma_{F,i}^{g'}(\vec{r})\,\Phi_1^{g'}(\vec{r}),
\end{aligned} \tag{2.26}$$

with $\chi_i^g = \int_{E_g}^{E_{g-1}} dE\,\chi_i(E)$. In the next chapters, this equation will be the starting point for the description of MOC step-constant and higher-order approaches. The energy group index will usually be omitted, unless necessary for understanding, assuming that equations refer to a generic group.

Chapter 3

Transport solution in the APOLLO3® code

The APOLLO3® French deterministic code contains a set of solvers of the neutron transport problem. The applications range from the production of energy-condensed cross sections, starting from nuclear data and models, to full core calculations. At present, the latter is still preceded by a preliminary phase, which computes the spatially homogenized cross sections required, at a larger scale, in the following step.

This chapter aims to provide a global view of the numerical strategy followed to attain the transport solution. The self-shielding formalism, responsible for the proper evaluation of multi-group cross sections, is presented, as well as the main related methods; the contents are taken from the thesis work of M. Coste-Delclaux [18]. Then, the two-step calculation is described, highlighting the effort that are being made to render the single-step approach feasible. The discussion will also consider the usage of the power-iteration method [19] in neutron transport, before lastly focusing on the coupling scheme adopted for depletion calculations.

The two method of characteristics solvers available in the code, TDT and IDT, will be the object of the next chapter, entirely dedicated to the widely used method they are based on.

3.1 The self-shielding formalism

As anticipated in section 2.3.3, the multi-group cross section expression demands a specific treatment because of its dependence on the neutron flux, which is unknown. Formally, the total and differential scattering cross sections would need be weighted by the angular flux, but it is common practice to neglect this dependence in favor of the scalar flux. Won and Cho's work [21] represents a notable exception: starting from a fine energy mesh, condensation leads to Eqs. (2.25) - like cross sections, which are evaluated and used maintaining their angular dependence. In the most widespread approach, instead, the microscopic cross section for

reaction R, isotope i and energy group g is written as

$$\sigma_{R,i}^{g} = \frac{\int_{E_g}^{E_{g-1}} dE\, \sigma_{R,i}(E)\, \phi(\vec{r}, E)}{\phi^{g}(\vec{r})}, \tag{3.1}$$

with $\phi^{g}(\vec{r}) = \int_{E_g}^{E_{g-1}} dE\, \phi(\vec{r}, E)$. The scalar flux, which is also unknown, is then substituted by a neutron energy spectrum ϕ_w, representative of the group range: a Maxwellian spectrum in the thermal domain, a slowing-down spectrum ($\phi_w \propto \frac{1}{E}$) in the epithermal range and a fission spectrum for higher energies. By $\sigma_{R,i}(E)$ one denotes the *pointwise* cross sections that are obtained by conversion of the available ENDF (Evaluated Nuclear Data File) [22] to PENDF (Pointwise ENDF) format [23]. Several computing modules, all part of the NJOY code [24], are in charge of this step, which involves data interpolation and Doppler broadening at the chosen temperature.

In this framework, the lethargy variable $u = ln\left(\frac{E_0}{E}\right)$ is commonly adopted instead of energy, with E_0 being a reference arbitrary value, usually greater or equal to the maximum energy considered so that u is always positive. The previous PENDF to GENDF conversion (where "G" stands for "Groupwise") becomes

$$\sigma_{R,i}^{g} \approx \frac{\int_g du\, \sigma_{R,i}(u)\, \phi_w(u)}{\phi_w^g}. \tag{3.2}$$

The multi-group microscopic cross sections thus obtained are called of *infinite-dilution* kind, because the weight function adopted can be obtained from Boltzmann equation under the hypothesis, among others, that isotope i is diluted with negligible concentration in a solution of moderating isotopes. Based on their expression, these quantities do not depend on the geometry of the problem and can therefore be evaluated once and for all and made available in the GENDF format.

However, the flux used as weight in Eq. (3.2) does not take into account the effect that resonances have on the actual flux, which sinks in correspondence of the energies relative to the isotope resonances (Fig. 3.1); moreover, this effect depends on the spatial concentrations of resonant isotopes. It follows that adopting infinite-dilution multi-group cross sections in the resonance domain and in computational regions containing resonant isotopes can lead to a dramatic overestimation of reaction rates, which results in a completely wrong k_{eff} (differing by tens of percents from the acceptable value) [25].

The phenomenon whereby, in proximity of a resonance, the strong and opposite variations of flux and cross section compensate, producing a smoother reaction rate profile, is called self-shielding. This identifies also the procedure leading to the correct evaluation of cross sections within resonance groups. In the following, the two main methodologies used in APOLLO3® are briefly outlined; the choice between them depends directly on the available energy mesh.

3.1.1 Livolant-Jeanpierre method

In order to produce multi-group cross sections that take into account the self-shielding phenomenon, the algorithms adopt a more accurate representation of the in-group flux, which

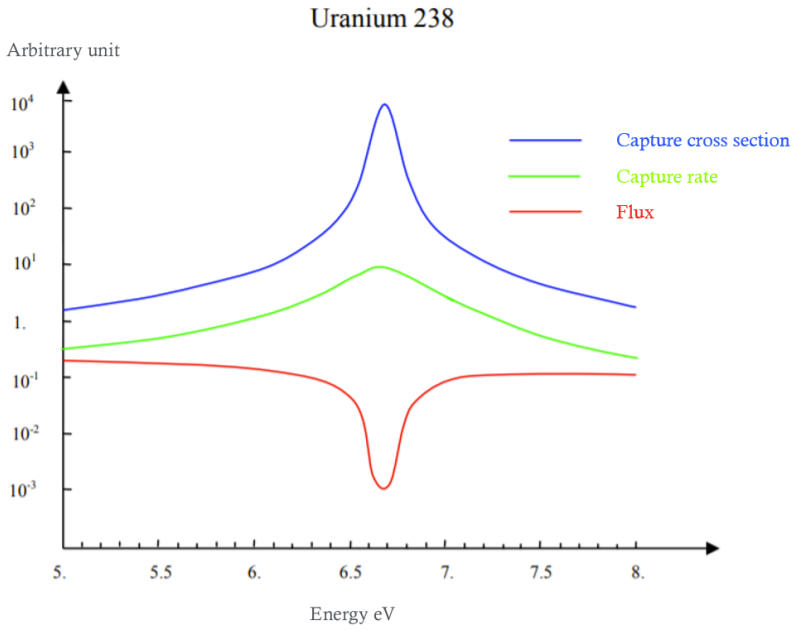


Figure 3.1: Self-shielding phenomenon in correspondence of the first resolved resonance of U-238 (image modified from the original in [18]). The four-decade increase of the capture cross section entails a three-decade sinking for the flux which, if neglected, leads to overestimate the capture rate variation.

is solution of a slowing-down problem under the hypotheses of pure elastic and isotropic scattering. The Livolant-Jeanpierre formalism is based on a homogeneous-heterogeneous equivalence: the slowing-down equations for the two cases are now presented.

For an isotropic infinite homogeneous medium, the time independent transport equation (2.9), written in the lethargy variable and integrated over directions, reads

$$\Sigma_T(u)\,\phi(u) = \int_0^{+\infty} du'\,\Sigma_S(u' \to u)\,\phi(u'), \tag{3.3}$$

where $\Sigma_S(u' \to u) = 4\pi\,\Sigma_S(\vec{\Omega}\cdot\vec{\Omega}', u' \to u)\ \forall\,\vec{\Omega},\vec{\Omega}'$. The fission term does not appear in the equation, as its contribution to neutron emission in the resonance domain is negligible. For the sake of simplicity, the present discussion is limited to the analysis of a single resonant isotope at a time, but the method was extended to the study of isotope mixtures, to take into account the effect of overlapping resonances. Denoting by index "0" the quantities relating to the resonant isotope considered and by "1" the background mixture of moderators and limiting the energy domain to a group wherein isotope "0" is resonant, Eq. (3.3) becomes

$$\begin{aligned}\Big(\Sigma_{T,0}(u) + \Sigma_{T,1}(u)\Big)\,\phi(u) &= \int_{u-\epsilon_0}^{u} du'\,\Sigma_{S,0}(u' \to u)\,\phi(u') + \int_{u-\epsilon_1}^{u} du'\,\Sigma_{S,1}(u' \to u)\,\phi(u') \\ &= \mathcal{R}_0\,\phi(u) + \mathcal{R}_1\,\phi(u),\end{aligned} \tag{3.4}$$

where the slowing-down operators $\mathcal{R}_0$ and $\mathcal{R}_1$ have been implicitly defined. ϵ_0 and ϵ_1 represent the maximum lethargy gains of neutrons due to the scattering with the respective

isotopes; since the process has been assumed to be elastic, the following relations hold

$$\epsilon_i = ln\left(\frac{1}{\alpha_i}\right), \qquad \alpha_i = \left(\frac{A_i - 1}{A_i + 1}\right)^2, \qquad A_i = \frac{M_i}{m_n}, \tag{3.5}$$

with M_i and m_n being the isotope and the neutron mass, respectively. For isotopes "1" the greatest ϵ is considered. At this point, one can introduce the fine-structure factorization of the flux which characterizes the method: $\phi(u) = \chi(u)\,\varphi(u)$, where χ is the *macroscopic* flux defined as

$$\chi(u) = \frac{\mathcal{R}_1\,\phi(u)}{\Sigma_{S,1}(u)}, \tag{3.6}$$

whereas φ is the *fine-structure* factor. This latter is expected to account for rapid changes with lethargy, while χ is supposed to be quite regular. The important assumption is made to neglect the weak variation of the macroscopic flux within the lethargy interval, which is all the more justified as the resonant isotope is heavier (ϵ_0 smaller) and, in parallel, as the moderators are lighter (ϵ_1 larger). The resulting fine-structure approximation reads $\mathcal{R}_0\,\phi(u) \approx \chi(u)\,\mathcal{R}_0\,\varphi(u)$.

Dividing Eq. (3.4) by the concentration of the resonant isotope, N_0, the fine-structure formulation of the slowing-down problem is finally written as

$$\Big(\sigma_{T,0}(u) + \sigma_b\Big)\,\varphi(u) = r_0\,\varphi(u) + \gamma_b\,\sigma_b, \tag{3.7}$$

where

$$\sigma_b(u) \approx \sigma_b = \frac{\Sigma_{T,1}(u)}{N_0}, \qquad r_0* = \frac{\mathcal{R}_0*}{N_0}, \qquad \gamma_b(u) \approx \gamma_b = \frac{\Sigma_{S,1}(u)}{\Sigma_{T,1}(u)}. \tag{3.8}$$

As denoted above, a further assumption adopted consists in neglecting the lethargy dependence of the dilution cross section σ_b and of the γ_b ratio.

Considering now a heterogeneous geometry, composed of several homogeneous regions, one can resort to the collision-probability version [17] of the slowing-down problem as a formula for the uniform ϕ_j flux within the j^{th} region:

$$V_j\,\Sigma_{T,j}(u)\,\phi_j(u) = \sum_k V_k\,P_{jk}(u)\,\Big[\mathcal{R}_{0,k}\,\phi_k(u) + \mathcal{R}_{1,k}\,\phi_k(u)\Big]. \tag{3.9}$$

Here, $\{V_k\}$ are the region volumes and P_{jk} indicates the probability that a neutron born in region k undergoes its first collision in region j. As for the homogeneous case, the fine-structure representation is introduced, and regions $\{j\}$ are grouped into self-shielding regions $\{\alpha\}$ containing the same resonant isotope at the same temperature and thus sharing the same microscopic cross sections and slowing-down operators. Also, a common fine-structure factor is adopted within each self-shielding region. Eq. (3.9) therefore becomes

$$\sum_{j\in\alpha} V_j\,\Sigma_{T,j}(u)\,\chi_j(u)\,\varphi_\alpha(u) = \sum_{j\in\alpha}\sum_k V_k\,P_{jk}(u)\,\chi_k(u)\,\Big[\mathcal{R}_{0,k}\,\varphi_k(u) + \Sigma_{S,1,k}(u)\Big]. \tag{3.10}$$

The equation just obtained can be further simplified by neglecting the region dependence of the macroscopic flux χ and taking into account that the contribution of $\mathcal{R}_0\,\varphi$ is null outside the regions containing resonant isotopes:

$$\sum_{j\in\alpha} V_j\,\Sigma_{T,j}(u)\,\varphi_\alpha(u) = \sum_{j\in\alpha}\sum_\beta\sum_{k\in\beta} V_k\,P_{jk}(u) N_{0,k}\,r_{0,\beta}\,\varphi_\beta(u) + \sum_{j\in\alpha}\sum_k V_k\,P_{jk}(u)\,\Sigma_{S,1,k}(u), \tag{3.11}$$

which is compacted into

$$\varphi_\alpha(u) = \sum_\beta \mathcal{C}_{\alpha\beta}(u)\, r_{0,\beta}\, \varphi_\beta(u) + \mathcal{S}_\alpha(u) \tag{3.12}$$

by defining

$$\mathcal{C}_{\alpha\beta}(u) = \frac{\sum_{j\in\alpha}\sum_{k\in\beta} V_k\, P_{jk}(u) N_{0,k}}{\sum_{j\in\alpha} V_j\, \Sigma_{T,j}(u)}, \qquad \mathcal{S}_\alpha(u) = \frac{\sum_{j\in\alpha}\sum_k V_k\, P_{jk}(u)\, \Sigma_{S,1,k}(u)}{\sum_{j\in\alpha} V_j\, \Sigma_{T,j}(u)}. \tag{3.13}$$

In doing so one obtains the fine-structure equations (3.7) and (3.12) relating to the homogeneous and heterogeneous geometries, respectively. In both cases, a relation for the fine-structure factor is derived by approximating the expression of the slowing-down term $r_0\,\varphi$ by means of one of the available models (Narrow Resonances NR, Wide Resonances WR, Statistic ST and All Resonances AR). The heterogeneous factor obtained, φ^*_α, is then used to compute an approximation of the reaction rate at the numerator of Eq. (3.1), by a suitable quadrature formula. Here, " * " indicates that φ^*_α is obtained through Eq. (3.12) by approximating $r_0\,\varphi$. For group g, reaction R, isotope i and self-shielding region α, this quantity is given by

$$T^{g*}_{R,i,\alpha} = \int_g du\, \sigma_{R,i}(u)\, \varphi^*_\alpha(u). \tag{3.14}$$

It is worth noting that the microscopic cross section is weighted by the fine-structure factor instead of the actual flux. The homogeneous counterpart $T^{g*}_{R,i,\mathrm{hom}}$ is instead expressed as a function of the dilution cross section σ_b. By setting the two terms equal for a chosen reaction, one can retrieve the value of σ_b (homogenization phase). The corresponding exact value, $T^g_{R,i,\mathrm{hom}}(\sigma_b)$, is then found through interpolation of the infinite homogeneous medium reaction rates in the available tabulations, for the obtained dilution cross section, and set equal to the exact heterogeneous rate $T^g_{R,i,\alpha}$:

$$T^g_{R,i,\alpha} = T^g_{R,i,\mathrm{hom}}(\sigma_b). \tag{3.15}$$

The self-shielded version of Eq. (3.2) is therefore

$$\sigma^g_{R,i,\alpha} = \frac{T^g_{R,i,\alpha}}{\varphi^g_\alpha}, \tag{3.16}$$

where $\varphi^g_\alpha = \int_g du\, \varphi_\alpha(u)$ is obtained through integration over the energy group of the heterogeneous fine-structure equation (3.12), exploiting the already computed scattering reaction rates in the derivation. Finally, the general multi-group self-shielded cross section for group g and region j, under the relaxable hypothesis of a single resonant isotope, reads

$$\Sigma^g_{R,j} = N_{0,j}\, \sigma^g_{R,i,\alpha} + \Sigma^g_{R,1,j}, \tag{3.17}$$

where $\Sigma^g_{R,1,j} = \int_g du\, \Sigma_{R,1,j}(u)$, the multi-group moderator cross section for reaction R, is also evaluated through quadrature. Due to the dependence of φ_α on the fine-structure factors of all the other self-shielding regions (Eq. (3.12)), the problem is solved through an iterative scheme starting from a flat φ in all regions and updating terms (3.13) with the newly obtained macroscopic total cross sections at each iterate. However, repeating the loop more than once is generally unnecessary, since practical experience has shown that the eigenvalue would change by no more than a few *pcm* [26].

3.1.2 Subgroups method

An alternative to the approach just described is the subgroups method, based on the following property: if the group discretization is fine enough, one can approximate the average of the product of two quantities over a group as the product of their averages. This method is therefore particularly suited to very refined energy meshes, for which it is reasonable to assume that the resonances of different isotopes do not overlap and thus it is acceptable to treat one resonant nuclide at a time. The basics of the subgroups method are presented below, according to its implementation within the predecessor of APOLLO3®, APOLLO2 [27].

Introducing the collision densities $\{\tau_k\}$ in the right-hand side of the slowing-down equation (3.9),

$$\begin{aligned} V_j\,\Sigma_{T,j}(u)\,\phi_j(u) &= \sum_k V_k\,P_{jk}(u)\left[\mathcal{R}_{0,k}\,\phi_k(u) + \mathcal{R}_{1,k}\,\phi_k(u) + S_k(u)\right] \\ &= \sum_k V_k\,P_{jk}(u)\,\tau_k(u), \end{aligned} \tag{3.18}$$

with S_k denoting the external source for region k, and defining the group average of quantity f as

$$\langle f\rangle^g = \frac{1}{\Delta u^g}\int_g du\, f(u), \tag{3.19}$$

where $\Delta u^g = u_{g+1} - u_g$, the averages of the flux and of the rate for reaction R and isotope i read

$$\langle\phi_j\rangle^g = \sum_k \frac{V_k}{V_j}\,\langle\frac{P_{jk}}{\Sigma_{T,j}}\,\tau_k\rangle^g, \qquad \langle\sigma_{R,i}\,\phi_j\rangle^g = \sum_k \frac{V_k}{V_j}\,\langle\frac{\sigma_{R,i}\,P_{jk}}{\Sigma_{T,j}}\,\tau_k\rangle^g. \tag{3.20}$$

The product property mentioned above allows expressing the multi-group cross section (Eq. (3.1)) as

$$\sigma^g_{R,i,j} = \frac{\langle\sigma_{R,i}\,\phi_j\rangle^g}{\langle\phi_j\rangle^g} \approx \frac{\sum_k \frac{V_k}{V_j}\,\langle\frac{\sigma_{R,i}\,P_{jk}}{\Sigma_{T,j}}\rangle^g\,\langle\tau_k\rangle^g}{\sum_k \frac{V_k}{V_j}\,\langle\frac{P_{jk}}{\Sigma_{T,j}}\rangle^g\,\langle\tau_k\rangle^g}, \tag{3.21}$$

where $\langle\frac{P_{jk}}{\Sigma_{T,j}}\rangle^g$ and $\langle\frac{\sigma_{R,i}\,P_{jk}}{\Sigma_{T,j}}\rangle^g$ can be computed by the same kinds of quadrature formula that are adopted in Livolant-Jeanpierre formalism, so that the group averages of collision densities $\{\langle\tau_k\rangle^g\}$ are the only unknowns of the problem. In fact, the approximation introduced in Eq. (3.21) is valid also for a less refined energy mesh in the unresolved resonances domain, where cross section resonances are not statistically correlated with collision densities; this is due to the fact that, in this range, the average lethargic gain is big with respect to the resonance width. However, since the self-shielding phase is required in the whole resonance domain, adopting a fine mesh represents the main requirement of the method.

Based on the implicit definition of collision densities in Eq. (3.18), $\langle\tau_j\rangle^g$ can be expressed in the following form

$$\langle\tau_j\rangle^g = \sum_{g'\le g}\sum_k g_{0,jk}\Big(\langle\tau_k\rangle^{g'}\Big) + g_{1,jk}\Big(\langle\tau_k\rangle^{g'}\Big) + \langle S_j\rangle^g, \tag{3.22}$$

where g_0 and g_1 result from the reformulation of the slowing-down terms of corresponding index and are two functions of the collision density averages. The previous relation is obtained

approximating the scalar and homogeneous version of the scattering function f_S (Eq. (2.7)) for the resonant nuclide and the differential scattering cross section for the moderators by their infinite-dilution values (available in tabulations). The average of the external source is assumed equal to 1 for the first group and to 0 for the others. Taking into account all groups, Eq. (3.22) constitutes a triangular system in energy: while computing $\langle \tau_j \rangle^g$, $\langle \tau_j \rangle^{g'}$ is known for all $g' < g$.

APOLLO3® currently contains only a *hybrid* technique, which applies the subgroups procedure to the fine-structure equation (3.12). This approach, although accurate for mixtures of heavy and light isotopes, may produce less good results when the presence of intermediate isotopes is more relevant (as in the reflector, see section 7.2.2), because the fine-structure hypothesis is hardly satisfied in such a case.

3.1.3 Remarks

The discussed methods represent two possible ways of approaching the self-shielding problem. Without entering too much in detail, the Livolant-Jeanpierre technique introduces greater approximations, particularly regarding the fine-structure treatment of the slowing-down term. On the other hand, the subgroups method deals with the "real" scalar flux (constant within each region), but requires a finer energy discretization to be consistent. This is the main reason why the Livolant-Jeanpierre approach is adopted for the case studies of this thesis, since an energy mesh of "only" 281 groups is used for all the PWR subsystems considered.

Another important aspect concerns the application of the self-shielding model to 3D geometries: as shown in D. Sciannandrone's thesis work [2], at least for a small 3D subsystem of the ASTRID reactor project [14] and while applying the subgroups method, introducing further self-shielding regions at the axial interfaces between two 3D computational regions does not have a significant impact on result accuracy. Hence, the choice has been made to neglect the axial interdependence of multi-group cross sections of neighboring axial regions in this work, so that each axial layer is treated separately by the self-shielding module.

Finally, the higher-order spatial description that this thesis extends to macroscopic cross sections does not involve the self-shielding procedure. The fine-structure flux and the infinite-dilution multi-group cross sections, as well as the resulting self-shielded ones, remain constant within each computational region. An ad-hoc procedure is thus demanded each time the flux and the self-shielding solvers have to interact: more specifically, the step-constant shielded cross sections need be converted into polynomial at the beginning of each flux calculation and, if more than one depletion step is to be performed, the computed flux is then reconverted into step-constant format and used as input for the depletion solver (see section 3.4), before the next application of the self-shielding method. This strategy implies the need to have two spatial meshes, as will be detailed in paragraph 5.4.

3.2 Two-step and one-step calculations

Due to the relevant computational resources required, simulations concerning significant portions of the reactor are commonly divided into two steps. Firstly, a *lattice* calculation analyzes a small, infinitely repeated domain which should be as representative as possible of

the properties of the overall system. This is equivalent to say that the total geometry could be obtained by repetition of the subdomain and that its global physical behavior should be well described based on the lattice results, at least far from the system edges.

Reflective conditions are therefore set at the boundaries of the lattice domain. Then, the flux calculation allows evaluating reaction rates, which are to be conserved during the successive spatial homogenization phase. A condensation of the energy groups is also conceivable in this step. The obtained spatially homogenized and condensed in energy equivalent cross sections are used in the following *core* calculation. This is the traditional procedure employed by nuclear industry: performing a lattice simulation as a preliminary, necessary step for the actual calculation at larger scale.

What said above is valid for both 2D and 3D geometries, the latter being the object of this work. The case studies that will be described represent relatively small portions of PWR reactors, which suggests that the developments of this thesis mainly concern lattice calculations; however, it is also true that considering axially extruded geometries and adopting higher-order spatial descriptions allow treating cases with exact axial boundary conditions nowadays and, as the available computational resources improve, simulating increasingly larger systems. Ideally, therefore, the neat distinction between lattice and core calculation should gradually fade due to the combined effect of better calculation tools and more refined numerical methods.

The polynomial treatment of cross sections proposed here aspires to contribute in this perspective too: as will be explained in section 5.4 and shown in 7.4, the present work actually makes the depletion study of a full PWR assembly far more affordable, in terms of computing time and memory usage, with respect to the pre-existing step-constant approach.

3.3 The power method in neutron transport

The customary deterministic way to solve the steady-state neutron transport eigenvalue problem is the power-iteration method [19]. Considering the compacted expression (2.10), one aims to find the fundamental eigenfunction ψ_∞ corresponding to the greatest eigenvalue (k_{eff}) of the following operator:

$$\mathcal{M}* = \Big(\mathcal{L} - \mathcal{H}\Big)^{-1} \tilde{\mathcal{F}}*, \tag{3.23}$$

with $\tilde{\mathcal{F}}* = \oint d\vec{\Omega}\, \mathcal{F}*$. The eigenvalue problem can therefore be written as

$$\mathcal{M}\, \psi_\infty = k_{\text{eff}}\, \psi_\infty. \tag{3.24}$$

The effective multiplication factor is evaluated by an iterative scheme that fixes the fission emission density at the beginning of each power iteration, computes the new flux iterate and then updates the eigenvalue iterate. In a multi-group framework, these iterations are usually called *outer*, to distinguish them from the ones regarding the in-group flux convergence. Introducing the iteration index o, the resolution strategy of Eq. (3.24) reads, for any energy group,

$$\Big(\mathcal{L} - \mathcal{H}\Big)\psi^o = \frac{1}{k_{\text{eff}}^{o-1}} \mathcal{F}\phi^{o-1}, \tag{3.25}$$

consistently with [20], and the eigenvalue is updated by

$$k_{\text{eff}}^{o} = \frac{\langle\, \mathcal{F}\phi^{o} \,\rangle}{\frac{1}{k_{\text{eff}}^{o-1}} \langle\, \mathcal{F}\phi^{o-1} \,\rangle} = \frac{\sum_g \int_{\mathcal{D}} d\vec{r} \left(\mathcal{F}\phi^{o}\right)^g}{\frac{1}{k_{\text{eff}}^{o-1}} \sum_g \int_{\mathcal{D}} d\vec{r} \left(\mathcal{F}\phi^{o-1}\right)^g}, \tag{3.26}$$

with $\left(\mathcal{F}\phi^{o}\right)^g = \sum_i \chi_i^g \sum_{g'} \nu\Sigma_{F,i}^{g'}(\vec{r})\, \Phi_1^{o,g'}(\vec{r})$. The update condition makes the normalized integral of the fission emission density $\frac{\langle \mathcal{F}\phi \rangle}{k_{\text{eff}}}$ independent of o, so that the flux will also converge to a value independent of the iteration index: convergence requirements actually concern both k_{eff} and $\mathcal{F}\phi$ of all regions.

Each outer iteration requires the flux to converge within every energy group. As the fission source is fixed, groups are coupled by means of the transfer operator. If one could neglect the up-scattering, groups would be simply solved in sequence, starting from the first one (recall that groups are numbered for decreasing energy values). While this hypothesis is legitimate at high energies, for a sufficiently large number of groups it becomes unacceptable in the thermal domain, where neutrons are close to the equilibrium with the medium and have a non-negligible probability of being scattered at an energy higher than the collision one. The up-scattering imposes the use of a further iterative cycle, the *thermal* iterations, to guarantee the convergence in all groups. This cycle involves only the thermal groups and adopts a Gauss-Seidel method to invert the non-triangular portion of the $\left(\mathcal{L}-\mathcal{H}\right)*$ operator.

Inner iterations are in charge of the in-group convergence of the flux: based on Eq. (3.25), one can write

$$\mathcal{L}\,\psi^{it} = \mathcal{H}_{g\to g}\,\psi^{it-1} + S, \tag{3.27}$$

where it is the inner iteration index, $\mathcal{H}_{g\to g}\,\psi$ the in-group scattering term, defined as

$$\mathcal{H}_{g\to g}\,\psi^{it} = \sum_n A_n(\vec{\Omega})\, \Sigma_{S,l(n)}^{g\to g}(\vec{r})\, \Phi_n^{g,it}(\vec{r}) \tag{3.28}$$

and S denotes the source external to the group:

$$\begin{aligned} S = \mathcal{H}_{g'<g}\,\psi^{th} + \mathcal{H}_{g'>g}\,\psi^{th-1} + \frac{1}{k_{\text{eff}}^{o-1}}\mathcal{F}\phi^{o-1} &= \sum_n A_n(\vec{\Omega}) \sum_{g'<g} \Sigma_{S,l(n)}^{g'\to g}(\vec{r})\, \Phi_n^{g',th}(\vec{r}) \\ &+ \sum_n A_n(\vec{\Omega}) \sum_{g'>g} \Sigma_{S,l(n)}^{g'\to g}(\vec{r})\, \Phi_n^{g',th-1}(\vec{r}) \\ &+ \frac{1}{k_{\text{eff}}^{o-1}} \sum_i \chi_i^g \sum_{g'} \nu\Sigma_{F,i}^{g'}(\vec{r})\, \Phi_1^{g',o-1}(\vec{r}). \end{aligned} \tag{3.29}$$

In the previous relation, th is the thermal iteration index and $\mathcal{H}_{g'<g}*$ and $\mathcal{H}_{g'>g}*$ represent the down-scattering and the up-scattering operators, respectively. The way problem (3.27) is solved depends on the chosen method: the MOC strategy will be discussed in details in chapter 4. The iterative scheme presented in this section is summarized in Algorithm 1, wherein the external source S is progressively updated for the group considered.

Algorithm 1 Power-iteration strategy for solving multi-group transport

Initialize k_{eff} and $\{\Phi_n^g\}$
while k_{eff} and $\mathcal{F}\phi$ are not converged **do**
    Outer iteration o
    while The flux is not converged in all groups **do**
        Thermal iteration th
        for $g = 1, N_g$ **do**
            $S = \mathcal{H}_{g'<g}\,\psi^{th} + \mathcal{H}_{g'>g}\,\psi^{th-1} + \frac{1}{k_{\text{eff}}^{o-1}}\mathcal{F}\phi^{o-1}$
            while The flux is not converged in group g **do**
                Inner iteration it
                $\mathcal{L}\,\psi^{it} = \mathcal{H}_{g\to g}\,\psi^{it-1} + S$
            end while
        end for
    end while
    Update the fission source: $\mathcal{F}\phi^o$
    $k_{\text{eff}}^o = \frac{\langle \mathcal{F}\phi^o \rangle}{\frac{1}{k_{\text{eff}}^{o-1}}\langle \mathcal{F}\phi^{o-1} \rangle}$
end while

3.4 Neutron transport - depletion coupling scheme

From a practical point of view, this thesis makes it possible to apply an already developed higher-order method in space to depleted systems, that is, non-negligible burnup cases wherein the spatial gradients of the neutron flux have significantly modified the cross section spatial profiles. An exact calculation of the system depletion in time would require solving a very stiff problem, which couples the time-dependent neutron transport equation (2.6) with the set of Bateman equations [28] ruling the decay and the transmutation of all the nuclides present in the system. Besides, thermal hydraulic feedbacks should also be taken into account.

The current infeasibility of the exact calculation imposes the adoption of a simplified strategy, addressing the problem stiffness by separating the neutron transport calculation from the depletion study and assigning them to specific solvers. However, this does not mean that the depletion-neutronics coupling involves strong approximations: for the typical power levels of a working reactor, the variations in isotope concentrations are so slow that the quasi-static approach, which neglects their time derivative while computing the flux, can practically be regarded as exact. As for thermal hydraulic quantities, such as the temperature and the coolant density, in the framework of lattice calculations it is customary to treat them as constant (in time) input parameters. The two sub-problems are then solved alternately: the steady-state flux solution of the transport solver at time t is used to compute the microscopic reaction rates (Eq. (2.4), with the microscopic cross section in place of the macroscopic one), which are kept constant by the depletion solver while evaluating the nuclide concentrations at time $t + \Delta t$, corresponding to the next burnup level. The obtained concentrations allow updating the macroscopic cross sections to perform the subsequent transport calculation. The flux and the depletion calculation are therefore repeated *at least once* for each burnup

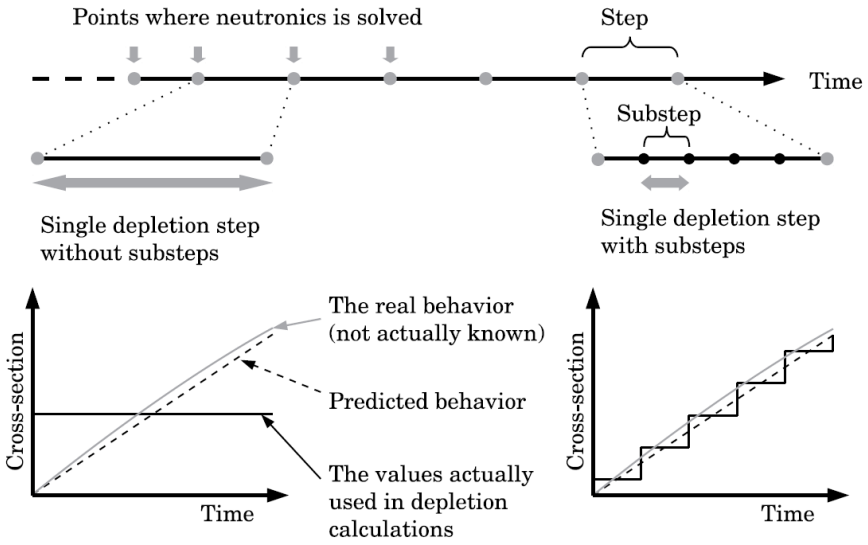


Figure 3.2: Depletion-neutronics coupling with and without the sub-step method. It can be seen that the additional sub-step mesh for the sole depletion allows considering, for reaction rates, a piece-wise constant profile which is much closer to the predicted value. Image source: [32].

value of the depletion mesh.

Coupled with the APOLLO3® transport simulation, the MENDEL code [29] is in charge of the depletion resolution for a given time step: the system of differential equation in time is solved either by a 4^{th}-order Runge-Kutta method [30] or a CRAM solver (Chebyshev Rational Approximation Method) [31]. The Δt choice for the depletion step (or, equivalently, of the $\Delta\, burnup$) impacts directly on the solution accuracy and computing time: since microscopic rates are fixed, adopting shorter time steps is more accurate, but requires more calculations to be performed.

In order to optimize the joint work of the two solvers, various coupling schemes have been developed, based on predictor-corrector strategies that allow for a coarser time discretization than the simpler approach that updates the concentrations taking into account only the microscopic rates at the beginning of the time step. The scheme coupling APOLLO3® and MENDEL used in this thesis is based on A. Isotalo's thesis work [32]: as the overall computing time is dominated by the flux calculations, the rationale is to reduce the times these are performed by considering a further time mesh for the sole depletion. In doing so, time sub-steps are introduced between two successive transport simulations and the flat microscopic rates assumption is substituted by a piece-wise constant description over each step, the rates being constant within every sub-step. Figure 3.2 reports the dual discretization and the approximated time profiles of cross sections (hence, of reaction rates). The depletion solver needs be applied many times, once per sub-step, but the method permits a significant increase of the step sizes and, ultimately, a reduction of the overall computing time.

Also the sub-steps approach envisages a predictor and a corrector phase for each time step: during the former, the nuclide concentrations are computed using an extrapolated flux, resulting from the latest transport solutions, and applying MENDEL within sub-steps.

The predicted concentrations are then used to perform a new flux calculation, yielding the predicted flux. The following corrector phase repeats the procedure but employing a flux obtained by interpolation of the predicted flux and the ones of the previous steps. The resulting corrected concentrations are finally compared to the predicted values: if their difference is greater than the tolerance, the time step is divided into two equal portions and the whole procedure restarts from the first one. Hence, each unsuccessful corrector phase entails flux calculation that can be avoided if the proper time mesh is adopted from the beginning: in other words, an optimized depletion mesh for a given system leads only to successful corrector phases.

As a final remark, similarly to what have been said about self-shielding, the spatial higher-order method presented in this manuscript has not yet been extended to the available depletion solver, which at present is capable to deal only with spatially constant quantities. The equivalence procedure between the two approximations will be discussed in section 5.4.

Chapter 4

The method of characteristics in neutronics

In the vast domain of computational methods, the Method Of Characteristics (MOC) plays a prominent role as it allows reducing a Partial Differential Equation (PDE) to an Ordinary Differential Equation (ODE). This is possible by applying the involved equation over its characteristic lines, along which the gradient projection, generally dependent on more than one partial derivative, can be expressed as a derivative with respect to a single variable. Moreover, the MOC can be applied to arbitrary geometries, in principle, which contributes to make it one of the most widespread deterministic methods in neutronics.

This chapter aims to present the MOC declination to the field of reactor physics simulations, describing how Boltzmann equation is rephrased along characteristic lines and reporting a non-exhaustive list of the main MOC applications in nuclear deterministic codes. The discussion will then focus on TDT, the 3D MOC solver of APOLLO3®, dealing with the already cited step-constant and axial polynomial approximations that constitute the basis for the developments of this thesis.

4.1 The transport equation along characteristics

As anticipated, Eq. (2.26) represents the starting point for the derivation of the characteristic form of the transport equation. Introducing the mono-group emission density $q(\vec{r},\vec{\Omega})$, the equation becomes

$$\left(\vec{\Omega}\cdot\nabla+\Sigma_T(\vec{r})\right)\psi(\vec{r},\vec{\Omega})=q(\vec{r},\vec{\Omega})\approx\sum_{n=1}^{N_m}A_n(\vec{\Omega})\,q_n(\vec{r}) \tag{4.1}$$

where $\{q_n\}$ are the angular moments of q defined as

$$q_n(\vec{r})=\sum_{g'=1}^{N_g}\Sigma_{S,l(n)}^{g'\to g}(\vec{r})\,\Phi_n^{g'}(\vec{r})+\frac{\delta_{1n}}{k_{\text{eff}}}\sum_{i=1}^{N_f}\chi_i^g\sum_{g'=1}^{N_g}\nu\Sigma_{F,i}^{g'}(\vec{r})\,\Phi_1^{g'}(\vec{r}), \tag{4.2}$$

δ_{1n} being the Kronecker delta. The idea is to substitute the streaming operator $\vec{\Omega}\cdot\nabla*$ by a term containing a single derivative; to do that, one can consider the flux derivative with

respect to a parameter s, and compare its expression with the original streaming term:

$$\begin{aligned}\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial s} &= \boxed{\frac{dx}{ds}}\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial x} + \boxed{\frac{dy}{ds}}\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial y} + \boxed{\frac{dz}{ds}}\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial z} \\ \vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega}) &= \boxed{\Omega_x}\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial x} + \boxed{\Omega_y}\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial y} + \boxed{\Omega_z}\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial z}.\end{aligned} \tag{4.3}$$

Hence, along the characteristic $\vec{r} = \vec{r}_0 + s\,\vec{\Omega}$, the two terms above are equal and Eq. (4.1) can be written as

$$\frac{\partial\psi(\vec{r},\vec{\Omega})}{\partial s} + \Sigma_T(\vec{r})\,\psi(\vec{r},\vec{\Omega}) = q(\vec{r},\vec{\Omega}), \tag{4.4}$$

which is analytically solvable. The integration between $\vec{r}_0$ and $\vec{r}$ yields

$$\psi(\vec{r}_0 + s\,\vec{\Omega},\vec{\Omega}) = \psi(\vec{r}_0,\vec{\Omega})\,e^{-\tau(s)} + \int_0^s ds'\,q(\vec{r}_0 + s'\,\vec{\Omega},\vec{\Omega})\,e^{-[\tau(s)-\tau(s')]}, \tag{4.5}$$

where the optical path length τ is defined as

$$\tau(s) = \int_0^s ds'\,\Sigma_T(\vec{r}_0 + s'\,\vec{\Omega}). \tag{4.6}$$

The equation just obtained is the integral form of the time-independent mono-group Boltzmann equation and can be used to solve the inner-iteration problem (3.27) by evaluating how the flux propagates along each chord[1] of the system. In sections 4.4.1 and 4.4.2 its numerical forms will be presented, depending on the chosen spatial approximation.

4.2 Trajectory-based surface integration

In MOC-based solvers, characteristic lines are traced over the entire domain, parallel to the directions chosen for the angular quadrature, and the angular flux transmission is evaluated *only* along them: around each characteristic, one can therefore identify a channel (reducing to a rectangle in 2D problems), within which a homogeneous front propagates along the quadrature direction, as shown in Fig. 4.1. In other words, the approximation is made that the angular flux is constant on the cross-sectional area of the channel, whereas its value varies along the characteristic according to an adapted version of Eq. (4.5).

In the limit of an infinite number of characteristics per direction (or equivalently, of channels with infinitesimal sections), the sum of the volumes of the channel portions contained in a computational region coincides with the volume of the region itself. On the other hand, a finite number of characteristics leads to approximate the region volume as the sum over the chord-related sub-channel volumes, which depends on the quadrature direction considered; for region r and direction $\vec{\Omega}$, the $\tilde{V}$ approximation reads

$$\tilde{V}_r(\vec{\Omega}) = \Delta_{\perp\vec{\Omega}} \sum_{\substack{t\parallel\vec{\Omega} \\ t\subset r}} l_t, \tag{4.7}$$

[1] A chord is any segment resulting from the intersections of the characteristics with the spatial mesh

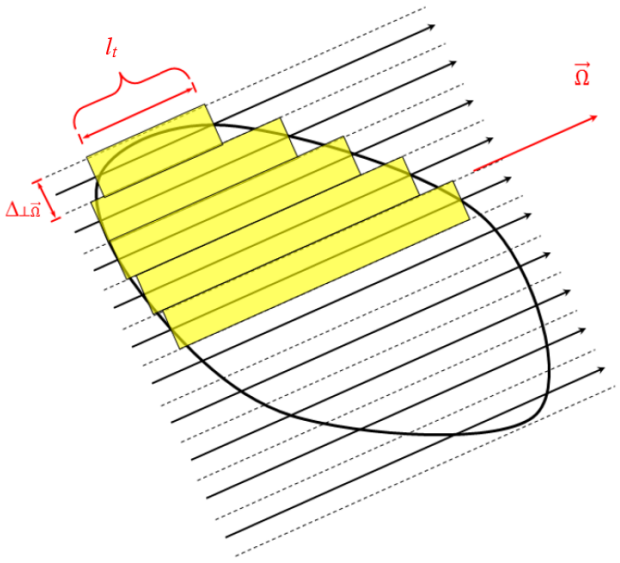


Figure 4.1: Trajectory-based discretization of the region volume. A yellow sub-channel is associated to each chord, with the angular flux being constant on its cross-sectional area. The sum of the sub-channel volumes yields an angular-dependent approximation of the volume of the region. The image is taken from [11].

where the sum is performed over chords $\{t\}$ belonging to r and parallel to $\vec{\Omega}$, l_t is the chord length and $\Delta_{\perp\vec{\Omega}}$ is the cross-sectional area of the channel, which is usually kept constant per direction (approach adopted in TDT, as explained in [4]). The real volume of the region, V_r, can then be approximated by averaging over directions:

$$V_r \approx \oint \frac{d\vec{\Omega}}{4\pi} \tilde{V}_r(\vec{\Omega}) \approx \sum_{k=1}^{N} w(\vec{\Omega}_k)\, \tilde{V}_r(\vec{\Omega}_k), \tag{4.8}$$

using the same S_N notation as in Eq. (2.23). The approximated surface integration used in Eq. (4.7) will come in handy later while showing how the $\vec{\Omega}$-contribution to the streaming term is cumulated over chords.

4.3 Neutron transport solvers

Before dealing with the specific features of the MOC implemented in APOLLO3®, an overview of the major applications of the method is presented in this paragraph. Due to the vast literature available, it would certainly be impossible to exhaust the topic here. Hence, the purpose is rather to outline the relevant developments, regarding the complexity of the geometries treated and the refinement of the methods adopted.

The introduction of the MOC in the field of nuclear reactor calculations was firstly proposed by Askew in 1972 [33], which led to its implementation in the English code CACTUS [34]. This was followed by several applications of the method, such that of Suslov in MCCG3D [35], Knott and Edenius in CASMO-4 [36], Cho and Hong in CRX [37] and, more

recently, Boyd et al. in OpenMOC [38], establishing the MOC as a prominent approach for 2D lattice solvers.

The method has actually many advantages: it can be applied to very complex geometry, as it does not need particular regularities, it allows for arbitrary anisotropy order and, evaluating the transport solution along characteristics, it is capable to treat exactly the streaming operator. As drawbacks, though, the MOC demands important computational resources: in terms of floating-point operations, which leads to implement massively parallelized routines (see section 4.4.3), and memory, needed to store the position of all the chords. These problematics pushed for the development of more sophisticated techniques; regarding memory, spatial high-order methods have been introduced, aiming at limiting the required storage by reducing the number of computational regions. An example is provided by the work of Ferrer and Rhodes in 2016 [39].

The limited resources have long slowed the MOC extension to the third dimension. In more recent times, however, the availability of more powerful computers and the interest in considering more realistic physical systems have brought to the introduction of 2D/1D *fusion* approaches as well as actual 3D MOC's. The first class of methods couples a fine MOC solution on the radial plane with a lower-order approximation along the axial direction; thus they can only be applied to axially extruded geometries. Examples of this technique can be found in the Korean DeCART code [40], where a one-dimensional S_N solver is adopted along the axis, and the Japanese GENESIS code [41], which uses the Legendre polynomial Expansion of the Angular Flux (LEAF) to express the spatial dependence of the angular flux over *characteristics planes*, orthogonal to the radial plane and resting on 2D characteristics. The latter strategy is quite similar to the one employed by Short-MOC solvers (see next section). Further relevant applications of fusion techniques can be found, among others, in the MPACT [42] and CRX 3D [43] codes.

Concerning the full 3D solvers, another work of Suslov [44] represents the first known application involving 3D characteristics, in this case limited to axially extruded geometries. The possibility of treating more general (even if not arbitrary) geometries is offered by the MCI module of DRAGON code [45], wherein the domain can be defined as a combination of different axially extruded cells (constituting a so-called *supercell* geometry). The problem solved by MCI is not precisely a full 3D MOC, though, because the entering fluxes is supposed isotropic at the supercells interfaces. Yet, the code provides accelerated higher order schemes [46]. The already cited MPACT and OpenMOC also contain full 3D solvers: references can be found in [6] and [7], respectively. Moreover, the higher order method implemented in OpenMOC is akin to the one of TDT (see section 4.4.2), as it assumes a quadratic axial expansion of emission densities. A very recent and breakthrough work is the implementation of the MOCkingbird code [10], capable of simulating high-fidelity full-core 3D cases by resorting to massive computational resources (without acceleration, at the present time).

To conclude this concise overview, a mention apart is reserved to a relatively recent type of solvers: the stochastic-deterministic hybrid methods, which aim combine the Monte Carlo adherence to physics with the relatively high calculation speed offered by deterministic approaches. Here a single example of hybrid method involving MOC is reported, The Random Ray Method (TRRM), which has been implemented in the ARRC code as a promising numerical tool targeting full core 3D simulation [9]. The peculiarity of TRRM is the fact

that the characteristics tracking is performed stochastically and iteratively repeated while updating the angular flux in all regions. Besides the capability to be applied to arbitrary geometries, the advantages are related to the great memory efficiency, as TRRM does not require to store the tracking data, and to the possibility of exploiting a three-way hybrid parallel programming: SIMD (Single Instruction Multiple Data) vectorization, OpenMP [47] and MPI [48].

4.3.1 APOLLO3® MOC-based solvers: TDT and IDT

APOLLO3® contains two different lattice solvers adopting the MOC solvers, both applicable to 3D geometries.

IDT [49] uses the already mentioned Short-MOC: within each computational region, the surface fluxes are represented by approximated functions, usually polynomials, and the characteristics are used only in a preliminary phase, during which a set of coefficients coupling the volume emission densities and the surface fluxes is evaluated. These coefficients are computed once and for all but demand a relevant memory storage, which can pose a problem for large systems. Also, for highly heterogeneous cases the surface approximation has to be well refined to reach a good spatial convergence and, even though concentric circles are admitted as region boundaries [50], the spatial expansion over plane surfaces limits the arbitrariness of the domain. However, it must be emphasized that the solver takes advantage of both the OpenMP and the MPI parallel strategies.

This thesis work has been implemented in the second module, TDT, based on the more widespread (Long-)MOC approach. For the record, TDT is also a 2D Collision Probability solver. Differently from the S-MOC, characteristics are followed throughout the domain, which allows treating any system (and region) shape, in principle. In practice, however, identifying the intersections of characteristics with the spatial mesh is quite a difficult task and, at present, the only recognizable boundaries are line segments and circumference arcs. This is not really a limiting factor, as the vast majority of nuclear systems are well represented by axial geometries obtained by extrusion of a grid of square or rectangular cells, containing various concentric circles to reproduce fuel rods and guide tubes. A more detailed presentation of TDT 3D MOC is the object of the next section.

4.4 3D MOC in TDT

Based on its homonymous predecessor, developed in the APOLLO2 code [51], TDT was implemented in APOLLO3® and extended to 3D axially extruded geometries [2]; at present, the solver continues to be improved and updated. Its classical and higher-order spatial approximations of the angular flux moments are described in the following, as well as the main optimization features of the 3D method.

4.4.1 Step-constant approximation

The traditional hypothesis assumed in MOC solvers is that the moments of the angular flux (more precisely, the moments of the emission densities) and the macroscopic cross sections

are *spatially constant* within each computational region (2D or 3D). This is usually referred to as Step-Characteristic or Step-Constant approximation; in the rest of the manuscript it will be named SC. On the basis of Eq. (2.13), the assumption is equivalent to consider, for all regions, $N_e = 1$ and $f_{r,1} = 1$, in such way that the n^{th} moment of the flux and the macroscopic cross section for the generic reaction R in region r (for an arbitrary energy group) read

$$\Phi_n(\vec{r}) \approx \Phi_{n,r}, \qquad \Sigma_R(\vec{r}) \approx \Sigma_{R,r}. \tag{4.9}$$

Consequently, the mono-group emission density introduced in Eq. (4.1) simplifies as

$$q(\vec{r}, \vec{\Omega}) \approx q_r(\vec{\Omega}) = \sum_{n=1}^{N_m} A_n(\vec{\Omega})\, q_{n,r}, \tag{4.10}$$

where the angular moments $\{q_{n,r}\}$ are equal to

$$q_{n,r} = \sum_{g'=1}^{N_g} \Sigma_{S,l(n),r}^{g' \to g}\, \Phi_{n,r}^{g'} + \frac{\delta_{1n}}{k_{\text{eff}}} \sum_{i=1}^{N_f} \chi_i^g \sum_{g'=1}^{N_g} \nu\Sigma_{F,i,r}^{g'}\, \Phi_{1,r}^{g'}. \tag{4.11}$$

One can take advantage of q being spatially constant while applying Eq. (4.5) between the endpoints of a chord, as the latter belongs entirely to a single region. The resulting numerical transmission equation is

$$\psi^+(\vec{\Omega}) = \psi^-(\vec{\Omega})\, e^{-\tau(l)} + \frac{q_r(\vec{\Omega})}{\Sigma_{T,r}} \left(1 - e^{-\tau(l)}\right), \tag{4.12}$$

where ψ^- and ψ^+ denote the values of the angular flux at the entry point $\vec{r}_0$ and at the exit point $\vec{r}_0 + l\vec{\Omega}$ of the chord, respectively, l is the chord length and $\tau(s) = \Sigma_{T,r}\, s$.

Knowing the emission density value in all regions, Eq. (4.12) is repeatedly used, starting from the system boundary, to compute the angular flux along all trajectories, in what is commonly known as *transport sweep*. Once this step is done, a balance relation is needed to update the flux moments and, therefore, the emission densities through Eqs. (4.10) and (4.11) for the next inner iteration. This equation is obtained by averaging Eq. (4.1), under the SC hypotheses, over the region volume:

$$\frac{1}{V_r} \int_r d\vec{r}\, \vec{\Omega} \cdot \nabla\psi(\vec{r}, \vec{\Omega}) + \Sigma_{T,r}\, \psi_r(\vec{\Omega}) = q_r(\vec{\Omega}), \tag{4.13}$$

$\psi_r(\vec{\Omega})$ being the volume average angular flux and $\int_r d\vec{r}$ the integral over the region volume. Since $\vec{\Omega} \cdot \nabla\psi = \nabla \cdot (\vec{\Omega}\, \psi)$, the first term can be modified using the divergence theorem, converting the volume integral into an integral over the region boundary ∂r with normal vector $\hat{n}$:

$$\begin{aligned} \frac{1}{V_r} \int_r d\vec{r}\, \vec{\Omega} \cdot \nabla\psi(\vec{r}, \vec{\Omega}) &= \frac{1}{V_r} \int_{\partial r} d\vec{r}_s\, \vec{\Omega} \cdot \hat{n}\, \psi(\vec{r}_s, \vec{\Omega}) \\ &= \frac{1}{V_r} \int_{\vec{\Omega}\cdot\hat{n} > 0} d\vec{r}_s\, \vec{\Omega} \cdot \hat{n}\, \psi(\vec{r}_s, \vec{\Omega}) + \frac{1}{V_r} \int_{\vec{\Omega}\cdot\hat{n} < 0} d\vec{r}_s\, \vec{\Omega} \cdot \hat{n}\, \psi(\vec{r}_s, \vec{\Omega}). \end{aligned} \tag{4.14}$$

Then, adopting the same trajectory-based surface integration as in Eq. (4.7), that is, substituting the actual boundary ∂r by a *numerical* boundary constituted by the ends of the characteristics channels and by surfaces on which $\vec{\Omega} \perp \hat{n}$, one can exploit the fact that $\vec{\Omega} \cdot \hat{n}$ is equal to 1 and -1 at the chord endpoints. The surface integral is therefore approximated as

$$\frac{1}{V_r} \int_{\partial r} d\vec{r}_s \, \vec{\Omega} \cdot \hat{n} \, \psi(\vec{r}_s, \vec{\Omega}) \approx \frac{\Delta_{\perp\vec{\Omega}}}{V_r} \sum_{\substack{t \parallel \vec{\Omega} \\ t \subset r}} \left[\psi_t^+(\vec{\Omega}) - \psi_t^-(\vec{\Omega}) \right] = \Delta J_r(\vec{\Omega}), \tag{4.15}$$

so that the resulting SC balance reads

$$\Sigma_{T,r} \, \psi_r(\vec{\Omega}) = q_r(\vec{\Omega}) - \Delta J_r(\vec{\Omega}). \tag{4.16}$$

The ΔJ term is cumulated during the transport sweep with the angular fluxes given by the transmission equation. The only unknown in the previous relation is therefore the average angular flux, from which the new flux moments iterates are obtained as

$$\Phi_{n,r} = \oint \frac{d\vec{\Omega}}{4\pi} \, A_n(\vec{\Omega}) \, \psi_r(\vec{\Omega}), \qquad n = 1, N_m. \tag{4.17}$$

4.4.2 Axial polynomial flux expansion

As already mentioned, the MOC requires huge computational resources which hinder its application to large nuclear domains. In order to improve performances by reducing the number of regions and, therefore, the tracking data to be stored and the times Eq. (4.12) needs be applied along a trajectory, a polynomial development was adopted for the moments of the neutron flux along the sole extrusion direction z [11]. This approximation will be named POL in the following. In fact, with the same number of regions removed along the axis or in the radial direction, the axial gradients, much smaller than the radial ones, require a lower expansion order to attain an equally accurate description. The higher-order description along z thus allows for a great reduction of the number of axial layers with a relatively low polynomial order and while keeping the SC approximation radially.

Referring again to Eq. (2.13), the POL method introduces $N_e(r) = N_p(r) + 1$ polynomial functions, N_p being the expansion order which may vary depending on the region. The $\vec{f}$ functions of the basis are defined as

$$\vec{f}_r(\vec{r}) = \vec{P}_r(\vec{r}) = \{(\tilde{z}_r)^p, \, p = 0, N_p(r)\}, \qquad \tilde{z}_r = \frac{z_r - \bar{z}_r}{\Delta z_r / 2}. \tag{4.18}$$

z_r, $\bar{z}_r$ and Δz_r are the absolute axial coordinate, the mid-height value for region r and its height, respectively, so that the value of $\tilde{z}_r$ ranges between -1 and 1 in the whole domain. Considering one region at a time will allow omitting the region index r for the polynomial basis and the order N_p. In this way, the n^{th} angular flux moment is written as

$$\Phi_n(\vec{r}) \approx \vec{P}(\vec{r}) \cdot \vec{\Phi}_{n,r}, \qquad \vec{\Phi}_{n,r} = \{\Phi_{n,r,p}, \, p = 0, N_p\}, \tag{4.19}$$

$\vec{\Phi}_{n,r}$ being the vector of the polynomial coefficients for the adopted expansion. It is important to recall that this approach does not concern cross sections, which are kept constant in

all regions. Using the polynomial definition of the flux moments in the expression of the moments of the mono-group emission density (Eq. (4.2)) leads to write an analogous scalar product relation for the angular emission density:

$$q(\vec{r},\vec{\Omega}) \approx \vec{P}(\vec{r}) \cdot \vec{q}_r(\vec{\Omega}) = \vec{P}(\vec{r}) \cdot \sum_{n=1}^{N_m} A_n(\vec{\Omega})\, \vec{q}_{n,r}, \tag{4.20}$$

where the p^{th}-order component of $\vec{q}_{n,r}$ reads

$$q_{n,r,p} = \sum_{g'=1}^{N_g} \Sigma_{S,l(n),r}^{g' \to g}\, \Phi_{n,r,p}^{g'} + \frac{\delta_{1n}}{k_{\text{eff}}} \sum_{i=1}^{N_f} \chi_i^g \sum_{g'=1}^{N_g} \nu\Sigma_{F,i,r}^{g'}\, \Phi_{1,r,p}^{g'}. \tag{4.21}$$

Substituting the q expression (4.20) in the analytical transmission (4.5) expressed over the $(\vec{r}_0, \vec{r}_0 + l\vec{\Omega})$ chord, one gets

$$\begin{aligned} \psi^+(\vec{\Omega}) - \psi^-(\vec{\Omega})\, e^{-\tau(l)} &= \int_0^l ds\, \vec{P}(\vec{r}_0 + s\vec{\Omega}) \cdot \vec{q}_r(\vec{\Omega})\, e^{-[\tau(l)-\tau(s)]} \\ &= \int_0^l ds \left[\sum_{p=0}^{N_p} \left(\frac{z_0 + s\mu - \bar{z}_r}{\Delta z_r/2} \right)^p q_{r,p}(\vec{\Omega}) \right] e^{-[\tau(l)-\tau(s)]} = \vec{P}(\vec{r}_0) \cdot \vec{T}, \end{aligned} \tag{4.22}$$

where z_0 is the axial coordinate of $\vec{r}_0$, μ the cosine of the polar component of $\vec{\Omega}$ and $\vec{T}$ is the following vector:

$$\begin{aligned} &\vec{T} = \{T_k,\, k = 0, N_p\}, \\ &T_k = \sum_{p=k}^{N_p} \binom{p}{k} \left(\frac{2\mu}{\Delta z_r} \right)^{p-k} \frac{q_{r,p}(\vec{\Omega})}{\Sigma_{T,r}}\, E_{p-k}(\tau), \qquad E_j(\tau) = \frac{1}{(\Sigma_{T,r})^j} \int_0^{\tau(l)} d\tau'\, \tau'^j\, e^{-[\tau(l)-\tau']}. \end{aligned} \tag{4.23}$$

The values of the escape coefficients $\{E_j\}$ are pre-tabulated (one table for each $j \in \{0, 1, \ldots, N_p\}$) and retrieved, using τ as interpolation parameter, while computing the $\vec{T}$ terms. This latter operation considerably benefits from the Chords Classification Method described in section 4.4.3.

As in the SC case, a balance equation is needed to close the iterative problem. In the POL method, the approach followed is to project Eq. (4.1) on the polynomial basis. In this regard it is useful to define the polynomial moments of the angular flux as

$$'\vec{\psi}_r(\vec{\Omega}) = \frac{1}{V_r} \int_r d\vec{r}\, \vec{P}(\vec{r})\, \psi(\vec{r},\vec{\Omega}) \tag{4.24}$$

and the tensorial product of two n-dimension vectors $\vec{a}$ and $\vec{b}$ as

$$\vec{a} \otimes \vec{b} = \begin{pmatrix} a_1\, b_1 & a_1\, b_2 & \ldots & a_1\, b_n \\ a_2\, b_1 & a_2\, b_2 & \ldots & a_2\, b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n\, b_1 & a_n\, b_2 & \ldots & a_n\, b_n \end{pmatrix}. \tag{4.25}$$

Applying thus the $\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})*$ projection yields

$$\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\Big[\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega})\Big]+\Sigma_{T,r}\,'\vec{\psi}_r(\vec{\Omega})=\bar{\bar{\mathcal{P}}}\,\vec{q}_r(\vec{\Omega}), \tag{4.26}$$

where matrix $\bar{\bar{\mathcal{P}}}$ is a Galerkin-like matrix equal to

$$\bar{\bar{\mathcal{P}}}=\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\otimes\vec{P}(\vec{r}). \tag{4.27}$$

Actually, a numerical version of the previous matrix needs be used to guarantee particle conservation and, therefore, numerical stability. This form will be described in chapter 5 while presenting its equivalent for the polynomial cross sections method. Exploiting the properties of the $\nabla*$ operator:

- $\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega})=\nabla\cdot\Big[\vec{\Omega}\,\psi(\vec{r},\vec{\Omega})\Big]$
- $\Big(\nabla\cdot\vec{a}\Big)\,\vec{b}=\nabla\Big(\vec{a}\otimes\vec{b}\Big)-\vec{a}\,\Big(\nabla\otimes\vec{b}\Big)$,

the first term of Eq. (4.26) is modified as

$$\begin{aligned}\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\Big[\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega})\Big]=&\underbrace{\frac{1}{V_r}\int_r d\vec{r}\,\nabla\cdot\Big[\vec{\Omega}\,\psi(\vec{r},\vec{\Omega})\otimes\vec{P}(\vec{r})\Big]}_{A}\\&-\underbrace{\frac{1}{V_r}\int_r d\vec{r}\,\vec{\Omega}\,\psi(\vec{r},\vec{\Omega})\Big[\nabla\otimes\vec{P}(\vec{r})\Big]}_{B}.\end{aligned} \tag{4.28}$$

Term A is expressed in an analogous way to Eq. (4.15), but accounting for all polynomial components, as

$$\begin{aligned}\frac{1}{V_r}\int_r d\vec{r}\,\nabla\cdot\Big[\vec{\Omega}\,\psi(\vec{r},\vec{\Omega})\otimes\vec{P}(\vec{r})\Big]&=\frac{1}{V_r}\int_{\partial r}d\vec{r}_s\,\vec{\Omega}\cdot\hat{n}\,\vec{P}(\vec{r}_s)\,\psi(\vec{r}_s,\vec{\Omega})\\&\approx\frac{\Delta_{\perp\vec{\Omega}}}{V_r}\sum_{\substack{t\parallel\vec{\Omega}\\ t\subset r}}\Big[\vec{P}(\vec{r}_{0,t}+l_t\vec{\Omega})\,\psi_t^+(\vec{\Omega})-\vec{P}(\vec{r}_{0,t})\,\psi_t^-(\vec{\Omega})\Big]=\Delta\vec{J}_r(\vec{\Omega})\end{aligned} \tag{4.29}$$

and term B exploits the fact that the only non-zero elements of $\nabla\otimes\vec{P}(\vec{r})$ are along z, becoming

$$\frac{1}{V_r}\int_r d\vec{r}\,\vec{\Omega}\,\psi(\vec{r},\vec{\Omega})\,\Big[\nabla\otimes\vec{P}(\vec{r})\Big]=\frac{1}{V_r}\int_r d\vec{r}\,\mu\,\frac{d\vec{P}(\vec{r})}{dz}\,\psi(\vec{r},\vec{\Omega}). \tag{4.30}$$

Moreover, the derivative of the p^{th}-degree polynomial of the basis reads

$$\frac{dP_p(\vec{r})}{dz}=\frac{p}{\Delta z_r/2}\,P_{p-1}(\vec{r})\quad\forall p>0 \tag{4.31}$$

and is equal to 0 otherwise. Term B can then be written as $\mu\,\bar{\bar{\mathcal{C}}}_r\,'\vec{\psi}_r(\vec{\Omega})$, $\bar{\bar{\mathcal{C}}}_r$ being matrix of rank N_p+1 whose non-zero elements are all along the subdiagonal:

$$\bar{\bar{\mathcal{C}}}_r = \frac{2}{\Delta z_r}\begin{pmatrix} 0 & \dots & \dots & \dots & 0 \\ 1 & \ddots & & & \vdots \\ 0 & 2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & N_p & 0 \end{pmatrix}. \tag{4.32}$$

Finally, denoting by $\bar{\bar{\mathcal{I}}}$ the identity matrix, Eq. (4.26) becomes

$$\left(\Sigma_{T,r}\,\bar{\bar{\mathcal{I}}} - \mu\,\bar{\bar{\mathcal{C}}}_r\right)'\vec{\psi}_r(\vec{\Omega}) = \bar{\bar{\mathcal{P}}}\,\vec{q}_r(\vec{\Omega}) - \Delta\vec{J}_r(\vec{\Omega}), \tag{4.33}$$

which is a sub-triangular system of equations having as unknowns the polynomial moments of the angular flux, easily solved by forward substitution after cumulating $\Delta\vec{J}_r$ during the sweep. The expansion coefficients can then be retrieved by inversion of matrix $\bar{\bar{\mathcal{P}}}$:

$$\vec{\psi}_r(\vec{\Omega}) = \bar{\bar{\mathcal{P}}}^{-1}\,'\vec{\psi}_r(\vec{\Omega}) \tag{4.34}$$

and eventually used to update the angular moments through Eq.(4.17), for all polynomial coefficients.

Eq. (4.33) (as well as its SC equivalent, Eq. (4.16)) represents an angular formulation of neutron balance. In the actual calculation another version is employed, evaluating directly the polynomial moments of the flux angular moments $\{'\Phi_{n,r,p}\}$, to which $\bar{\bar{\mathcal{P}}}^{-1}$ is applied to obtain the new iterates. In order to preserve the triangular structure of the problem, a strategy was developed introducing higher-order flux angular moments and consequently enlarging the matrices involved in the balance. These additional terms are then discarded once all the needed polynomial moments are obtained. A direct approach has been adopted also in this thesis work, but without adding higher-order terms (see chapter 5).

4.4.3 Transport sweep optimization

The transport sweep is, from a computational point view, the heaviest part of the non-accelerated MOC power-iteration scheme. For this reason, various optimization techniques were introduced in TDT to limit the impact on memory of the 3D tracking and to render the sweep of the trajectories more efficient. The details concerning these methods are found in [2] and [5]; here they are shortly reported, since all the following developments have taken advantage of them, including the work reported in this thesis.

Chords Classification Method (CCM) Differently from arbitrary 3D domains, axially extruded geometries allow reducing significantly the size of the tracking storage by tracing the 3D characteristics based on an underlying 2D footprint: the radial plane xy is filled with 2D characteristics of distance equal to the radial transversal step Δ_r, for all the considered azimuthal angles. Identifying an s direction, each of these lines

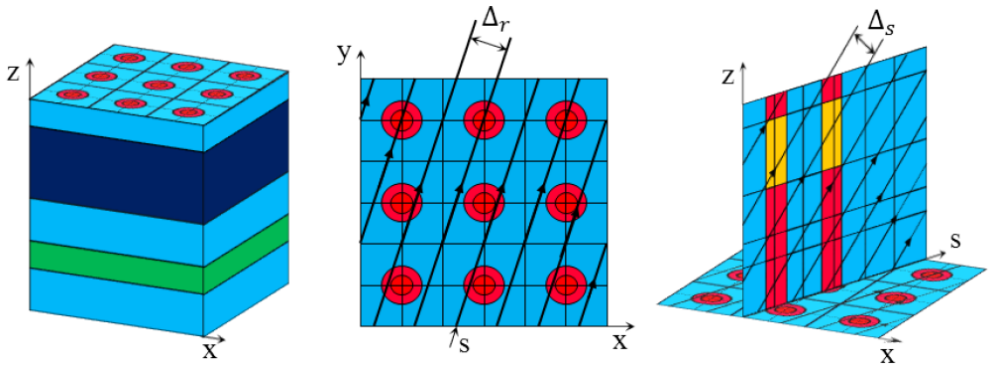
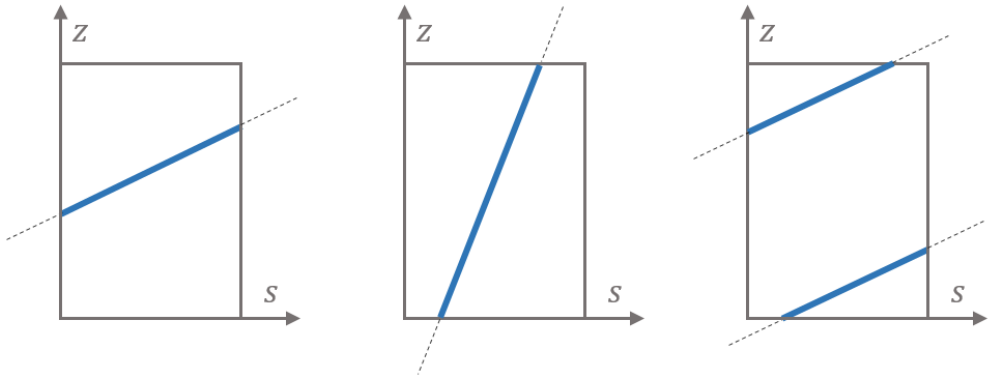


Figure 4.2: 3D tracking for axially extruded geometries. The 3D trajectories are traced on the sz planes obtained through extrusion of the 2D characteristics, belonging to the xy plane. Image source: [2].



(a) (b) (c)

Figure 4.3: Classes of 3D chords: V-chords (a), H-chords (b) and M-chords (c).

permits considering an sz plane extending along the axis, on which the 3D tracking is performed; for each polar angle, the corresponding 3D characteristics are traced at a distance Δ_s, by which one denotes the transversal step on sz. This can be seen in Fig. 4.2: the resulting sz mesh is Cartesian, property that is exploited to regroup the 3D chords into classes and to consequently avoid the storage of all of them. Three classes are identified (Fig. 4.3):

- V-chords, having both endpoints on two subsequent vertical surfaces. A class of V-chords is uniquely defined by their common footprint on the radial plane, chord i_{2D}, by the polar angle θ and by the axial plane wherein they are contained. Their length, however, is the same regardless of the axial coordinate, as it is given by

$$l^V_{i_{3D}} = \frac{l_{i_{2D}}}{\sin\theta}, \tag{4.35}$$

  $l_{i_{2D}}$ being the length of the underlying 2D chord.
- H-chords, which are comprised between two horizontal surfaces. Hence, their length depends only on the height of the axial plane and on $\mu = \cos\theta$, according to the

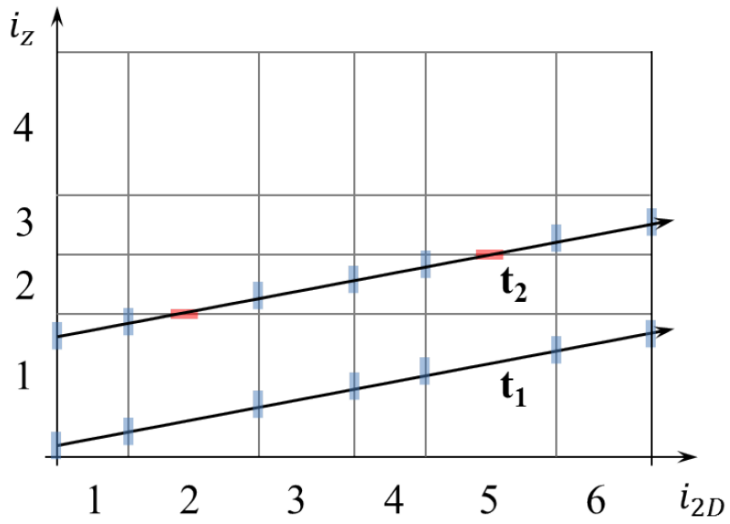


Figure 4.4: HSS examples for two trajectories crossing vertical (blue) and horizontal (red) surfaces in the sz plane. Since t_1 crosses 7 successive vertical surfaces, its sequence reads only $HSS_1 = \{-7\}$. On the other hand, $HSS_2 = \{-2, 1, -3, 1, -2\}$, because t_2 meets 2 vertical surfaces (1 V-chord + 1 joining M-chord), 1 horizontal surface (0 H-chords + 1 joining M-chord), 3 vertical surfaces (2 V-chords + 1 joining M-chord) and so forth. Note that the last HSS element does not involve a joining M-chord. The image is adapted from the original in [2].

formula

$$l^H_{i_{3D}} = \frac{\Delta z_r}{\mu}. \tag{4.36}$$

To identify a class, on the other hand, one needs to specify also the 2D region the projection of the H-chord on xy belongs to.

- M-/Non-recognized chords, having the endpoints on surfaces of different types ("M" stands for "Mixed").

The CCM allows optimizing the storage and the evaluation of the transmission coefficients $e^{-\tau}$, $1 - e^{-\tau}$ (SC case) and $\vec{T}$ (POL case), which, for classified chords, are precomputed and then retrieved during the transport sweep. In fact, since they have the same length, in the SC cross sections approximation all the V-chords and the H-chords of a class share the same optical path length and therefore the same coefficients. As for the M-chords, some attempts were made to classify particular sub-cases [2], but generally the most efficient strategy is to construct, for each 3D trajectory, a vector containing the chords lengths in the same order as they are encountered and to compute the corresponding transmission coefficients on-the-fly. Since classified chords constitute the vast majority in all the treated geometries, the CCM proves to be extremely advantageous. This is all the more true when using the POL method, as the smaller number of axial meshes reduces the number of classes by increasing the average number of chords belonging to each class.

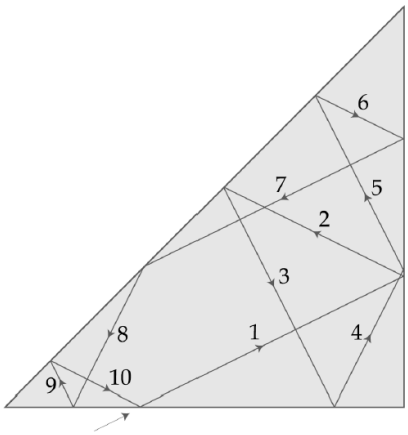


Figure 4.5: Example of cyclic 2D trajectory. The chords are oriented according to the 10 connected angles. Image source: [52].

Hit Surfaces Sequence (HSS) The second optimization technique concerns the way 3D chords are subsequently recognized during the sweep. This is done compacting the tracking information into sequences of integers, whose values indicate how many chords of equal class are encountered along the trajectory. In practice, the HSS will contain a 1 whenever an M-chord is met or a number of module > 1 defining the number of successive V- or H-chords, depending on the integer sign (see Fig. 4.4). Based on the information provided by the HSS, one can update the current 2D chord and axial plane, needed to determine the class, while following the 3D trajectory: in fact, encountering an M-chord causes a change in the axial plane, while a vertical surface (after each V-chord or an M-chord traveled from the horizontal border) modifies the 2D chord counter. In order to update the axial plane one has to know whether the trajectory is being swept upwards or downwards; similarly, the 2D sweeping direction (forward or backward) determines the new 2D chord.

Aside from the advantage deriving from the information compaction, the HSS method entails a further optimization because, when combined with the POL approximation, it allows vectorizing the $\vec{P} \cdot \vec{T}$ scalar products for the set of successive chords of the same class.

Load balancing parallel sweep The transport sweep of TDT largely benefits from the OpenMP parallel programming, which is currently the only available parallel strategy in the code. In fact, trajectories are swept independently by different threads, which cumulate the corresponding ΔJ_r terms ($\Delta \vec{J}_r$ in the POL case) on private copies to avoid *race condition*, that is, multiple threads attempting to update the same value at the same time. Once each thread has finished its task, the reduction operation is performed on the shared copy, but it is evident that the frequency with which this operation is done has a negative impact on the efficiency of the parallel execution. In this perspective, it is important to assign the proper task load to each thread, aiming at minimizing the reduction frequency. First of all, each trajectory has access to a subset of the S_N directions, with the size of the ΔJ_r private copy being set accordingly; unless

all the boundaries coincide with the actual borders of the reactor, multiple *connected* angles constitute the trajectory subset, due to the reflective boundary conditions. Figure 4.5 reports the set of connected angles for a cyclic 2D trajectory. In case the set of connected angles changes while taking charge of a new trajectory, a thread is firstly obliged to reduce its private copy. Therefore, the developed load balancing algorithm assigns to each thread a package of trajectories that are likely to have the same set of connected angles: in this way, if all sets are not spread over more than one thread, the reduction will be done only once per set. Moreover, as it is customary in applications, a *greedy* algorithm is adopted, which starts charging the threads with the most computationally costly tasks and then assigns the remaining ones to the idle threads by decreasing cost.

Part II

NEW DEVELOPMENTS

Chapter 5

The polynomial cross sections method

Depleted systems are characterized by spatially dependent cross sections. This dependence arises as a consequence of the non-uniform isotopes consumption, which is related to their rates of interaction with neutrons and therefore varies in space according to the flux profile. As anticipated in the previous chapters, the customary way of dealing with spatial variations is to adopt a refined mesh while considering the quantities as constant within each regions. In the MOC framework, the simplicity of this approach is counterbalanced by heavy computational costs, due to the memory for the storage of transmission coefficients and to the large number of necessary applications of the transmission equation, during the transport sweep, and of the system of balance equations (once per region for each inner iteration).

In Beginning-Of-Cycle (BOC) conditions, material homogeneity allows enlarging the mesh size by introducing a higher-order flux description. In the TDT code, this was done along the axis of 3D axially extruded geometries [11], according to the POL method reported in section 4.4.2. Results proved the clear advantage of using the higher-order MOC, suggesting the potential convenience of extending this approach to non-zero burnup cases: this thesis takes this further step, adopting a polynomial development for cross sections and thus permitting the use of a coarser axial mesh.

Literature provides very few examples of neutronics applications not involving the constant cross sections approximation. What is probably the first work dates back to 1977 [53]: the two authors compared the results of a continuous cubic Finite-Element (FEM) diffusion method adopting a 1^{st}-order polynomial spatial profile for cross sections with the ones obtained through the piece-wise constant approach. More recently, a similar comparison has been made using the MOC [54]: cross sections are supposed to vary linearly along trajectories, but the method is limited to 2D geometries. A relevant example is given by [55], where the discontinuous FEM is applied to slab geometry without making any assumption on the shape of cross sections spatial variation. Contrary to the previously cited applications, however, the present work introduces the polynomial expansion in the analysis of more complex and heterogeneous systems, as the PWR subsystems that will be shown in section 7.1.

This chapter starts with the description of the polynomial expansion employed and the adaptation of the MOC equations relating to *free* (i.e. non-accelerated) power iterations to the polynomial cross sections case. Then, the discussion presents the vectorization strategy and the adaptive quadrature algorithm developed to optimize the transmission coefficients evaluation and to properly compute the integrals, appearing in the equations, for a which a

closed-form expression is not available, respectively. The TDT coupling with MENDEL and the APOLLO3® self-shielding module is also described, pointing out the current need for a different spatial mesh for all operations other than the flux calculation. Lastly, a simple fix-up technique, preventing the polynomial cross sections from being locally negative, is addressed.

5.1 Adaptation of MOC equations

Since the spatial profiles of the cross sections are ultimately the result of the flux dependence on position, it seemed natural to adopt for them the same expansion basis. The moments of the angular flux and the cross sections are thus both developed in terms of polynomials along the axial direction, while maintaining the SC description on the radial plane. In a completely analogous way to Eq. (4.19), the generic macroscopic cross section for reaction R and energy group g is written as

$$\Sigma_R^g(\vec{r}) \approx \vec{P}(\vec{r}) \cdot \vec{\Sigma}_{R,r}^g \qquad \vec{\Sigma}_{R,r}^g = \{\Sigma_{R,r,p}^g,\ p = 0, N_p\}, \tag{5.1}$$

with $\vec{\Sigma}_{R,r}^g$ being the vector of the cross section polynomial coefficients and $\vec{P}$ the same set of polynomials as in Eq. (4.18). As before, the group index will be omitted henceforth. The next two paragraphs present the derivation of the adapted versions of the MOC equations.

5.1.1 Transmission equation

Starting again from the analytical Eq. (4.5), one needs to consider how the optical path lengths and the emission densities modify according to the new cross sections expression. Concerning the former, the evaluation of Eq. (4.6) up to the s coordinate of a chord of entry point $\vec{r}_0$ and direction $\vec{\Omega}$ yields

$$\begin{aligned}
\tau(s) &= \int_0^s ds'\, \Sigma_T\Big(\vec{r}(s')\Big) = \int_0^s ds'\, \vec{P}\Big(\vec{r}(s')\Big) \cdot \vec{\Sigma}_{T,r} = \sum_{p=0}^{N_p} \int_0^s ds' \left(\frac{z_0 + s'\mu - \bar{z}_r}{\Delta z_r/2}\right)^p \Sigma_{T,r,p} \\
&= \sum_{i=0}^{N_p} \Lambda_{i0}\, s + \sum_{i=0}^{N_p-1} \Lambda_{i1}\, s^2 + \sum_{i=0}^{N_p-2} \Lambda_{i2}\, s^3 + \cdots + \Lambda_{0N_p}\, s^{N_p+1} \\
&= \tau_1\, s + \tau_2\, s^2 + \tau_3\, s^3 + \cdots + \tau_{N_p+1}\, s^{N_p+1},
\end{aligned} \tag{5.2}$$

where Λ's are given by

$$\Lambda_{ij} = P_i(\vec{r}_0)\, \Sigma_{T,r,i+j} \left(\frac{2\mu}{\Delta z_r}\right)^j \frac{1}{j+1} \binom{j+i}{i}. \tag{5.3}$$

An expansion of order N_p therefore leads to expressing the optical path lengths as polynomials of degree $N_p + 1$. This implies that, for any $N_p > 0$, the τ-involving integrals over the chord length of Eq. (4.5) do not have solutions that can be written in terms of elementary functions. Such integrals are computed by an adaptive quadrature formula which is the

object of section 5.3. As for the mono-group emission densities, they can be written in a scalar product form as in Eq. (4.20), but involving polynomials up to the $(2N_p)^{\text{th}}$ degree:

$$\begin{aligned} q(\vec{r},\vec{\Omega}) &= \vec{P}^{\,2N_p}(\vec{r})\cdot\vec{q}_r(\vec{\Omega}) \\ \vec{P}^{\,2N_p}(\vec{r}) &= \{(\tilde{z}_r)^p,\ p=0,2N_p\} \\ \vec{q}_r(\vec{\Omega}) &= \{q_{r,p}(\vec{\Omega}),\ p=0,2N_p\}, \end{aligned} \tag{5.4}$$

due to the dual axial expansion (flux and cross sections). Polynomial coefficients $\vec{q}_r$ are actually given by

$$\begin{aligned} q_{r,p}(\vec{\Omega}) = &\sum_{n=1}^{N_m} A_n(\vec{\Omega}) \sum_{g'} \sum_{j+k=p} \Sigma^{g'\to g}_{S,l(n),r,j}\,\Phi^{g'}_{n,r,k} \\ &+ \frac{1}{k_{\text{eff}}} \sum_i \chi^g_i \sum_{g'} \sum_{j+k=p} \nu\Sigma^{g'}_{F,i,r,j}\Phi^{g'}_{1,r,k}, \end{aligned} \tag{5.5}$$

where $\sum_{j+k=p}$ denotes the sum over components having indices j and k with sum equal to p. Besides, the spatial behavior of the average numbers of neutrons emerging from fission per unit path and for each fissile isotope, $\{\nu\Sigma^g_{F,i}(\vec{r})\}$, and of the coefficients of the scattering cross section expansion in terms of Legendre polynomials, $\{\Sigma^{g'\to g}_{S,l}(\vec{r})\}$, is described as in Eq. (5.1).

Similarly to what was done in Eq. (4.22), Eq. (5.4) can be substituted in (4.5) to obtain the transmission equation accounting for polynomial cross sections:

$$\psi^+(\vec{\Omega}) = \psi^-(\vec{\Omega})\, e^{-\tau(l)} + \vec{P}^{\,2N_p}(\vec{r}_0)\cdot\vec{T}^{\,2N_p}, \tag{5.6}$$

where

$$\begin{aligned} &\vec{T}^{\,2N_p} = \{T_k,\ k=0,2N_p\} \\ &T_k = \sum_{p=k}^{2N_p} \binom{p}{k} \left(\frac{2\mu}{\Delta z_r}\right)^{p-k} q_{r,p}(\vec{\Omega})\, E_{p-k}, \qquad E_j = \int_0^l ds\, s^j\, e^{\tau(s)-\tau(l)}. \end{aligned} \tag{5.7}$$

Comparing Eq. (5.6) to its homologous for the POL method (Eq. (4.22)), it is apparent that introducing the higher-order description also for cross sections increases the number of floating-point operations, due to the greater dimensions of arrays and, above all, to the τ dependence on z. Even though this is compensated by largely resorting to vectorization (see section 5.2), it is convenient to apply the polynomial cross sections approach only within the regions where it is actually needed, that is, where the concentrations of isotopes undergo non-negligible variations as the burnup increases, and to expand the neutron flux only elsewhere. The former regions will be referred to as *polynomial* in the following. Polynomial regions are determined at the beginning of the calculation based on their material compositions.

5.1.2 Balance equation

Angular formulation

In this paragraph an analogous equation to (4.33) is derived. As will be shown, the obtained formulation will not be suitable for the higher-order cross sections case and further

rearrangements will be needed. The starting point for the derivation is again the mono-group integro-differential transport equation (4.1), which, projected on the set of expansion polynomials $\vec{P}(\vec{r})$ and averaged over the region volume, becomes

$$\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\big[\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega})\big]+\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\,\Sigma_T(\vec{r})\,\psi(\vec{r},\vec{\Omega})=\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\,q(\vec{r},\vec{\Omega}). \quad (5.8)$$

The expression of first term is exactly the same as in the POL case, reading

$$\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\big[\vec{\Omega}\cdot\nabla\psi(\vec{r},\vec{\Omega})\big]\approx\Delta\vec{J}_r(\vec{\Omega})-\mu\,\bar{\bar{\mathcal{C}}}_r\,'\vec{\psi}_r(\vec{\Omega}). \quad (5.9)$$

On the basis of Eq. (5.4), the right-hand side requires the definition of the following rectangular matrix:

$$\bar{\bar{\mathcal{P}}}^{\,2N_p}=\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\otimes\vec{P}^{\,2N_p}(\vec{r}), \quad (5.10)$$

whose components can be analytically evaluated as

$$\mathcal{P}_{ij}=\frac{1}{V_r}\int_r d\vec{r}\,P_i(\vec{r})P_j(\vec{r})=\frac{1}{2}\int_{-1}^{1}d\tilde{z}_r\,(\tilde{z}_r)^{i+j}=\begin{cases}\dfrac{1}{i+j+1} & i+j\ even\\ 0 & i+j\ odd\end{cases}. \quad (5.11)$$

Hence, Eq. (5.10) allows writing

$$\frac{1}{V_r}\int_r d\vec{r}\,\vec{P}(\vec{r})\,q(\vec{r},\vec{\Omega})=\bar{\bar{\mathcal{P}}}^{\,2N_p}\,q_r(\vec{\Omega}). \quad (5.12)$$

As anticipated in section 4.4.2, though, matrix $\bar{\bar{\mathcal{P}}}^{2N_p}$ needs be calculated via the same trajectory-based spatial discretization used for the sweep: as discussed in [11], this is a required condition to preserve particles conservation and numerical stability, and the reason for it acquiring a dependence on direction and region. Its $\mathcal{P}_{r,pp'}$ element is actually computed as

$$\begin{aligned}\mathcal{P}_{r,pp'}(\vec{\Omega})&=\frac{1}{\tilde{V}_r(\vec{\Omega})}\int_r d\vec{r}\,P_p(\vec{r})P_{p'}(\vec{r})=\frac{1}{\tilde{V}_r(\vec{\Omega})}\int_{\partial r_\perp}d_2r_\perp\int_0^l ds\left(\frac{z_0+s\mu-\bar{z}_r}{\Delta z_r/2}\right)^{p+p'}\\&\approx\frac{\Delta_\perp(\vec{\Omega})}{\tilde{V}_r(\vec{\Omega})}\sum_{k=0}^{p}\binom{p}{k}\sum_{k'=0}^{p'}\binom{p'}{k'}\sum_{\substack{t\parallel\vec{\Omega}\\ t\cap r}}\left(\frac{z_0-\bar{z}_r}{\Delta z_r/2}\right)^{p+p'-k-k'}\left(\frac{2\mu}{\Delta z_r}\right)^{k+k'}\frac{l^{k+k'+1}}{k+k'+1},\end{aligned} \quad (5.13)$$

where $\tilde{V}_r(\vec{\Omega})$ is the region volume approximation related to direction $\vec{\Omega}$ (Eq. (4.7)) and $\int_{\partial r_\perp}$ represents the integral over the surface perpendicular to $\vec{\Omega}$ and contained in r. In fact, stability requirements involve also the $\Delta\vec{J}_r(\vec{\Omega})$ term: in its expression (4.29), $\tilde{V}_r(\vec{\Omega})$ has to be used instead of the V_r average volume over directions.

Lastly, the projected removal term is modified by using the cross sections expansion. Its p^{th} component reads

$$\begin{aligned}\frac{1}{V_r}\int_r d\vec{r}\,P_p(\vec{r})\,\Sigma_T(\vec{r})\,\psi(\vec{r},\vec{\Omega})&=\frac{1}{V_r}\int_r d\vec{r}\,P_p(\vec{r})\left(\sum_{k=0}^{N_p}P_k(\vec{r})\,\Sigma_{T,r,k}\right)\psi(\vec{r},\vec{\Omega})\\&=\sum_{k=0}^{N_p}\Sigma_{T,r,k}\,'\psi_{r,p+k}(\vec{\Omega}),\end{aligned} \quad (5.14)$$

so that the p^{th} projection of Eq. (4.1), in terms of the ${}'\vec{\psi}_r$ polynomial moments of the angular flux, can be written as

$$\sum_{k=0}^{N_p} \Sigma_{T,r,k}\, {}'\psi_{r,p+k}(\vec{\Omega}) - p\,\frac{2\mu}{\Delta z_r}\, {}'\psi_{r,p-1}(\vec{\Omega}) = \left(\bar{\bar{\mathcal{P}}}_r^{\,2N_p}(\vec{\Omega})\,\vec{q}_r(\vec{\Omega})\right)_p - \Delta J_{r,p}(\vec{\Omega}). \tag{5.15}$$

Looking at the equation above, not only does it constitute no more a sub-triangular system alongside with the other projections, as in the POL method, but now the whole system contains also more unknowns than equations: $2N_p+1$ polynomial moments with only N_p+1 equations. The adopted solution is then to pass directly to the polynomial coefficients $\vec{\psi}_r$, without computing the moments beforehand. To this purpose, it is convenient to introduce matrix $\bar{\bar{\mathcal{P}}}$, defined as the transpose of matrix (5.10):

$$\bar{\bar{\mathcal{P}}} = \left(\bar{\bar{\mathcal{P}}}^{\,2N_p}\right)^T = \frac{1}{V_r}\int_r d\vec{r}\,\vec{P}^{\,2N_p}(\vec{r}) \otimes \vec{P}(\vec{r}). \tag{5.16}$$

It is therefore possible to link the $2N_p+1$ flux polynomial moments to the N_p+1 coefficients, similarly to Eq. (4.34):

$${}'\vec{\psi}_r(\vec{\Omega}) = \frac{1}{V_r}\int_r d\vec{r}\,\vec{P}^{\,2N_p}(\vec{r})\,\psi(\vec{r},\vec{\Omega}) = \bar{\bar{\mathcal{P}}}\,\vec{\psi}_r(\vec{\Omega}). \tag{5.17}$$

It follows that ${}'\psi_{r,i}(\vec{\Omega}) = \sum_{j=0}^{N_p}\mathcal{P}_{ij}\,\psi_{r,j}(\vec{\Omega})$, which can be substituted in Eq. (5.15) to finally obtain

$$\sum_{k=0}^{N_p}\Sigma_{T,r,k}\sum_{j=0}^{N_p}\mathcal{P}_{(p+k)j}\,\psi_{r,j}(\vec{\Omega}) - p\frac{2\mu}{\Delta z_r}\sum_{j=0}^{N_p}\mathcal{P}_{(p-1)j}\,\psi_{r,j}(\vec{\Omega}) = \left(\bar{\bar{\mathcal{P}}}_r^{\,2N_p}(\vec{\Omega})\,\vec{q}_r(\vec{\Omega})\right)_p - \Delta J_{r,p}(\vec{\Omega}). \tag{5.18}$$

Coupled with the other N_p projections, this equation forms a solvable system having only the $\vec{\psi}_r$ flux coefficients as unknowns.

Flux moments formulation

Equation (5.18) can be used in combination with a vectorized form of Eq. (4.17), involving all the polynomial terms. However, solving the system for all the quadrature angles and then retrieving one by one the angular moments is more computationally expensive than solving directly a system for the latter, as in general the number of angles is much greater than N_m. Deriving a balance formula for the angular moments is convenient also in view of the acceleration technique (see chapter 6), where one has to resort to a moment-based relation (similar to the one presented here), as no use of the angular flux information is made in that framework.

In order to obtain the sought balance relation, all projections (5.18) have to be further projected on the real spherical harmonics by applying operator $\oint \frac{d\vec{\Omega}}{4\pi}\,A_l^m(d\vec{\Omega})*$ for all the considered angular moments. From now on, a bold notation for vectors and matrices will indicate that they involve, or are combinations of vectors that involve, both polynomial and

angular components. For instance, $\vec{\mathbf{\Phi}}_r$ is the vector of all polynomial coefficients of every angular moment of the flux, given by

$$\vec{\mathbf{\Phi}}_r = \left\{ \vec{\Phi}_{n,r},\, n = 1, N_m \right\}, \tag{5.19}$$

where $\vec{\Phi}_{n,r}$ has been defined in Eq. (4.19). Before projecting on harmonics, it is useful to introduce the two vectors

$$\begin{aligned} \vec{\mathbf{Z}}_r(\vec{\Omega}) &= \left\{ A_n(\vec{\Omega})\, \vec{\mathbb{P}}_r(\vec{\Omega}),\, n = 1, N_m \right\} \\ \vec{\mathbf{Z}}_r^{2N_p}(\vec{\Omega}) &= \left\{ A_n(\vec{\Omega})\, \vec{\mathbb{P}}_r^{2N_p}(\vec{\Omega}),\, n = 1, N_m \right\}, \end{aligned} \tag{5.20}$$

with $\vec{\mathbb{P}}_r$ and $\vec{\mathbb{P}}_r^{2N_p}$ being the vectors producing matrix $\bar{\vec{\mathcal{P}}}_r^{2N_p}$ through the tensorial product

$$\bar{\vec{\mathcal{P}}}_r^{2N_p}(\vec{\Omega}) = \vec{\mathbb{P}}_r(\vec{\Omega}) \otimes \vec{\mathbb{P}}_r^{2N_p}(\vec{\Omega}), \tag{5.21}$$

so as to define matrix $\boldsymbol{\mathcal{Z}}_r$ as

$$\boldsymbol{\mathcal{Z}}_r = \oint \frac{d\vec{\Omega}}{4\pi}\, \vec{\mathbf{Z}}_r(\vec{\Omega}) \otimes \vec{\mathbf{Z}}_r^{2N_p}(\vec{\Omega}). \tag{5.22}$$

Besides, the following property of spherical harmonics comes in handy [56]:

$$\mu A_l^m(\vec{\Omega}) = \alpha_{l+1}^m A_{l+1}^m(\vec{\Omega}) + \beta_{l-1}^m A_{l-1}^m(\vec{\Omega}), \tag{5.23}$$

where

$$\alpha_{l+1}^m = \frac{\sqrt{(l+m+1)(l-m+1)}}{2l+1} \qquad \beta_{l-1}^m = \begin{cases} \frac{\sqrt{(l+m)(l-m)}}{2l+1} & |m| \leq l-1 \\ 0 & |m| > l-1. \end{cases} \tag{5.24}$$

By further defining vector $\vec{\Delta}_{l,r}^m = \oint \frac{d\vec{\Omega}}{4\pi}\, A_l^m(\vec{\Omega})\, \Delta\vec{J}_r(\vec{\Omega})$, of components $\{\Delta_{l,r,p}^m\}$ corresponding to the $N_p + 1$ elements of $\Delta\vec{J}_r$, one can finally express the projection of the whole set of equations (5.18), relating to all polynomial coefficients, on the A_l^m harmonic as

$$\begin{aligned} \sum_{k=0}^{N_p} \Sigma_{T,r,k} \sum_{j=0}^{N_p} \mathcal{P}_{(p+k)j}\, \Phi_{l,r,j}^m - \frac{2p}{\Delta z_r} \left(\alpha_{l+1}^m \sum_{j=0}^{N_p} \mathcal{P}_{(p-1)j}\, \Phi_{l+1,r,j}^m + \beta_{l-1}^m \sum_{j=0}^{N_p} \mathcal{P}_{(p-1)j}\, \Phi_{l-1,r,j}^m \right) \\ = \left(\boldsymbol{\mathcal{Z}}_r\, \vec{\boldsymbol{q}}_r \right)_{l,\,p}^m - \Delta_{l,r,p}^m, \qquad p = 0, N_p, \end{aligned} \tag{5.25}$$

where $\vec{\boldsymbol{q}}_r$ contains all the angular and the polynomial components of the emission densities:

$$\vec{\boldsymbol{q}}_r = \left\{ \oint \frac{d\vec{\Omega}}{4\pi}\, A_n(\vec{\Omega})\, \vec{q}_r(\vec{\Omega}),\, n = 1, N_m \right\}. \tag{5.26}$$

System (5.25) is coupled, through the second term of the left-hand side, to equivalent sets of equations for the other moments. However, since this coupling regards only one index

m at a time, the global system of $N_m(N_p+1)$ equations can be decomposed into smaller subsystems: this is done by defining matrices

$$\begin{aligned}\bar{\bar{G}}_r &= \Big\{G_{pi},\ p=0,N_p,\ i=0,N_p\Big\}, \qquad G_{pi} = \sum_{k=0}^{N_p} \Sigma_{T,r,k}\mathcal{P}_{(p+k)i} \\ \bar{\bar{H}}_r &= \Big\{H_{pi},\ p=0,N_p,\ i=0,N_p\Big\}, \qquad H_{pi} = -\frac{2p}{\Delta z_r}\mathcal{P}_{(p-1)i},\end{aligned} \tag{5.27}$$

so that, for each m, the obtained subsystem reads

$$\begin{pmatrix} \bar{\bar{G}}_r & \alpha_1^m\bar{\bar{H}}_r & 0 & \dots & \dots \\ \beta_0^m\bar{\bar{H}}_r & \bar{\bar{G}}_r & \alpha_2^m\bar{\bar{H}}_r & 0 & \dots \\ 0 & \beta_1^m\bar{\bar{H}}_r & \bar{\bar{G}}_r & \alpha_3^m\bar{\bar{H}}_r & \ddots \\ \vdots & 0 & \beta_2^m\bar{\bar{H}}_r & \bar{\bar{G}}_r & \ddots \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix} \begin{bmatrix} \vec{\Phi}^m_{0,r} \\ \vec{\Phi}^m_{1,r} \\ \vec{\Phi}^m_{2,r} \\ \vec{\Phi}^m_{3,r} \\ \vdots \end{bmatrix} = \begin{bmatrix} \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^m_0 \\ \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^m_1 \\ \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^m_2 \\ \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^m_3 \\ \vdots \end{bmatrix} - \begin{bmatrix} \vec{\Delta}^m_{0,r} \\ \vec{\Delta}^m_{1,r} \\ \vec{\Delta}^m_{2,r} \\ \vec{\Delta}^m_{3,r} \\ \vdots \end{bmatrix}, \tag{5.28}$$

which therefore comprises $(K-|m|+1)(N_p+1)$ equations, one for each admissible value of l for the given m, multiplied by the number of polynomial coefficients. The advantage of this strategy is related to the strong reduction of floating-point operations (flops) necessary for matrices inversion: for instance, if $K=3$, $N_p=2$ and the cost of inverting a square matrix of rank n is assumed to be $2/3\ n^3$ flops, 2448 flops are required to invert the matrices of the 7 subsystems, whereas 73 728 would be needed for one huge matrix applied to the whole vector $\vec{\boldsymbol{\Phi}}_r$. The 7 subsystems of the example are

$$\bar{\bar{G}}_r\,\vec{\Phi}^{-3}_{3,r} = \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^{-3}_3 - \vec{\Delta}^{-3}_{3,r} \tag{5.29a}$$

$$\begin{pmatrix} \bar{\bar{G}}_r & \alpha_3^{-2}\bar{\bar{H}}_r \\ \beta_2^{-2}\bar{\bar{H}}_r & \bar{\bar{G}}_r \end{pmatrix} \begin{bmatrix} \vec{\Phi}^{-2}_{2,r} \\ \vec{\Phi}^{-2}_{3,r} \end{bmatrix} = \begin{bmatrix} \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^{-2}_2 \\ \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^{-2}_3 \end{bmatrix} - \begin{bmatrix} \vec{\Delta}^{-2}_{2,r} \\ \vec{\Delta}^{-2}_{3,r} \end{bmatrix} \tag{5.29b}$$

$$\begin{pmatrix} \bar{\bar{G}}_r & \alpha_2^{-1}\bar{\bar{H}}_r & 0 \\ \beta_1^{-1}\bar{\bar{H}}_r & \bar{\bar{G}}_r & \alpha_3^{-1}\bar{\bar{H}}_r \\ 0 & \beta_2^{-1}\bar{\bar{H}}_r & \bar{\bar{G}}_r \end{pmatrix} \begin{bmatrix} \vec{\Phi}^{-1}_{1,r} \\ \vec{\Phi}^{-1}_{2,r} \\ \vec{\Phi}^{-1}_{3,r} \end{bmatrix} = \begin{bmatrix} \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^{-1}_1 \\ \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^{-1}_2 \\ \big(\boldsymbol{\mathcal{Z}}_r\,\vec{\boldsymbol{q}}_r\big)^{-1}_3 \end{bmatrix} - \begin{bmatrix} \vec{\Delta}^{-1}_{1,r} \\ \vec{\Delta}^{-1}_{2,r} \\ \vec{\Delta}^{-1}_{3,r} \end{bmatrix} \tag{5.29c}$$

$$\begin{pmatrix} \bar{\bar{G}}_r & \alpha_1^0 \bar{\bar{H}}_r & 0 & 0 \\ \beta_0^0 \bar{\bar{H}}_r & \bar{\bar{G}}_r & \alpha_2^0 \bar{\bar{H}}_r & 0 \\ 0 & \beta_1^0 \bar{\bar{H}}_r & \bar{\bar{G}}_r & \alpha_3^0 \bar{\bar{H}}_r \\ 0 & 0 & \beta_2^0 \bar{\bar{H}}_r & \bar{\bar{G}}_r \end{pmatrix} \begin{bmatrix} \vec{\Phi}_{0,r}^0 \\ \vec{\Phi}_{1,r}^0 \\ \vec{\Phi}_{2,r}^0 \\ \vec{\Phi}_{3,r}^0 \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_0^0 \\ \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_1^0 \\ \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_2^0 \\ \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_3^0 \end{bmatrix} - \begin{bmatrix} \vec{\Delta}_{0,r}^0 \\ \vec{\Delta}_{1,r}^0 \\ \vec{\Delta}_{2,r}^0 \\ \vec{\Delta}_{3,r}^0 \end{bmatrix} \tag{5.29d}$$

$$\begin{pmatrix} \bar{\bar{G}}_r & \alpha_2^1 \bar{\bar{H}}_r & 0 \\ \beta_1^1 \bar{\bar{H}}_r & \bar{\bar{G}}_r & \alpha_3^1 \bar{\bar{H}}_r \\ 0 & \beta_2^1 \bar{\bar{H}}_r & \bar{\bar{G}}_r \end{pmatrix} \begin{bmatrix} \vec{\Phi}_{1,r}^1 \\ \vec{\Phi}_{2,r}^1 \\ \vec{\Phi}_{3,r}^1 \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_1^1 \\ \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_2^1 \\ \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_3^1 \end{bmatrix} - \begin{bmatrix} \vec{\Delta}_{1,r}^1 \\ \vec{\Delta}_{2,r}^1 \\ \vec{\Delta}_{3,r}^1 \end{bmatrix} \tag{5.29e}$$

$$\begin{pmatrix} \bar{\bar{G}}_r & \alpha_3^2 \bar{\bar{H}}_r \\ \beta_2^2 \bar{\bar{H}}_r & \bar{\bar{G}}_r \end{pmatrix} \begin{bmatrix} \vec{\Phi}_{2,r}^2 \\ \vec{\Phi}_{3,r}^2 \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_2^2 \\ \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_3^2 \end{bmatrix} - \begin{bmatrix} \vec{\Delta}_{2,r}^2 \\ \vec{\Delta}_{3,r}^2 \end{bmatrix} \tag{5.29f}$$

$$\bar{\bar{G}}_r \, \vec{\Phi}_{3,r}^3 = \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_3^3 - \vec{\Delta}_{3,r}^3 \, . \tag{5.29g}$$

In order to obtain a more compact formulation of the set of balance equations, the following vectors can be introduced:

$$\begin{aligned} \vec{\boldsymbol{\Phi}}_r^{\,m} &= \left\{ \vec{\Phi}_{l,r}^m, \; l = |m|, K \right\} \\ \vec{\boldsymbol{Q}}_r^{\,m} &= \left\{ \left(\boldsymbol{\mathcal{Z}}_r \, \vec{\boldsymbol{q}}_r\right)_l^m, \; l = |m|, K \right\} \\ \vec{\boldsymbol{\Delta}}_r^{\,m} &= \left\{ \vec{\Delta}_{l,r}^m, \; l = |m|, K \right\}, \end{aligned} \tag{5.30}$$

which finally allows expressing each m-subsystem as

$$\boldsymbol{\Gamma}_r^m \vec{\boldsymbol{\Phi}}_r^{\,m} = \vec{\boldsymbol{Q}}_r^{\,m} - \vec{\boldsymbol{\Delta}}_r^{\,m}, \tag{5.31}$$

$\boldsymbol{\Gamma}_r^m$ being the matrix on the left-hand side of Eq. (5.28). One can exploit its tridiagonal block structure to define the content of each of its $(K - |m| + 1)^2$ blocks: denoting by b_R and b_C the index of the row and column block, respectively, one can write

$$\left(\boldsymbol{\Gamma}_r^m\right)_{b_R, b_C} = \begin{cases} \bar{\bar{G}}_r & b_R = b_C \\ \alpha_{|m|+b_R}^m \bar{\bar{H}}_r & b_R = b_C - 1 \\ \beta_{|m|+b_C-1}^m \bar{\bar{H}}_r & b_R = b_C + 1 \\ 0 & \text{otherwise} \end{cases} , \tag{5.32}$$

keeping in mind that block (b_R, b_C) comprises the elements from row/column $1 + (b_{R/C} - 1)(N_p + 1)$ to $b_{R/C}(N_p + 1)$.

5.2 Vectorization of transport coefficients

In terms of per-region computing resources, the polynomial cross sections method is more expensive than the POL approach: with reference to Eq. (5.6), the τ dependence on z requires the calculation of more $e^{-\tau}$ and $\vec{T}^{2N_p}$ transport coefficients. As discussed in section 4.4.3, one can largely benefit from the CCM when computing these terms, since the classification allows for strong flops and memory reductions. In the development of the present method the aim was therefore to adhere as much as possible to the already implemented strategy, with the hope to attain at least partially the efficiency of the POL case.

To begin with, the H-chord coefficients evaluation is optimized exactly in the same way as in the previous method: for a given energy group, the transport coefficients of all the H-chords of a class (identified by the 3D region and the polar angle) can be calculated at once. One may consider it as an example of ideal vectorization, as the flop-reduction factor is equal to the chords number.

On the other hand, each V-chord contained in a polynomial region has its own optical path length and, therefore, transport coefficients. The optimization is not totally compromised, though: since the V-chords of a class have the same length and cross section coefficients, it is actually possible and convenient to vectorize the evaluation of their τ-dependent coefficients $e^{-\tau}$ and $\{E_j\}$ (Eq. (5.7)) using the OpenMP SIMD pragma for each innermost loop, which is particularly advantageous when loops are made over the number of chords. Algorithms 2 and 3 describe how the τ coefficients and the $\{E_j\}$ integrals are computed; for the latter a Gauss-Legendre based quadrature formula is used that will be presented in the next section. Since hundreds of chords may constitute the vectorized loops, this approach leads to quite a significant performance gain, even if far from the H-chords ideal condition. The results concerning the vectorization efficiency will be shown in section 7.2.3, along with the ones regarding the vectorization of acceleration coefficients (see section 6.3).

The M-chords are treated independently of each other, like in the POL case, calculating their transport coefficients on-the-fly as they are met during the sweep.

As a final note, even if the ideal vectorization strategy (1 class $\leftrightarrow$ 1 coefficient per kind and energy group) could be followed within polynomial regions, the computational burden of the new method would still be greater than with a constant cross section description: therefore, as already mentioned at the end of paragraph 5.1.1, the higher-order approach is adopted only in the regions where nuclides undergo a non-negligible depletion. Moreover, the transport coefficients relating to the chords of polynomial regions are stored in specific arrays whose dimensions correspond exactly to the number of such chords: the right values are then retrieved using suitably built arrays of integer addresses, while sweeping the trajectories. The scope is to limit the impact on memory which is, once again, heavier than in the POL method.

Algorithm 2 Vectorized computation of optical path lengths coefficients

1: ! $\tau(s) = \tau_1\, s + \tau_2\, s^2 + \cdots + \tau_{N_p+1}\, s^{N_p+1}$ (Eq. (5.2))
2:
3: $\tau_1(1 : N_{\text{V-chords}}) = \Sigma_{T,0}$
4: **for** $i = 1, N_p$ **do**
5: !$OMP SIMD
6: **for** $ic = 1, N_{\text{V-chords}}$ **do**
7: $\tau_1(ic) = \tau_1(ic) + P_i[\vec{r}_0(ic)]\, \Sigma_{T,i}$
8: **end for**
9: **end for**
10: **for** $p = 1, N_p$ **do**
11: !$OMP SIMD
12: **for** $ic = 1, N_{\text{V-chords}}$ **do**
13: $\tau_{p+1}(ic) = \Sigma_{T,p}$
14: **end for**
15: **for** $i = 1, N_p - p$ **do**
16: $aux = \binom{i+p}{i}\, \Sigma_{T,i+p}$
17: !$OMP SIMD
18: **for** $ic = 1, N_{\text{V-chords}}$ **do**
19: $\tau_{p+1}(ic) = \tau_{p+1}(ic) + P_i[\vec{r}_0(ic)]\, aux$
20: **end for**
21: **end for**
22: !$OMP SIMD
23: **for** $ic = 1, N_{\text{V-chords}}$ **do**
24: $\tau_{p+1}(ic) = \tau_{p+1}(ic)\Big(\frac{2\mu}{\Delta z_r}\Big)^p$
25: **end for**
26: **end for**

Legend

- $\tau_i(ic)$: i^{th}-order polynomial coefficient of the optical path length of chord ic;
- $\Sigma_{T,i}$: i^{th}-order polynomial coefficient of the macroscopic total cross section (since only one region is considered, the index is omitted);
- $N_{\text{V-chords}}$: number of V-chords contained in the considered set;
- $P_i[\vec{r}_0(ic)]$: i^{th}-degree polynomial evaluated at the entry point of chord ic.

Algorithm 3 Vectorized computation of transmission integrals (*Part 1*)

1: ! $E_p = \int_0^l ds\, s^p\, e^{\tau(s)-\tau(l)}$ (Eq. (5.7))
2: ! Gauss-Legendre quadrature formula of order m:
3: ! $E_p \approx \frac{l}{2} \sum_{j=1}^{m} w_j\, s_j^p\, e^{\tau(s_j)-\tau(l)}$
4:
5: ! Evaluation of Gauss-Legendre abscissae $\{s_j^p\}$ in $[0, l]$
6: !$OMP SIMD
7: **for** $j = 1, m$ **do**
8: $w^*(j, 1) = \frac{l}{2}(1 + x_j)$
9: **end for**
10: **for** $p = 2, 2N_p$ **do**
11: !$OMP SIMD
12: **for** $j = 1, m$ **do**
13: $w^*(j, p) = w^*(j, p-1)\, w^*(j, 1)$
14: **end for**
15: **end for**
16:
17: ! Calculation of $\{s_j^p - l^p\}$, to be used in
18: ! $\tau(s_j) - \tau(l) = \tau_1(s_j - l) + \tau_2(s_j^2 - l^2) + \cdots + \tau_{N_p+1}(s_j^{N_p+1} - l^{N_p+1})$ (Eq. (5.2))
19: **for** $p = 1, N_p + 1$ **do**
20: !$OMP SIMD
21: **for** $j = 1, m$ **do**
22: $s^*(j, p) = w^*(j, p) - l^p$
23: **end for**
24: **end for**
25:
26: ! Calculation of $\{\frac{l}{2}\, w_j\, s_j^p\}$
27: !$OMP SIMD
28: **for** $j = 1, m$ **do**
29: $w^*(j, 0) = \frac{l}{2}\, w_j$
30: **end for**
31: **for** $p = 1, 2N_p$ **do**
32: !$OMP SIMD
33: **for** $j = 1, m$ **do**
34: $w^*(j, p) = w^*(j, p)\, w^*(j, 0)$
35: **end for**
36: **end for**

Algorithm 3 Vectorized computation of transmission integrals (*Part 2*)

37: ! For all V-chords, evaluation of
38: ! $e^{\tau(s_j)-\tau(l)} = e^{\tau_1(s_j-l)+\tau_2(s_j^2-l^2)+\cdots+\tau_{N_p+1}\left(s_j^{N_p+1}-l^{N_p+1}\right)}$
39: $aux(1:N_{\text{V-chords}}, 1:m) = 0.$
40: **for** $j = 1, m$ **do**
41: **for** $p = 1, N_p + 1$ **do**
42: !$OMP SIMD
43: **for** $ic = 1, N_{\text{V-chords}}$ **do**
44: $aux(ic, j) = aux(ic, j) + \tau_p(ic)\, s^*(j, p)$
45: **end for**
46: **end for**
47: !$OMP SIMD
48: **for** $ic = 1, N_{\text{V-chords}}$ **do**
49: $aux(ic, j) = \exp\left[aux(ic, j)\right]$
50: **end for**
51: **end for**
52:
53: ! Quadrature sums to get $\{E_p\}$
54: $E_{0:2N_p}(1:N_{\text{V-chords}}) = 0.$
55: **for** $p = 0, 2N_p$ **do**
56: **for** $j = 1, m$ **do**
57: !$OMP SIMD
58: **for** $ic = 1, N_{\text{V-chords}}$ **do**
59: $E_p(ic) = E_p(ic) + w^*(j, p)\, aux(ic, j)$
60: **end for**
61: **end for**
62: **end for**

Legend

- l: chord length;
- x_j: j^{th} Gauss-Legendre abscissa in $[-1, 1]$;
- w_j: j^{th} Gauss-Legendre weight;
- $\tau_p(ic)$: p^{th}-order polynomial coefficient of the optical path length of chord ic;
- $N_{\text{V-chords}}$: number of V-chords contained in the considered set;
- $E_p(ic)$: p^{th} transmission integral for chord ic.

5.3 Adaptive Gauss-Legendre quadrature

The polynomial expansion of cross sections determines the appearance of terms for which a closed-form expression is not available. As anticipated in the previous section and shown in Algorithm 3, the resolution of the $\{E_j\}$ integrals of Eq. (5.7) is therefore carried out by means of a Gauss-Legendre quadrature rule [57]. Actually, the possibility of converting the integrands into functions whose integrals can be expressed in closed-form was initially investigated and implemented, but then abandoned in favor of the quadrature; the approach can be found in Appendix A. Adopting an m^{th}-order formula, the approximation of E_j reads

$$E_j = \int_0^l ds\, s^j\, e^{\tau(s)-\tau(l)} \approx \frac{l}{2} \sum_{k=1}^{m} w_k \left[s(x_k)\right]^j e^{\tau[s(x_k)]-\tau(l)}, \tag{5.33}$$

$\{x_k\}$ being the roots of the m^{th}-degree Legendre polynomial, $\{w_k\}$ the weights associated to the chosen quadrature order and $s(x_k) = \frac{l}{2}(x_k + 1)$. This formula permits integrating polynomials up to degree $2m-1$ exactly; however, since it has to be applied to all the chords contained in polynomial regions and for all energy groups, an adaptive technique is required to determine the (lowest) number of points that leads, for all of them, to an estimation of the integral in accordance with a prescribed tolerance. To do that, the actual flux calculation is preceded by a trajectory sweep during which, for each chord and energy group, the suitable quadrature order is iteratively determined and stored for later use: starting from the smallest admitted order, estimations of the integral are obtained using a greater and greater number of points until the relative difference between two successive results falls below the tolerance. This procedure is again optimized by resorting to the CCM: the storing of the m's is done per chord class so as to reduce the memory usage, and clearly it is limited to the regions containing evolving media.

Applying this strategy showed that, although for the vast majority of chords and energy group an order $m = 6$ is more than enough, a very large number of points is needed for some. Therefore, to avoid having to deal with too many different quadrature formulas, a maximum value is fixed for the order and, when this is not sufficient, the initial interval of integration is subdivided into an increasing number of sub-intervals wherein the maximum order formula is always applied. In doing so, a further chords classification is introduced, labeling as *pathological* (for a particular energy group) a chord requiring the interval subdivision; the pathological classes are therefore distinguished by their number, which indicates how many additional applications of the quadrature rule are needed to compute the integrals for the considered chord and group (or, in other terms, the required number of sub-intervals minus 1). For instance, denoting by $\mathcal{C}$ the class, E_j is approximated as

$$E_j = \sum_{c=1}^{\mathcal{C}+1} \int_{\frac{c-1}{\mathcal{C}+1}l}^{\frac{c}{\mathcal{C}+1}l} ds_c\, s_c^j\, e^{\tau(s_c)-\tau(l)} \approx \frac{l}{2(\mathcal{C}+1)} \sum_{c=1}^{\mathcal{C}+1} \sum_{k=1}^{m} w_k \left[s_c(x_k)\right]^j e^{\tau[s_c(x_k)]-\tau(l)}, \tag{5.34}$$

where $s_c(x_k) = \frac{l}{\mathcal{C}+1}\Big(c - 1 + \frac{x_k+1}{2}\Big)$.

The sweep in charge of calculating the quadrature orders is also parallelized, using the OpenMP pragmas, in the very same way as the subsequent flux calculation, by means of the load-balancing algorithm described in paragraph 4.4.3. The developed adaptive quadrature

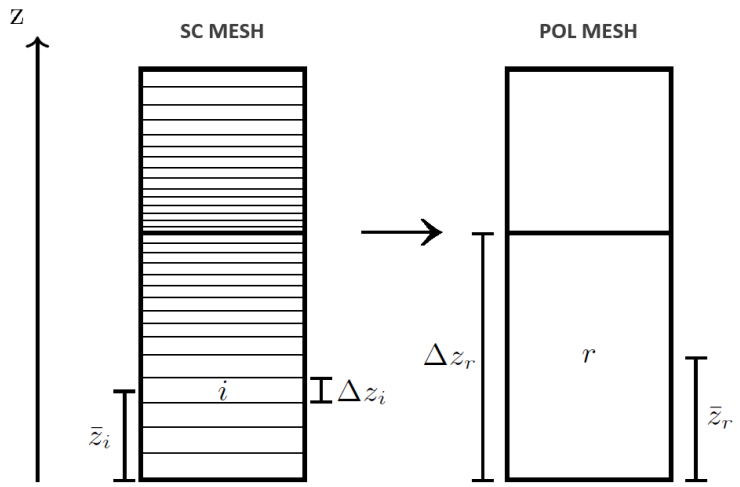


Figure 5.1: Basic examples of step-constant and corresponding polynomial meshes.

can also be intended as a validation procedure, providing a reference for calculations adopting a fixed quadrature order for all chords and energy groups. This latter approach only leads to slight reductions in memory and computing time, though. Section 7.3 will show in detail the orders required for the largest of the case studies considered.

5.4 Axial meshes for depletion study

As discussed in sections 3.1.3 and 3.4, while performing a depletion analysis TDT needs to interface with the self-shielding module and MENDEL, responsible for the actual depletion, and both can only handle spatially constant quantities. Therefore, the coupling strategy adopted envisages the use of both a step-constant and a polynomial mesh, named after the spatial approximations employed for the flux and the cross sections: starting from self-shielded macroscopic cross sections values which are constant in each region i of the former mesh, the polynomial coefficients for the corresponding region r of the coarser mesh are retrieved imposing the conservation of polynomial moments $'\vec{\Sigma}_r$:

$$\begin{aligned} \vec{\Sigma}_r &= \bar{\bar{\mathcal{P}}}_{Sq}^{-1} \, '\vec{\Sigma}_r \\ '\vec{\Sigma}_r &= \frac{1}{V_r} \int_r d\vec{r} \, \vec{P}(\vec{r}) \, \Sigma(\vec{r}) = \frac{1}{\Delta z_r} \sum_i \Sigma_i \int_{\bar{z}_i - \Delta z_i/2}^{\bar{z}_i + \Delta z_i/2} dz \, \vec{P}(\vec{r}). \end{aligned} \tag{5.35}$$

In the previous relation $\bar{\bar{\mathcal{P}}}_{Sq}$ is the square version of matrix $\bar{\bar{\mathcal{P}}}$, with rank $N_p + 1$, whereas the definition of the coordinates can be found in Fig. 5.1. After performing the flux calculation on the polynomial mesh, the flux is converted into its step-constant equivalent by the inverse procedure of the cross sections conversion (which is described in [11]), and received as input by the depletion solver, MENDEL. This is notably different from what is done in [55] where, in a FEM framework, the polynomial description is extended to nuclide concentrations.

A negative side effect is given by the macrolibrary size: since it contains all the properties

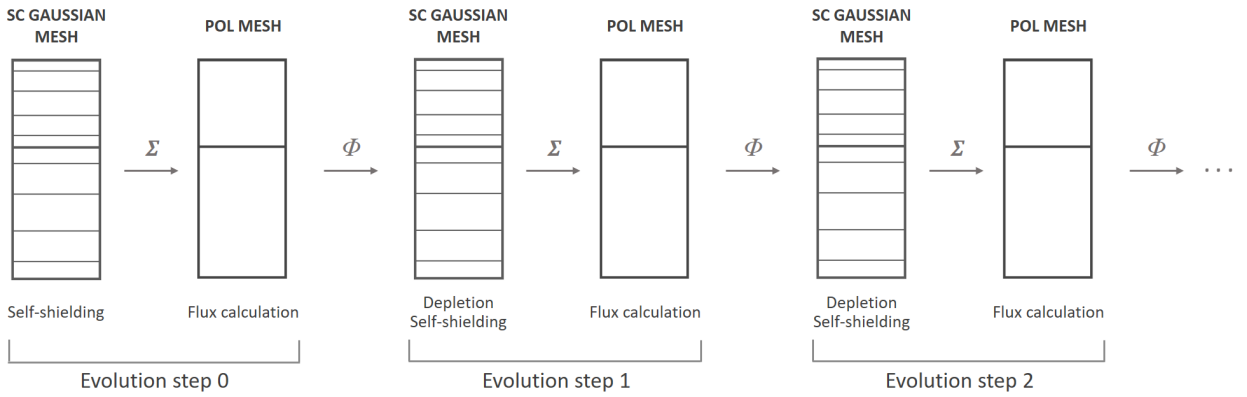


Figure 5.2: Strategy followed while performing a depletion study. Simplified views of the axial meshes are shown.

of the media[1] defined in the system, for the proposed method it would necessarily be greater than for a purely SC simulation, as it would be constituted by the sum of the macrolibrary of the step-constant mesh and the obtained cross sections coefficients. However, the fact that both depletion and self-shielding involve only intensive quantities implies that the refined mesh can be replaced by a coarser one, provided that it is capable of conserving the expected number of polynomial moments. Considering for each region r of the polynomial mesh a set of m axial layers of thicknesses proportional to the Gauss-Legendre weights, a "Gaussian" mesh is generated, with m being the order of the quadrature rule; from the moments conservation requirement it follows that $2m-1 \geq 2N_p \Rightarrow m > N_p$. This condition has always been met in the applications. Introducing the Gauss-Legendre quadrature, the expression of the p^{th} polynomial moment reads

$$'\Sigma_{r,p} = \frac{1}{2}\int_{-1}^{1} d\tilde{z}\,(\tilde{z})^p\,\Sigma(\tilde{z}) = \frac{1}{2}\sum_{i=1}^{m} w_i\,(x_i)^p\,\Sigma_i. \tag{5.36}$$

Then, after the flux calculation, the values of constant angular flux moments on the Gaussian mesh are set to those of the polynomial flux evaluated in the Gauss-Legendre abscissas $\{x_i\}$:

$$\Phi_i^n = \vec{P}(\tilde{z}=x_i)\cdot\vec{\Phi}_r^n = \Phi_{r,0}^n + x_i\Phi_{r,1}^n + x_i^2\Phi_{r,2}^n + \cdots + x_i^{N_p}\Phi_{r,N_p}^n. \tag{5.37}$$

Besides reducing the macrolibrary size, the introduction of the Gaussian mesh has therefore the (minor) advantage of accelerating the polynomial coefficients evaluation, as the smaller number of layers corresponds to fewer flops needed in Eq. (5.36) with respect to (5.35).

Figure 5.2 represents schematically how depletion, self-shielding and flux calculation are assigned to the mesh in charge at each burnup level, as well as the information required by the next operation (step-constant macroscopic cross sections or polynomial coefficients of per-region average angular flux moments), according to the presented strategy.

[1] A medium is defined as the ensemble of material properties being associated to one or more computational region. Two distinct media are treated separately by self-shielding and depletion.

5.5 Fix-up of cross sections coefficients

The way cross sections polynomial coefficients are obtained starting from the constant values of a support Gaussian mesh has been described in the previous section. It is however possible that the coefficients thus computed produce, locally, a negative axial profile, especially on region ends ($\tilde{z}_r = \pm 1$). To avoid this eventuality, a simple fix-up operation is used: if the resulting cross section $\Sigma(\tilde{z})$, of coefficients $\{\Sigma_p\}$ (omitting here the indices for the region and kind of reaction), is negative either for $\tilde{z}_r = -1$ or $\tilde{z}_r = 1$, $\Sigma(\tilde{z})$ is replaced by a new profile $\Sigma^*(\tilde{z})$, of coefficients $\{\Sigma^*_p\}$, according to the following $N_p + 1$ conditions

$$\begin{aligned}
&1. \quad \Sigma^*(\pm 1) = \sum_{p=0}^{N_p} (\pm 1)^p \, \Sigma^*_p = 0 \\
&2. \quad \int_{-1}^{1} d\tilde{z} \, \Sigma^*(\tilde{z}) = \int_{-1}^{1} d\tilde{z} \, \Sigma(\tilde{z}) \\
&3. \quad \int_{-1}^{1} d\tilde{z} \, \tilde{z} \, \Sigma^*(\tilde{z}) = \int_{-1}^{1} d\tilde{z} \, \tilde{z} \, \Sigma(\tilde{z}) \\
&4. \quad \int_{-1}^{1} d\tilde{z} \, \tilde{z}^2 \, \Sigma^*(\tilde{z}) = \int_{-1}^{1} d\tilde{z} \, \tilde{z}^2 \, \Sigma(\tilde{z}) \\
&\vdots \\
&N_p + 1. \quad \int_{-1}^{1} d\tilde{z} \, \tilde{z}^{N_p - 1} \, \Sigma^*(\tilde{z}) = \int_{-1}^{1} d\tilde{z} \, \tilde{z}^{N_p - 1} \, \Sigma(\tilde{z}).
\end{aligned} \tag{5.38}$$

This corresponds to set to 0 the negative extreme and to conserve the moments given by the previously obtained coefficients, up to the $(N_p - 1)^{\text{th}}$ order. For instance, if $N_p = 2$ and $\Sigma(-1) < 0$ the conditions will read

$$\begin{aligned}
&1. \quad \Sigma^*(-1) = 0 \iff \Sigma^*_0 - \Sigma^*_1 + \Sigma^*_2 = 0 \\
&2. \quad \int_{-1}^{1} d\tilde{z} \, \Sigma^*(\tilde{z}) = \int_{-1}^{1} d\tilde{z} \, \Sigma(\tilde{z}) \iff \Sigma^*_0 + \frac{\Sigma^*_2}{3} = \Sigma_0 + \frac{\Sigma_2}{3} \\
&3. \quad \int_{-1}^{1} d\tilde{z} \, \tilde{z} \, \Sigma^*(\tilde{z}) = \int_{-1}^{1} d\tilde{z} \, \tilde{z} \, \Sigma(\tilde{z}) \iff \frac{\Sigma^*_1}{3} = \frac{\Sigma_1}{3}.
\end{aligned} \tag{5.39}$$

The fix-up applies to the cross sections coefficients of every reaction type, but, for the differential scattering cross section $\{\Sigma_{S,l}\}$ (Eq. (2.16)), only the isotropic angular component is involved.

The case of both extremes being negative has never showed up in the performed calculations. Even though this could be dealt with by replacing the condition for the highest moment with another zero condition, it is highly probable that the adopted mesh is not adequately refined to catch the axial gradients.

Chapter 6

The DP_N acceleration in a polynomial cross sections framework

The implementation of the new method could not be considered complete without the development or adaptation of an adequate acceleration technique. The convergence of the free power iteration scheme is in general quite slow, in the sense that it requires a great number of iterations: acceleration is thus necessary to reduce their number and, ultimately, to make realistic simulations more affordable.

In the MOC context, the Coarse-Mesh Finite-Difference (CMFD) method is by far the most widespread acceleration technique, probably due to the simplicity of its implementation; references [38] and [41] are just two examples of codes relying on this approach. TDT, on the other hand, has adopted the DP_N synthetic acceleration since its introduction by R. Sanchez and A. Chetaine in its simple DP_0 formulation [58]. The reasons behind this choice are many [5, 12]:

- the DP_N method does not require an ad-hoc homogenization mesh and can therefore be applied to arbitrary spatial grids;
- it has proven to be totally stable, contrarily to CMFD which is notoriously hard to stabilize;
- it allows the acceleration of higher-order angular and spatial components.

The main drawbacks of this approach are related to the difficulty of its implementation and, above all, to the heavy memory requirements; as will be shown in section 7.2.3, the storage required by the DP_N method can actually constitute the major contribution to the total memory.

The DP_N acceleration is named after the approximation it introduces for the angular flux over region boundaries, which is described in terms of spatially constant angular moments over the two sides of each common interface between neighboring regions, according to a *Double* P_N expansion. In the polynomial case, this translates into the constancy of the polynomial coefficients of every angular moment. Furthermore, a lower anisotropy order is used, so as to deal with fewer components and therefore with smaller systems while iterating. The acceleration is achieved by *correcting* the latest transport solution: to do this, one has to reformulate the original problem in its equivalent version for the error (with respect to

the converged solution), replace in the latter the transport operator with a lower-order one that accounts for the approximations introduced and, finally, add the obtained error to the latest transport iterate.

As the procedure above can be applied both at the inner- and outer-iteration level, this chapter firstly describes the two corresponding error formulations of the transport problem. As will be shown, the chosen strategy always hinges on the resolution of a mono-group lower-order problem, for which a suitable definition of the transmission and balance equations is required: hence, the $\mathrm{DP_N}$ equations adapted to the polynomial cross sections case are derived, starting from their transport counterparts. Similarly to what done for the transport terms in paragraph 5.2, some details concerning the vectorized evaluation of the acceleration coefficients are presented. Lastly, the chapter ends with the simple adaptation of the non-linear least-square fitting technique, introduced in [11] to limit the memory demand of $\mathrm{DP_N}$ coefficients.

6.1 Acceleration problem: error formulations

Within each transport outer iteration the acceleration is carried out by accelerating the inner iterations: a transport inner is thus followed by a set of $\mathrm{DP_N}$ iterations to compute the flux correction. The latter cycle considers the problem for error ε, obtained subtracting Eq. (3.27) from the same equation involving the converged solution (denoted by "∞"):

$$\begin{array}{ll} & \mathcal{L}\,\psi^{\infty} = \mathcal{H}_{g\to g}\,\psi^{\infty} \quad + S \\ (-) \quad & \mathcal{L}\,\psi^{it} \;= \mathcal{H}_{g\to g}\,\psi^{it-1} + S \\ \hline (=) \quad & \mathcal{L}\,\varepsilon \;= \mathcal{H}_{g\to g}\,\varepsilon + S^{\,it} \end{array} \quad , \tag{6.1}$$

where $\varepsilon = \psi^{\infty} - \psi^{it}$ and $S^{\,it} = \mathcal{H}_{g\to g}(\psi^{it} - \psi^{it-1})$. The error problem is then solved adopting the $\mathcal{L}_{\mathrm{DP_N}}$ lower-order operator instead of $\mathcal{L}$ (and using fewer flux components, see paragraph 6.2); as a consequence, the error solution does not correspond anymore to the difference between the converged and the current transport iterates, but rather to an approximated correction. The accelerated transport solution, at the end of the $\mathrm{DP_N}$ inner cycle, will read

$$\psi^{it}_{acc} = \psi^{it} + \tilde{\varepsilon}, \tag{6.2}$$

where $\tilde{\varepsilon}$ is solution of problem

$$\mathcal{L}_{\mathrm{DP_N}}\,\tilde{\varepsilon} = \mathcal{H}_{g\to g}\,\tilde{\varepsilon} + S^{it}_{acc}, \tag{6.3}$$

the source term being now given by $S^{it}_{acc} = \mathcal{H}_{g\to g}\Big(\psi^{it} - \psi^{it-1}_{acc}\Big)$. In practice, however, a thermal iterative step is made, for each energy group, by a single transport iteration followed by a $\mathrm{DP_N}$ corrective cycle.

Based on the work of I.R. Suslov [59], an acceleration scheme for the outer iterations was also implemented in TDT [60], consisting in solving a whole multi-group error problem by using the $\mathrm{DP_N}$ operator at the mono-group level. In a similar way to Eq. (6.1), separating the in-group and down-scattering from the up-scattering contribution in problem (3.25) as

$$\Big(\mathcal{L} - \mathcal{H}_{g'\le g}\Big)\,\psi^{th,o} = \mathcal{H}_{g'>g}\,\psi^{th-1,o} + \frac{1}{k^{o-1}_{\mathrm{eff}}}\mathcal{F}\phi^{o-1}, \tag{6.4}$$

with $\mathcal{H}_{g'\leq g}* = \mathcal{H}_{g'<g}* + \mathcal{H}_{g\to g}*$, a lower-order error relation can be formulated for the transport solution relative to outer iteration o and thermal (unconverged) iteration th, reading

$$\Big(\mathcal{L}_{\mathrm{DP_N}} - \mathcal{H}\Big)\,\tilde{\varepsilon} = \frac{1}{k_{\mathrm{eff}}}\,\mathcal{F}\,\tilde{\varepsilon} + S_{acc}(k_{\mathrm{eff}}), \tag{6.5}$$

where the source term, depending on the current eigenvalue estimate, is expressed as

$$S_{acc}(k_{\mathrm{eff}}) = \mathcal{H}_{g'>g}\Big(\psi^{th,o} - \psi^{th-1,o}\Big) + \mathcal{F}\left(\frac{\phi^{th,o}}{k_{\mathrm{eff}}} - \frac{\phi^{o-1}_{acc}}{k^{o-1}_{\mathrm{eff}}}\right). \tag{6.6}$$

Problem (6.5) is solvable through power iterations, updating at each outer step both the fission term and the inhomogeneous source S_{acc}. The new k_{eff} is calculated by a relation similar to Eq. (3.26), but involving both the latest transport solution and the error iterates:

$$k^{a}_{\mathrm{eff}} = \frac{\langle\, \mathcal{F}(\phi^{th,o} + \tilde{\varepsilon}^{a})\,\rangle}{\frac{1}{k^{a-1}_{\mathrm{eff}}}\,\langle\, \mathcal{F}(\phi^{th,o} + \tilde{\varepsilon}^{a-1})\,\rangle}, \tag{6.7}$$

where a denotes the $\mathrm{DP_N}$ outer iteration index. The resulting accelerated flux iterate reads

$$\psi^{o}_{acc} = \psi^{th,o} + \tilde{\varepsilon}. \tag{6.8}$$

6.2 Derivation of the $\mathrm{DP_N}$ transmission and balance equations

According to the previous paragraph, a mono-group problem has to be solved, in the $\mathrm{DP_N}$ framework of TDT, to accelerate both the transport inner and outer iterations. The error solution is provided by Eq. (6.3) for the former and by an analogous equation but involving a different source term for the latter kind of iterations. While accelerating the outer cycle, the mono-group source is actually given by the sum of source (6.6) and a term equivalent to Eq. (3.29), with the error instead of the flux. Hence, adapted versions of transmission and balance equations, coherent with the $\mathrm{DP_N}$ hypotheses and constituting the $\mathrm{DP_N}$ inner-iteration problem, are derived in the following. The procedure is quite similar to the one followed in [11] for the POL method. As in the transport case, however, no additional angular moment other than the required ones will appear in the obtained balance relation.

The starting point is the introduction of the already mentioned *surface* approximation for the boundary fluxes and of a *volume* approximation for the emission densities, respectively referred to by subscripts S and V. The numbers of angular moments, smaller than N_m, are then labeled $N_{m,S}$ and $N_{m,V}$ for the corresponding terms. In the considered applications, both numbers are given by a reduced anisotropy order equal to 1 (so that $N_{m,S} = N_{m,V} = 4$), whereas the $(N_p)^{\mathrm{th}}$-order polynomial expansion is kept for the neutron flux and cross sections. It follows that the $\mathrm{DP_N}$ correction will only concern a subset of the angular components of the transport solution.

Defining vectors

$$\begin{aligned}
\vec{\boldsymbol{\mathcal{Z}}}^{\pm}_{S}(\vec{r}_s,\vec{\Omega}) &= \Big\{A_n(\vec{\Omega})\,\vec{P}^{\pm}(\vec{r}_s),\ n = 1, N_{m,S}\Big\}\\
\vec{\boldsymbol{\mathcal{Z}}}^{2N_p}_{V}(\vec{r},\vec{\Omega}) &= \Big\{A_n(\vec{\Omega})\,\vec{P}^{\,2N_p}(\vec{r}),\ n = 1, N_{m,V}\Big\},
\end{aligned} \tag{6.9}$$

the emission densities can be written as

$$q(\vec{r},\vec{\Omega}) \approx \vec{\boldsymbol{\mathcal{Z}}}_V^{2N_p}(\vec{r},\vec{\Omega}) \cdot \vec{\boldsymbol{q}}_{V,r} \tag{6.10}$$

and the boundary fluxes for the generic α interface as

$$\psi(\vec{r}_s,\vec{\Omega}) \approx \vec{\boldsymbol{\mathcal{Z}}}_S^{\pm}(\vec{r}_s,\vec{\Omega}) \cdot \vec{\boldsymbol{\Phi}}_{S,r}^{\alpha^{\pm}}, \qquad \vec{r}_s \in \alpha,\ \vec{\Omega} \in 2\pi^{\pm}, \tag{6.11}$$

where "$\pm$" indicates which side of α is considered: given an $\vec{\Omega}$ direction, the convention is adopted that "+" refers to the region left by $\vec{\Omega}$, whereas "$-$" to the region $\vec{\Omega}$ is entering; moreover, for any α interface of region r, $2\pi^{\pm}$ is the set of outgoing ("+") or incoming ("$-$") directions with respect to r. In the following, if two vectors $\vec{\boldsymbol{\mathcal{Z}}}^{+}$ and $\vec{\boldsymbol{\mathcal{Z}}}^{-}$ (or $\vec{P}^{+}$ and $\vec{P}^{-}$) refer to the same region, the $\pm$ notation will be omitted.

Considering the $\mathrm{DP_N}$ balance firstly, a projection of Eq. (4.1) on the $\vec{\boldsymbol{\mathcal{Z}}}_V$ basis leads to write, similarly to the transport case,

$$\boldsymbol{\Gamma}_{V,r}\,\vec{\boldsymbol{\Phi}}_{V,r} = \boldsymbol{\mathcal{Z}}_{V,r}\,\vec{\boldsymbol{q}}_{V,r} - \oint \frac{d\vec{\Omega}}{4\pi}\,\vec{A}_V(\vec{\Omega})\,\frac{1}{V_r}\int_{\partial r} d\vec{r}_s\ \vec{\Omega}\cdot\hat{n}\,\vec{P}(\vec{r})\,\psi(\vec{r}_s,\vec{\Omega}), \tag{6.12}$$

matrix $\boldsymbol{\mathcal{Z}}_{V,r}$ being defined as $\boldsymbol{\mathcal{Z}}_{V,r} = \oint \frac{d\vec{\Omega}}{4\pi}\int_r \frac{d\vec{r}}{\tilde{V}_r(\vec{\Omega})}\,\vec{\boldsymbol{\mathcal{Z}}}_V \otimes \vec{\boldsymbol{\mathcal{Z}}}_V^{2N_p}$. Due to the smaller number of moments, no subsystem is defined for acceleration and only one system of $N_{m,V}(N_p+1)$ equations is solved, with $\boldsymbol{\Gamma}_{V,r}$ suitably constructed from the elements of matrices $\{\boldsymbol{\Gamma}_r^m\}$. A further approximation is the use of the average region volume V_r in place of its direction-dependent numerical value in the streaming term. The latter is rearranged, expressing ∂r as the sum of region interfaces and distinguishing between incoming and outgoing directions, as

$$\begin{aligned}
&\frac{1}{V_r}\oint \frac{d\vec{\Omega}}{4\pi}\,\vec{A}_V(\vec{\Omega})\int_{\partial r} d\vec{r}_s\,\vec{\Omega}\cdot\hat{n}\,\vec{P}(\vec{r}_s)\,\psi(\vec{r}_s,\vec{\Omega}) \\
&\qquad = \frac{1}{V_r}\sum_{\alpha\in\partial r}\int_\alpha d\vec{r}_s\Bigg[\int_{2\pi^+}\frac{d\vec{\Omega}}{4\pi}\,|\vec{\Omega}\cdot\hat{n}|\,\vec{\boldsymbol{\mathcal{Z}}}_V(\vec{r}_s,\vec{\Omega})\,\psi(\vec{r}_s,\vec{\Omega}) \\
&\qquad\qquad - \int_{2\pi^-}\frac{d\vec{\Omega}}{4\pi}\,|\vec{\Omega}\cdot\hat{n}|\,\vec{\boldsymbol{\mathcal{Z}}}_V(\vec{r}_s,\vec{\Omega})\,\psi(\vec{r}_s,\vec{\Omega})\Bigg].
\end{aligned} \tag{6.13}$$

Then, introducing the partial currents defined as

$$\vec{\boldsymbol{J}}_\alpha^{\pm} = \int_\alpha d\vec{r}_s\int_{2\pi^\pm}\frac{d\vec{\Omega}}{4\pi}\,|\vec{\Omega}\cdot\hat{n}|\,\vec{\boldsymbol{\mathcal{Z}}}_V^{\pm}(\vec{r}_s,\vec{\Omega})\,\psi(\vec{r}_s,\vec{\Omega}), \tag{6.14}$$

a preliminary version of the balance equation is written in the form

$$\boldsymbol{\Gamma}_{V,r}\,\vec{\boldsymbol{\Phi}}_{V,r} = \boldsymbol{\mathcal{Z}}_{V,r}\,\vec{\boldsymbol{q}}_{V,r} - \frac{1}{V_r}\sum_{\alpha\in\partial r}\left(\vec{\boldsymbol{J}}_\alpha^{+} - \vec{\boldsymbol{J}}_\alpha^{-}\right). \tag{6.15}$$

The $\mathrm{DP_N}$ transmission equation is therefore needed to evaluate the currents term; its derivation starts from Eq. (5.6) limited to the reduced number of angular moments ($N_{m,V}$):

$$\psi(\vec{r}_s^{\,+},\vec{\Omega}) = \psi(\vec{r}_s^{\,-},\vec{\Omega})\,e^{-\tau} + \sum_{n=1}^{N_{m,V}} A_n(\vec{\Omega})\sum_{p=0}^{2N_p}\left[\sum_{k=0}^{p} P_k(\vec{r}^{\,-})\binom{p}{k}\left(\frac{2\mu}{\Delta z_r}\right)^{p-k} E_{p-k}\right] q_{n,r,p}, \tag{6.16}$$

where $\vec{r}_s^{\pm}$ refer to the chord endpoints, with the usual notation. The previous relation is then substituted into the currents definition (Eq. (6.14)) to obtain a flux-current hybrid formulation of the transmission equation:

$$\vec{\boldsymbol{J}}_{\alpha}^{+} = \sum_{\beta \in \partial r} \boldsymbol{\mathcal{T}}_{\alpha\beta}\, \vec{\boldsymbol{\Phi}}_{S,r}^{\beta^-} + \boldsymbol{\mathcal{E}}_{\alpha}\, \vec{\boldsymbol{q}}_{V,r}. \tag{6.17}$$

Besides the use of Eq. (6.11) for the expression of the flux entering through the β interface, the *transmission* and *escape* coefficients $\boldsymbol{\mathcal{T}}$'s and $\boldsymbol{\mathcal{E}}$'s have been introduced, so that the exiting current $\vec{\boldsymbol{J}}_{\alpha}^{+}$ is written in terms of the contributions due to the incomings from all the β's composing ∂r and to the emission within r. Their definitions read

$$\begin{aligned} \boldsymbol{\mathcal{T}}_{\alpha\beta} &= \int_{\alpha} d\vec{r}_s^{+} \int_{(\beta\to\alpha)} \frac{d\vec{\Omega}}{4\pi} \left|\vec{\Omega}\cdot\hat{n}\right| \vec{\boldsymbol{\mathcal{Z}}}_S(\vec{r}_s^{+},\vec{\Omega}) \otimes \vec{\boldsymbol{\mathcal{Z}}}_S(\vec{r}_s^{-},\vec{\Omega})\, e^{-\tau} \\ \boldsymbol{\mathcal{E}}_{\alpha} &= \int_{\alpha} d\vec{r}_s^{+} \int_{2\pi^+} \frac{d\vec{\Omega}}{4\pi} \left|\vec{\Omega}\cdot\hat{n}\right| \vec{\boldsymbol{\mathcal{Z}}}_S(\vec{r}_s^{+},\vec{\Omega}) \otimes \vec{\boldsymbol{W}}_V^{2N_p}(\vec{r}_s^{-},\vec{\Omega}), \end{aligned} \tag{6.18}$$

where $\vec{\boldsymbol{W}}_V^{2N_p}$ is a vector of $N_{m,V}(2N_p+1)$ components, with

$$\left[\vec{\boldsymbol{W}}_V^{2N_p}(\vec{r}_s,\vec{\Omega})\right]_{n,p} = A_n(\vec{\Omega}) \sum_{k=0}^{p} P_k(\vec{r}_s) \binom{p}{k} \left(\frac{2\mu}{\Delta z_r}\right)^{p-k} E_{p-k}, \tag{6.19}$$

and the following reformulation of the integral over outgoing directions crossing α is adopted for $\boldsymbol{\mathcal{T}}$'s, treating separately the $\vec{\Omega}$'s coming from different β interfaces:

$$\int_{\alpha} d\vec{r}_s \int_{2\pi^+} \frac{d\vec{\Omega}}{4\pi} * = \int_{\alpha} d\vec{r}_s \sum_{\beta\in\partial r} \int_{(\beta\to\alpha)} \frac{d\vec{\Omega}}{4\pi} * \,. \tag{6.20}$$

On the other hand, $\boldsymbol{\mathcal{E}}$'s exploit the subdivision of ∂r only during the construction phase, since $\boldsymbol{\mathcal{E}}_{\alpha}$ can be expressed as $\sum_{\beta\in\partial r} \boldsymbol{\mathcal{E}}_{\alpha\beta}$. It is worth noting that system (6.17) is limited to the dimensions of the surface approximation: if $N_{m,V} > N_{m,S}$, the current terms corresponding to the higher moments are set to 0. More details regarding the evaluation of coefficients (6.18) and its optimization through loop vectorization will be provided in the next paragraph.

Exactly as in the derivation of the POL method, the next step consists in substituting Eq. (6.17) into Eq. (6.15), in order to obtain a balance relation linking the flux to the emission densities and the entering currents. Before doing that, the former equation needs be modified, to involve these last quantities in place of the entering fluxes $\left\{\vec{\boldsymbol{\Phi}}^{\beta^-}\right\}$. To this purpose, Eqs. (6.11) and (6.14) (extended/limited to $N_{m,S}$ moments) can be combined, yielding

$$\vec{\boldsymbol{\Phi}}_r^{\alpha^{\pm}} = \left[\int_{\alpha} d\vec{r}_s \int_{2\pi^{\pm}} \frac{d\vec{\Omega}}{4\pi} \left|\vec{\Omega}\cdot\hat{n}\right| \vec{\boldsymbol{\mathcal{Z}}}_S^{\pm}(\vec{r}_s,\vec{\Omega}) \otimes \vec{\boldsymbol{\mathcal{Z}}}_S^{\pm}(\vec{r}_s,\vec{\Omega})\right]^{-1} \vec{\boldsymbol{J}}_{\alpha}^{\pm} = \boldsymbol{\aleph}_{\alpha^{\pm}}^{-1}\, \vec{\boldsymbol{J}}_{\alpha}^{\pm}, \tag{6.21}$$

where matrix $\boldsymbol{\aleph}_{\alpha}$ has been implicitely defined. Then, denoting by $\tilde{\boldsymbol{\mathcal{T}}}_{\alpha\beta}$ the result of the application of matrix $\boldsymbol{\mathcal{T}}_{\alpha\beta}$ to $\boldsymbol{\aleph}_{\beta^-}^{-1}$, Eq. (6.17) becomes

$$\vec{\boldsymbol{J}}_{\alpha}^{+} = \sum_{\beta\in\partial r} \tilde{\boldsymbol{\mathcal{T}}}_{\alpha\beta}\, \vec{\boldsymbol{J}}_{\beta}^{-} + \boldsymbol{\mathcal{E}}_{\alpha}\, \vec{\boldsymbol{q}}_{V,r}, \tag{6.22}$$

which can be substituted into the preliminary balance equation (6.15). Inverting the sum over exiting and entering surfaces and considering matrices $\tilde{\boldsymbol{\mathcal{T}}}_\beta = \sum_{\alpha\in\partial r} \tilde{\boldsymbol{\mathcal{T}}}_{\alpha\beta}$ and $\boldsymbol{\mathcal{E}}_r = \frac{1}{V_r}\sum_{\alpha\in\partial r}\boldsymbol{\mathcal{E}}_\alpha$, the new balance relation reads

$$\boldsymbol{\Gamma}_{V,r}\,\vec{\boldsymbol{\Phi}}_{V,r} = \left(\boldsymbol{\mathcal{Z}}_{V,r} - \boldsymbol{\mathcal{E}}_r\right)\vec{\boldsymbol{q}}_{V,r} + \frac{1}{V_r}\sum_{\beta\in\partial r}\left(\boldsymbol{I} - \tilde{\boldsymbol{\mathcal{T}}}_\beta\right)\vec{\boldsymbol{J}}^-_\beta\,, \tag{6.23}$$

$\boldsymbol{I}$ being the identity matrix. At this point it is convenient to separate in the emission density the contributions due to in-group scattering from the "external" emission, that is, fission and scattering from other energy groups:

$$\vec{\boldsymbol{q}}_{V,r} = \vec{\boldsymbol{q}}^{\,\mathrm{in}}_{V,r} + \vec{\boldsymbol{q}}^{\,\mathrm{ext}}_{V,r} = \boldsymbol{\Sigma}_{S,r}\vec{\boldsymbol{\Phi}}_{V,r} + \vec{\boldsymbol{q}}^{\,\mathrm{ext}}_{V,r}\,, \tag{6.24}$$

where the rectangular matrix $\boldsymbol{\Sigma}_{S,r}$ contains all the polynomial coefficients of the angular moments with respect to Legendre polynomials of the in-group scattering cross section, according to

$$\boldsymbol{\Sigma}_{S,r} = \begin{pmatrix} \bar{\bar{\Sigma}}^1_{S,r} & 0 & \dots & 0 \\ 0 & \bar{\bar{\Sigma}}^2_{S,r} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \bar{\bar{\Sigma}}^{N_{m,V}}_{S,r} \end{pmatrix} \qquad \bar{\bar{\Sigma}}^n_{S,r} = \begin{pmatrix} \Sigma^{g\to g}_{S,r,l(n),0} & 0 & \dots & 0 \\ \Sigma^{g\to g}_{S,r,l(n),1} & \Sigma^{g\to g}_{S,r,l(n),0} & \ddots & \vdots \\ \vdots & \Sigma^{g\to g}_{S,r,l(n),1} & \ddots & 0 \\ \Sigma^{g\to g}_{S,r,l(n),N_p} & \vdots & \ddots & \Sigma^{g\to g}_{S,r,l(n),0} \\ 0 & \Sigma^{g\to g}_{S,r,l(n),N_p} & \vdots & \Sigma^{g\to g}_{S,r,l(n),1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \Sigma^{g\to g}_{S,r,l(n),N_p} \end{pmatrix}. \tag{6.25}$$

Defining the following matrices:

$$\begin{aligned} \boldsymbol{\mathcal{C}}_r &= \boldsymbol{\mathcal{Z}}_{V,r} - \boldsymbol{\mathcal{E}}_r \\ \tilde{\boldsymbol{\Gamma}}_r &= \boldsymbol{\Gamma}_{V,r} - \boldsymbol{\mathcal{C}}_r\,\boldsymbol{\Sigma}_{S,r} \\ \tilde{\boldsymbol{\mathcal{C}}}_r &= \tilde{\boldsymbol{\Gamma}}_r^{-1}\boldsymbol{\mathcal{C}}_r \\ \boldsymbol{\mathcal{I}}_\beta &= \frac{1}{V_r}\,\tilde{\boldsymbol{\Gamma}}_r^{-1}\left(\boldsymbol{I} - \tilde{\boldsymbol{\mathcal{T}}}_\beta\right), \end{aligned} \tag{6.26}$$

Eq. (6.23) can finally be written as

$$\vec{\boldsymbol{\Phi}}_{V,r} = \tilde{\boldsymbol{\mathcal{C}}}_r\,\vec{\boldsymbol{q}}^{\,\mathrm{ext}}_{V,r} + \sum_{\beta\in\partial r}\boldsymbol{\mathcal{I}}_\beta\,\vec{\boldsymbol{J}}^-_\beta\,. \tag{6.27}$$

As for the transmission equation, Eq. (6.24) and the just obtained $\mathrm{DP_N}$ balance equation can be substituted into the last term of Eq. (6.22), yielding

$$\vec{\boldsymbol{J}}_{\alpha}^{+} = \sum_{\beta \in \partial r} \left(\tilde{\boldsymbol{\mathcal{T}}}_{\alpha\beta} + \boldsymbol{\mathcal{E}}_{\alpha}\, \boldsymbol{\Sigma}_{S,r}\, \boldsymbol{\mathcal{I}}_{\beta} \right) \vec{\boldsymbol{J}}_{\beta}^{-} + \boldsymbol{\mathcal{E}}_{\alpha} \left(\boldsymbol{\Sigma}_{S,r}\, \tilde{\boldsymbol{\mathcal{C}}}_{r} + \boldsymbol{I} \right) \vec{\boldsymbol{q}}_{V,r}^{\,\mathrm{ext}}. \tag{6.28}$$

A further step is taken considering a single system for all the currents present in the domain, which is obtained assembling the vector $\vec{\boldsymbol{J}}^{+}$ with the $\left\{ \vec{\boldsymbol{J}}_{\alpha}^{+} \right\}$ of all the N_{curr} interfaces:

$$\vec{\boldsymbol{J}}^{+} = \left\{ \vec{\boldsymbol{J}}_{\alpha}^{+},\ \alpha = 1, N_{curr} \right\}, \tag{6.29}$$

and doing the same for the other vectors and matrices. In this way, the final transmission system can be written in the form

$$\vec{\boldsymbol{J}}^{+} = \boldsymbol{\mathcal{T}}^{*}\, \vec{\boldsymbol{J}}^{-} + \boldsymbol{\mathcal{E}}^{*}\, \vec{\boldsymbol{q}}_{V,r}^{\,\mathrm{ext}}. \tag{6.30}$$

This equation constitutes a closed system in itself: for a given external source $\vec{\boldsymbol{q}}_{V,r}^{\,\mathrm{ext}}$, the transmission system is applied to retrieve the partial currents of all regions, which are needed to get the correction fluxes through the balance relation (6.27). Notably, the transmission relations derived for acceleration significantly differ from their transport counterpart: while the output of the latter is a first-flight (uncollided) quantity, the former takes into account possible collisions of the inputs, according to a multi-collisional approach. This formulation permits limiting the in-group iterative problem to the transmission system only and to resort to the balance equation only once convergence has been achieved. The solution of this problem is found using the preconditioned Bi-Conjugate Gradient Stabilized Method (BiCGSTAB) [19], like for the step-constant case [5]: its application to the presented method differs only for the matrices dimensions.

6.3 Vectorization of $\mathrm{DP_N}$ coefficients

Contrarily to the transport coefficients, the acceleration matrices appearing in Eqs. (6.27) and (6.30) are computed and stored once and for all, for every energy group, before the actual calculation. Once again, the classification of chords can be exploited: the surface integrals appearing in Eq. (6.18) are actually approximated cumulating the contributions due to all the chords comprised between the interfaces α and β (recalling that $\boldsymbol{\mathcal{E}}_{\alpha} = \sum_{\beta \in \partial r} \boldsymbol{\mathcal{E}}_{\alpha\beta}$), so that the evaluation strategy of the transmission and escape terms can be differentiated based on the surfaces orientation (that is, on the chords classes). For the step-constant cross sections case this procedure is again especially advantageous for V-chords: as a matter of fact, the components of $\boldsymbol{\mathcal{T}}$ and $\boldsymbol{\mathcal{E}}$ for two vertical surfaces α_v and β_v read [11]

$$\begin{aligned}
\left[\boldsymbol{\mathcal{T}}_{\alpha_v\beta_v} \right]_{n,n'}^{p,p'} &= \int_{s_{\alpha_v}} ds \int_{(\beta_v \to \alpha_v)} \frac{d\vec{\Omega}}{4\pi} \left| \vec{\Omega} \cdot \hat{n} \right| A_n(\vec{\Omega})\, A_{n'}(\vec{\Omega}) \int_{z_{\alpha_v}} dz\, P_p(\vec{r}_s^{\,+})\, P_{p'}(\vec{r}_s^{\,-})\, e^{-\tau} \\
\left[\boldsymbol{\mathcal{E}}_{\alpha_v\beta_v} \right]_{n,n'}^{p,p'} &= \int_{s_{\alpha_v}} ds \int_{(\beta_v \to \alpha_v)} \frac{d\vec{\Omega}}{4\pi} \left| \vec{\Omega} \cdot \hat{n} \right| A_n(\vec{\Omega})\, A_{n'}(\vec{\Omega}) \cdot \\
&\qquad \cdot \sum_{k=0}^{p'} \binom{p'}{k} \left(\frac{2\mu}{\Delta z} \right)^{p'-k} \int_{z_{\alpha_v}} dz\, P_p(\vec{r}_s^{\,+})\, P_k(\vec{r}_s^{\,-})\, E_{p'-k},
\end{aligned} \tag{6.31}$$

where the surface integral has been decomposed into its radial and axial contributions; if cross sections are constant within each region, the $e^{-\tau}$ and $E_{p'-k}$ terms do not depend on z and can therefore be taken out of the integrals. Moreover, each integral $\int_{z_{\alpha_v}} dz\, P_p(\vec{r}_s^{\,+})\, P_k(\vec{r}_s^{\,-})$ needs be computed only once, since its value does not depend on energy: this is the approach used in [11] and, for the presented method, in every constant cross sections region.

If, instead, cross sections have polynomial profiles, these calculations can still benefit from vectorization: for each 2D chord i, having its endpoints on β_v and α_v, and direction $\vec{\Omega}_i$ of radial projection parallel to i, a set of V-chords $\{v\}$ is identified for which the z-dependent terms $\vec{P}(\vec{r}_s^{\,\pm})$, $e^{-\tau}$ and $\left\{E_{p'-k}\right\}$ are evaluated in a vectorized fashion, for all energy groups. The double integral over α_v and over the directions coming from β_v is therefore approximated as

$$\begin{aligned}\int_{\alpha_v} d\vec{r}_s \int_{(\beta_v\to\alpha_v)} \frac{d\vec{\Omega}}{4\pi} \left|\vec{\Omega}\cdot\hat{n}\right| * &= \int_{s_{\alpha_v}} ds \int_{(\beta_v\to\alpha_v)} \frac{d\vec{\Omega}}{4\pi} \left|\vec{\Omega}\cdot\hat{n}\right| \int_{z_{\alpha_v}} dz * \\ &\approx \sum_i \int_{\{\vec{\Omega}_i\}} \frac{d\vec{\Omega}}{4\pi}\, \Delta_{\perp\vec{\Omega}} \sum_{\substack{v\parallel\vec{\Omega}\\ v\subset r}} * \ ,\end{aligned} \tag{6.32}$$

where $\{\vec{\Omega}_i\}$ is the set of all directions whose radial projection is parallel to i and $\sum_{\substack{v\parallel\vec{\Omega}\\ v\subset r}}$ adopts the same notation introduced in Eq. (4.7); hence, the components of Eq. (6.31) can be expressed in the following way:

$$\begin{aligned}\left[\boldsymbol{\mathcal{T}}_{\alpha_v\beta_v}\right]^{p,p'}_{n,n'} &= \sum_i \int_{\{\vec{\Omega}_i\}} \frac{d\vec{\Omega}}{4\pi}\, A_n(\vec{\Omega})\, A_{n'}(\vec{\Omega})\, \Delta_{\perp\vec{\Omega}}\, \mathbb{I}_T^{p,p'} \\ \left[\boldsymbol{\mathcal{E}}_{\alpha_v\beta_v}\right]^{p,p'}_{n,n'} &= \sum_i \int_{\{\vec{\Omega}_i\}} \frac{d\vec{\Omega}}{4\pi}\, A_n(\vec{\Omega})\, A_{n'}(\vec{\Omega})\, \Delta_{\perp\vec{\Omega}}\, \mathbb{I}_E^{p,p'},\end{aligned} \tag{6.33}$$

$\mathbb{I}_T^{p,p'}$ and $\mathbb{I}_E^{p,p'}$ being defined as

$$\mathbb{I}_T^{p,p'} = \sum_{\substack{v\parallel\vec{\Omega}\\ v\subset r}} P_p(\vec{r}^{\,+}_{s,v})\, P_{p'}(\vec{r}^{\,-}_{s,v})\, e^{-\tau_v} \tag{6.34a}$$

$$\mathbb{I}_E^{p,p'} = \sum_{k=0}^{p'} \binom{p'}{k} \left(\frac{2\mu}{\Delta z}\right)^{p'-k} \sum_{\substack{v\parallel\vec{\Omega}\\ v\subset r}} P_p(\vec{r}^{\,+}_{s,v})\, P_k(\vec{r}^{\,-}_{s,v})\, E_{p'-k,v}. \tag{6.34b}$$

The OpenMP SIMD vectorization is exploited for the evaluation of the two terms above. As shown in Algorithm 4, the energy dependence of $e^{-\tau}$ and $\left\{E_{p'-k}\right\}$ is dealt with by applying the SIMD pragma on two subsequent loops: the former computes the products of the P polynomials for all the V-chords of the set, while the latter, of *reduction* type, cumulates the chords contributions for each energy groups. For $\mathbb{I}_E^{p,p'}$, the number of flops is further reduced by extending the first loop to the $\binom{p'}{k}\left(\frac{2\mu}{\Delta z}\right)^{p'-k}$ product, to not have to compute it once per group after the reduction.

Algorithm 4 Vectorized computation of acceleration coefficients

! $\mathbb{I}_T^{p,p'} = \sum_v P_p(\vec{r}_{s,v}^{+})\, P_{p'}(\vec{r}_{s,v}^{-})\, e^{-\tau_v}$ (Eq. (6.34a)), for all energy groups

!$OMP SIMD
for $v = 1, N_{\text{V-chords}}$ **do**
    $aux(v) = P_p(\vec{r}_{s,v}^{+})\, P_{p'}(\vec{r}_{s,v}^{-})$
end for
for $g = 1, N_g$ **do**
    $sum = 0.$
    !$OMP SIMD REDUCTION $(+ : sum)$
    for $v = 1, N_{\text{V-chords}}$ **do**
        $sum = sum + aux(v) * \exp[-\tau_v(g)]$
    end for
    $\mathbb{I}_T^{p,p'}(g) = sum$
end for

! $\mathbb{I}_E^{p,p'} = \sum_{k=0}^{p'} \binom{p'}{k} \left(\frac{2\mu}{\Delta z}\right)^{p'-k} \sum_v P_p(\vec{r}_{s,v}^{+})\, P_k(\vec{r}_{s,v}^{-})\, E_{p'-k,v}$ (Eq. (6.34b)),
! for all energy groups

$\mathbb{I}_E^{p,p'}(1 : N_g) = 0.$
for $k = 0, p'$ **do**
    !$OMP SIMD
    for $v = 1, N_{\text{V-chords}}$ **do**
        $aux(v) = P_p(\vec{r}_{s,v}^{+})\, P_k(\vec{r}_{s,v}^{-}) \binom{p'}{k} \left(\frac{2\mu}{\Delta z}\right)^{p'-k}$
    end for
    for $g = 1, N_g$ **do**
        $sum = 0.$
        !$OMP SIMD REDUCTION $(+ : sum)$
        for $v = 1, N_{\text{V-chords}}$ **do**
            $sum = sum + aux(v) * E_{p'-k,v}(g)$
        end for
        $\mathbb{I}_E^{p,p'}(g) = \mathbb{I}_E^{p,p'}(g) + sum$
    end for
end for

Legend

- $N_{\text{V-chords}}$: number of V-chords contained in the considered set;
- $P_i(\vec{r}_{s,v}^{\pm})$: i^{th}-degree polynomial evaluated at the exit (+, surface α_v) or entry (-, surface β_v) point of chord v;
- $\tau_v(g)$: optical path length of chord v for the g group;
- $E_{i,v}(g)$: i^{th} transmission integral for chord v and group g.

6.4 Adaptation of the DP_N matrices fitting technique

The DP_N method entails a quite relevant memory usage to store the acceleration coefficients that are used in iterations. In 3D calculations, and especially in higher-order frameworks, this may easily represent the major contribution to the total memory demand. For this reason, the development of the POL method included the introduction of a fitting strategy aimed at avoiding the storage of all the components of the DP_N matrices for every energy group: for each 3D region, matrix to fit and element of the matrix, the energy dependence is expressed by a fitting law whose parameters are obtained solving a least-square problem. Clearly, the downside of this approach is that additional flops are required to reconstruct the coefficients each time these are to be used. In order to minimize this cost, the adopted technique is applied to the multi-collisional forms of the matrices, that is, the ones that are found in Eqs. (6.27) and (6.30), after the preconditioning phase needed for the BiCGSTAB. Since the energy dependence of these terms is only due to the total cross section and to the in-group scattering cross section moments up to the first order of anisotropy (according to the DP_N hypotheses), the chosen form of the fitting functions, in the POL case, reads

$$f(\vec{\alpha}, \Sigma_T, \Sigma_{S,0}, \Sigma_{S,1}) = \alpha_1 \, e^{\alpha_2 \Sigma_T} (\Sigma_T)^{\alpha_3} + \alpha_4 \, e^{\alpha_5 \Sigma_{S,0}} (\Sigma_{S,0})^{\alpha_6} + \alpha_7 \, e^{\alpha_8 \Sigma_{S,1}} (\Sigma_{S,1})^{\alpha_9}, \tag{6.35}$$

where the region index is omitted and $\vec{\alpha}$ contains the nine parameters appearing in the equation. More details about the choice of the function and of the coefficients to fit can be found in [11]. If the value resulting from the previous formula is too far from the correct one, the corresponding element is not fitted and its position is recorded into a suitable array: the memory occupied by the unfitted terms is thus higher than what would be in case of direct storage and it is therefore convenient to limit as much as possible their number.

The method employed to determine the α parameters is described in [61]. This paragraph adopts, as in the reference, the notation $\vec{x}^{\,T}\vec{y} = \vec{y}^{\,T}\vec{x}$ for the scalar product of two vectors $\vec{x}$ and $\vec{y}$. Like any least-square approach, its scope is to minimize the E function defined as

$$E(\vec{\alpha}) = \sum_{g=1}^{N_g} \left[e_g(\vec{\alpha}) \right]^2 = \vec{e}^{\,T}\vec{e}, \tag{6.36}$$

where $\vec{e}(\vec{\alpha})$ is the vector function of the errors between the actual and the fitted values of the matrix element; for each energy group g, it is given by

$$e_g(\vec{\alpha}) = y_g - f(\vec{\alpha}, \Sigma_T^g, \Sigma_{S,0}^{g\to g}, \Sigma_{S,1}^{g\to g}), \tag{6.37}$$

y_g denoting the true value. According to the Gauss-Newton method, the idea is to find a correction vector $\delta\vec{\alpha}$ which, added to the initial $\vec{\alpha}$ parameters, reduces E, and to repeat the procedure iteratively. To do that, a first order Taylor expansion is introduced for $\vec{e}$, which allows writing

$$\vec{e}\,(\vec{\alpha} + \delta\vec{\alpha}) \approx \vec{e}\,(\alpha) + \bar{\bar{J}}\,\delta\vec{\alpha}, \qquad J_{ij} = \left.\frac{\partial e_i}{\partial \alpha_j}\right|_{\vec{\alpha}} \tag{6.38}$$

$\bar{\bar{J}}$ being the Jacobian matrix evaluated in $\vec{\alpha}$. Then, this relation is substituted into the right-hand side of Eq. (6.36) to obtain an approximation L of E that accounts for the linearization

of the vector function $\vec{e}$:

$$E(\vec{\alpha}+\delta\vec{\alpha}) \approx L(\delta\vec{\alpha}) = E(\vec{\alpha}) + 2\,\delta\vec{\alpha}^{\,T}\bar{\bar{J}}^{T}\vec{e}(\vec{\alpha}) + \delta\vec{\alpha}^{\,T}\bar{\bar{J}}^{T}\bar{\bar{J}}\,\delta\vec{\alpha}. \tag{6.39}$$

$\delta\vec{\alpha}$ is determined as the stationary point that minimizes L. It is therefore useful to introduce its gradient and Hessian:

$$L'(\delta\vec{\alpha}) = 2\left(\bar{\bar{J}}^{T}\vec{e}(\vec{\alpha}) + \bar{\bar{J}}^{T}\bar{\bar{J}}\,\delta\vec{\alpha}\right), \qquad L''(\delta\vec{\alpha}) = 2\,\bar{\bar{J}}^{T}\bar{\bar{J}}. \tag{6.40}$$

Provided that $\bar{\bar{J}}$ is non-singular, L'' will always be positive definite, which means that the stationary point of L will be its unique minimizer. This latter is found by setting L' equal to 0:

$$\bar{\bar{J}}^{T}\bar{\bar{J}}\,\delta\vec{\alpha} = -\bar{\bar{J}}^{T}\vec{e}(\vec{\alpha}), \tag{6.41}$$

which is the solving equation of a linear least-square problem. In fact, given an overdetermined linear system $\bar{\bar{A}}\,\vec{x} = \vec{b}$ ($\bar{\bar{A}}$ being an m x n matrix with $m > n$ and $rank(\bar{\bar{A}}) = n$), its *best* solution $\hat{x}$ is obtained solving the system

$$\bar{\bar{A}}^{T}\bar{\bar{A}}\,\vec{x} = \bar{\bar{A}}^{T}\vec{b} \iff \vec{x} = \left(\bar{\bar{A}}^{T}\bar{\bar{A}}\right)^{-1}\bar{\bar{A}}^{T}\vec{b} \tag{6.42}$$

or, equivalently, by replacing $\vec{b}$ in the original system with its projection $\bar{\bar{P}}\vec{b}$ on the column space of $\bar{\bar{A}}$, with $\bar{\bar{P}} = \bar{\bar{A}}\left(\bar{\bar{A}}^{T}\bar{\bar{A}}\right)^{-1}\bar{\bar{A}}^{T}$.

The $\delta\vec{\alpha}$ correction resulting from Eq. (6.41) is applied to $\vec{\alpha}$ to obtain the next iterates:

$$\vec{\alpha}_{k+1} = \vec{\alpha}_k + \delta\vec{\alpha}. \tag{6.43}$$

However, based on the goodness of approximation (6.38), $E(\vec{\alpha}+\delta\vec{\alpha})$ may be smaller of $E(\vec{\alpha})$, or not. To avoid the latter eventuality, the Levenberg-Marquardt variant of the Gauss-Newton approach, firstly proposed in [62] and taken again from [61], is followed. These authors suggested to consider the damped problem

$$\left(\bar{\bar{J}}^{T}\bar{\bar{J}} + \mu\bar{\bar{I}}\right)\delta\vec{\alpha} = -\bar{\bar{J}}^{T}\vec{e}(\vec{\alpha}) \tag{6.44}$$

in place of Eq. (6.41), where the damping parameter μ is gradually reduced as $E(\vec{\alpha})$ becomes smaller and smaller. The update of μ is ruled by the gain ration ρ, defined as

$$\rho = \frac{E(\vec{\alpha}) - E(\vec{\alpha}+\delta\vec{\alpha})}{L(0) - L(\delta\vec{\alpha})}. \tag{6.45}$$

If ρ is negative, the new iterate, certainly worse than the previous, is discarded and μ is increased. If, instead, ρ is positive, μ can be decreased to make problem (6.44) closer to (6.41).

In the attempt to extend the applicability of the fitting technique to the polynomial cross sections case, per-region average values are used in place of the constant terms in Eq. (6.35). For the generic R reaction, these are simply computed as

$$\bar{\Sigma}_R = \frac{\int_r d\vec{r}\,\Sigma_R(\vec{r})}{\int_r d\vec{r}} = \frac{\int_{\bar{z}_r-\Delta z_r/2}^{\bar{z}_r+\Delta z_r/2} dz\,\vec{P}(\vec{r})\cdot\vec{\Sigma}_R}{\int_{\bar{z}_r-\Delta z_r/2}^{\bar{z}_r+\Delta z_r/2} dz} = \sum_i \frac{\Sigma_{R,2i}}{2i+1}, \tag{6.46}$$

with the sum involving only the even-order polynomial coefficients. As will be shown in section 7.4, this approach proved successful in that it does not lead to an excessive number of unfitted elements, even at high burnup levels, and can therefore be adopted to fit the acceleration matrices in case of not spatially constant cross sections.

Part III

RESULTS AND CONCLUSIONS

Chapter 7

Results

The polynomial cross sections method proposed in this thesis represents a generalization of the POL approach, in that it extends its application to non-negligible burnup configurations. For such cases, the new method aims to be a preferable alternative to the traditional SC approximation: in order to show its advantage, this chapter provides comparisons between the classical and the innovative strategy applied to three PWR subsystems, whose geometries and compositions are described in section 7.1.

The first results to be presented concern BOC cases: even though cross sections are axially constant within each material region, if the polynomial cross sections option is enabled the code uses the same routines that are needed at higher burnup levels. As already stressed in section 3.4, the main cost of depletion analyses is due to the repeated application of the neutronics solver, so that, from the point of view of performances, zero-burnup configurations are sufficient to assess the benefit of the new method. Furthermore, the zero-burnup simulation is, in general, the one that takes more time.

A whole section is devoted to show the suitable quadrature orders for the evaluation of the transmission integrals $\{E_j\}$ (Eq. (5.7)), which are obtained through the adaptive procedure described in section 5.3.

Finally, the results of a complete depletion study are presented, regarding the most computationally expensive case study. Plots of the axial profiles of the total cross section are also provided, characterized by gradients that are locally very remarkable.

7.1 Case studies and input data

The three case studies treated are mainly distinguished by their 2D sections, given by a 5x5, 7x7 and full (17x17) PWR assembly grid. These are reported in Fig. 7.1, which also shows how structural grids are modeled: when they are present, each fuel rod is simply surrounded by an additional guide tube. A further, important difference between the systems is in the fuel composition: while the whole assembly contains only UOx, a UOx-MOx mix is present in the smaller cases which also have a relatively high concentration of gadolinium within some UOx cells. The fuel compositions is found in Tab. 7.1.

On the other hand, the axial mesh is the same for all cases, but of course it varies following the spatial approximation adopted for the neutron flux and cross sections along the z axis.

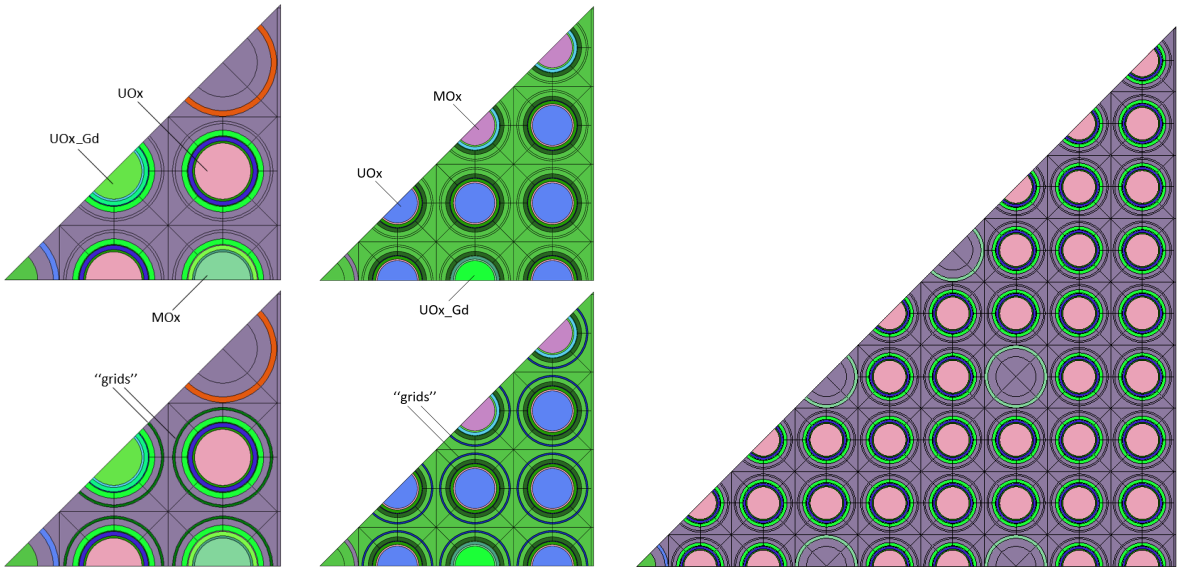


Figure 7.1: 2D geometries of the analyzed PWR systems: (a) 5x5 and (b) 7x7 sections with and without grids and (c) the whole 17x17 assembly (without grids, whose representation is straightforward).

Fig. 7.2 shows the three kinds of meshes involved: a coarse polynomial mesh (7.2a), which coincides with the material mesh[1] and is used for the flux calculation when adopting the new method; a 5^{th}-order Gaussian one (7.2b), required to perform self-shielding and depletion, as discussed in section 5.4, and obtained from the previous one by adding the Gaussian layers in the 3D regions where the polynomial expansion of cross sections is to be employed; a refined 87-layer mesh (7.2c), used for SC simulations. For symmetry reason, a 1/8 pattern is used radially and half of the column is treated along z. In doing so, starting from the bottom one has the axial reflector and the active height[2] containing three grid layers. The SC axial mesh and the three radial ones have been provided by [63]. It is important to point out that the definition of the Gaussian mesh is, at present, essential for the use of the polynomial cross sections approach: a polynomial region is actually identified by a set of Gaussian layers, adjacent along z, that contain different media. While assigning the material properties to the Gaussian regions, these will be distinguished if the corresponding 3D region in the coarse mesh is to be polynomial, that is, if it contains depleting isotopes. This mesh association is the same exploited when retrieving the expansion coefficients of cross sections. Based on the adopted Gaussian mesh, the axial layers that involve polynomial regions are therefore the ones belonging to the active height and not containing grids. The use of $m = 5$ layers for each polynomial region fulfills the $m > N_p$ requirement and represents a conservative choice aiming to avoid possible numerical errors.

In the following, a sequence of the form **ABB(-M#G#)(_fit)** will be used to uniquely

[1]By *material* one refers to the mesh that distinguishes two regions if and only if they have different compositions.

[2]The axial portion of the system wherein fission reactions take place.

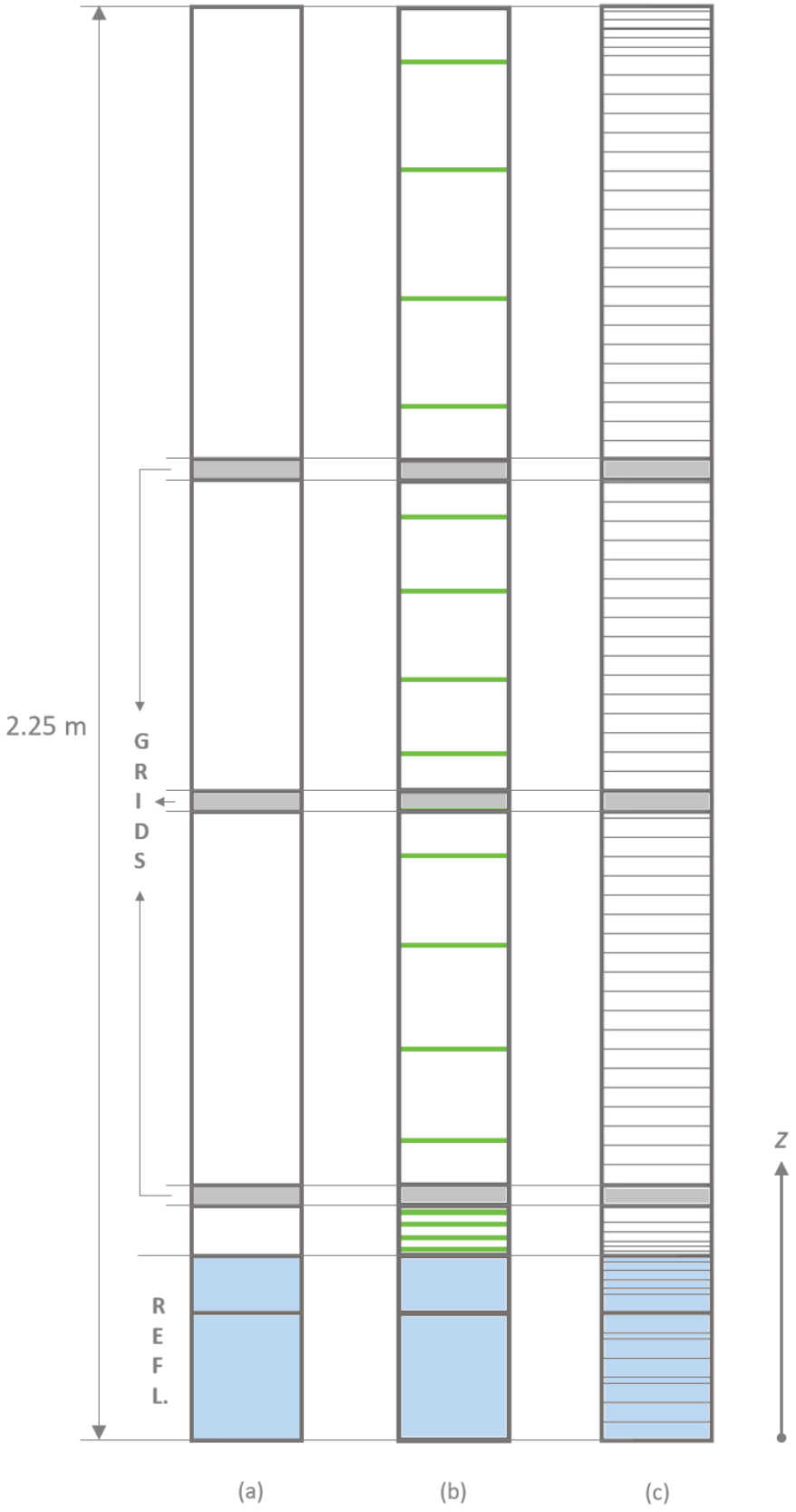


Figure 7.2: The axial meshes adopted: the polynomial (a) and its associated Gaussian mesh (b) and the step-constant one (c).

Table 7.1: Fuel compositions.

		UOx	**MOx**	**UOx_Gd**
Density [g/cm^3]		10.25	9.33	10.35
Fuel* mass fraction [%]				
	U-234	0.03	0.03	0.03
	U-235	3.70	0.38	3.66
	U-236	< 0.01	< 0.01	< 0.01
	U-238	96.26	97.42	95.17
	Pu-239	–	0.98	–
	Pu-240	–	0.98	–
	Pu-241	–	0.10	–
	Pu-242	–	0.10	–
	Gd-155	–	–	0.28
	Gd-156	–	–	0.28
	Gd-157	–	–	0.29
	Gd-158	–	–	0.29

**including* Gd*, if present*

identify the performed simulation, where

A can be “5”, “7” or “F”, to indicate the 2D geometry (5x5, 7x7 or full, respectively);

BB is either “SC” or “PX”, depending on the spatial approximation used (step-constant or polynomial for flux and cross sections);

M#G# denotes, for PX cases only, the axial mesh refinement (see section 7.2.1);

_fit if present, indicates the use of the non-linear least-square fitting for DP_N coefficients (described in section 6.4).

The input parameters for all calculations are reported in Tab. 7.2, involving the parameters for the Monte Carlo reference, provided by the TRIPOLI-4® code [64]. *Packet_length* is an option allowing for grouping batches of particles in packets and computing mean and variance on the resulting grouped series; this is done to counter the variance under-estimation, due to batch-to-batch correlation, when performing power iterations [64, 65].

In the way it is implemented, the polynomial method is compatible with any polynomial order. However, all the simulations performed have adopted a quadratic expansion; the reason is that, for the analyzed case studies, the spatial description provided good results from the accuracy standpoint, as will be shown in sections 7.2 and 7.4. Clearly, higher-order expansions might be preferable if the axial thicknesses considered could be further extended, which, according to the meshes described in the next section, was not possible in the applications presented.

The 5BB and 7BB simulations have been carried out on a compute node composed of two Xeon Silver 4214 @ 2.20 GHz CPUs, whereas the FBB ones on a node composed of two EPYC 7352 @ 2.3 GHz CPUs.

Table 7.2: Input parameters common to all simulations performed (except for the polynomial order).

TDT input data	
Axial polynomial order*	2
Δ_r $[cm]$	0.05
Δ_s $[cm]$	0.5
Anisotropy order	3
Azimuthal angles $(0, \pi)$	24
Polar angles $\left(0, \frac{\pi}{2}\right)$	4
Angular quadrature	Gauss-Legendre
Energy groups	281
Eigenvalue precision	10^{-5}
Fission integral precision	10^{-4}
OpenMP threads	48
TRIPOLI-4® input data	
Batches	10^5
Neutrons/batch	5000
Discarded batches	2000
Packet_length	100

*for PX cases

7.2 Zero-burnup comparisons

This section serves a twofold purpose: on one hand, it aims to show the performances and the goodness of the results of the proposed method by comparing them with the SC approach and with the available reference solutions at BOC conditions; on the other, it intends to present the efficiency of the DP_N acceleration, adapted to PX cases, and of the vectorization strategy followed in the implementation. The following paragraphs are meant to provide all the aspects that lead to prefer the new approach when dealing with depleted systems.

7.2.1 Axial mesh convergence study

The basic scope of higher-order spatial methods is to improve performances by reducing the number of regions; for the axial polynomial method, this translates into a smaller number of axial layers. It must be verified, however, that the chosen spatial order is sufficient for the adopted mesh, i.e. that it does not produce significant bias or instabilities. For this reason, the results of the polynomial mesh of Fig. 7.2a have been compared to the ones of more refined meshes, as well as to the SC approximation and the Monte Carlo reference. Here the validation concerns only the k_{eff} eigenvalue, whereas condensed reaction rates per cell are dealt with in the next paragraph.

As anticipated, the simulations carried out are identified by a sequence indicating, for the

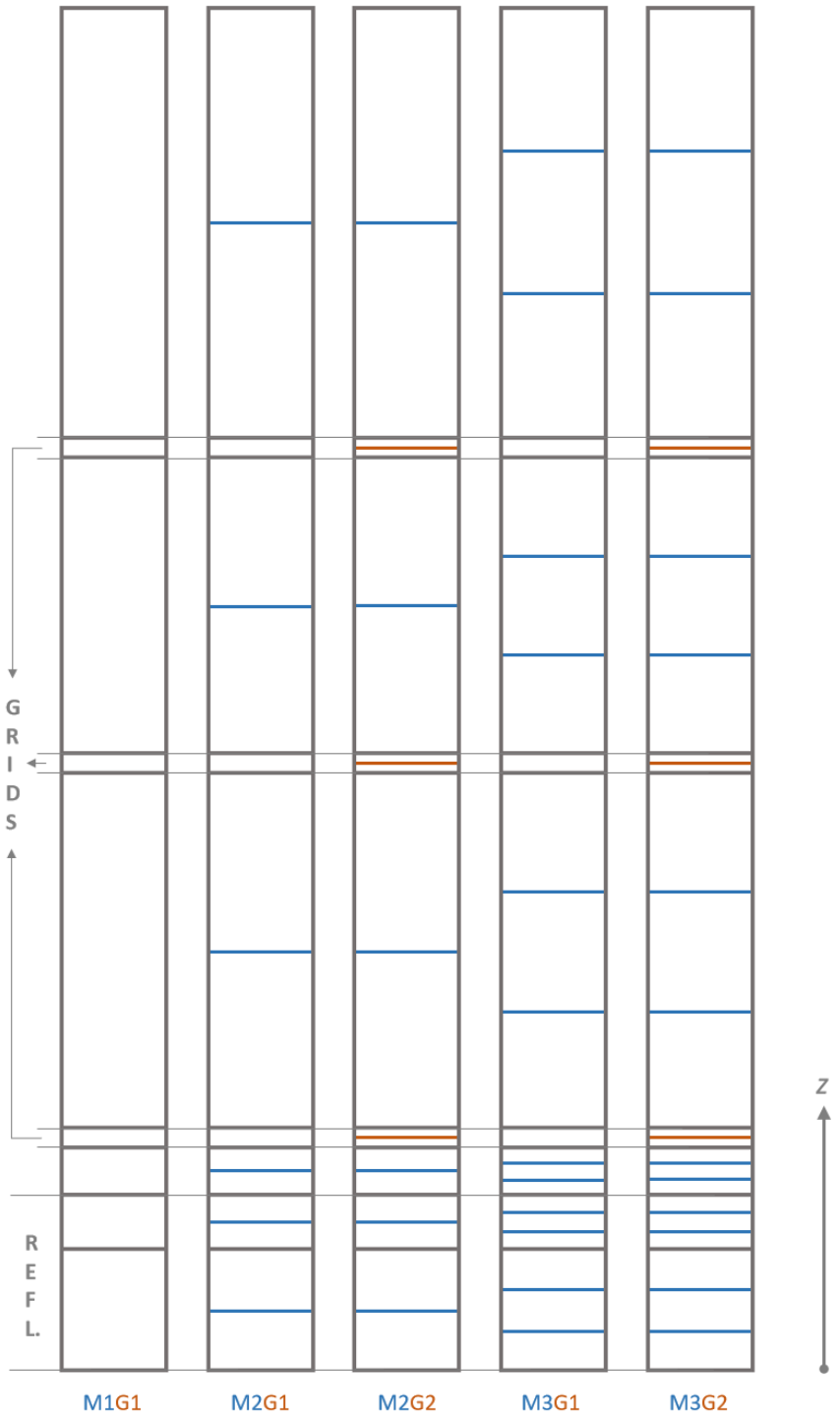


Figure 7.3: Polynomial axial meshes for convergence study.

Table 7.3: PX results for different axial meshes compared to the SC and TRIPOLI-4® ones. The highlighted columns refer to the methods and meshes that will be adopted for the depletion of the full assembly.

5x5						
T-4 k_{eff}	$0.90884 \pm 4\,pcm$					
	5PX-M1G1	5PX-M2G1	5PX-M2G2	5PX-M3G1	5PX-M3G2	5SC
$\frac{\Delta k_{\text{eff}}}{k_{\text{eff}}}^*$ $[pcm]$	-33	-29	-27	-27	-25	-94
Memory $[GB]$	14.05	19.61	21.81	25.16	27.36	11.83
Time $[h]$	1.21	1.68	1.87	2.10	2.43	2.83
7x7						
T-4 k_{eff}	$1.07228 \pm 4\,pcm$					
	7PX-M1G1	7PX-M2G1	7PX-M2G2	7PX-M3G1	7PX-M3G2	7SC
$\frac{\Delta k_{\text{eff}}}{k_{\text{eff}}}^*$ $[pcm]$	-140	-134	-127	-132	-124	-201
Memory $[GB]$	26.86	39.07	43.99	51.28	56.19	25.67
Time $[h]$	1.97	3.21	3.20	4.08	5.83	5.66

**with respect to reference* ($= \text{TDT}\,k_{\text{eff}}/\text{T-4}\,k_{\text{eff}} - 1$)

PX cases, which mesh is adopted. The two #'s in "M#G#" represent, in fact, the number of axial meshes considered within each material region, except for the grids, and that specific of grid regions, respectively. For instance, M2G1 will have a mesh coinciding with the material one in the grids and all other regions divided into two. Clearly, M1G1 is exactly the material mesh 7.2a. Figure 7.3 reports the ensemble of meshes used for the comparison with the corresponding sequences.

For the 5x5 and 7x7 2D grid cases, Tab. 7.3 shows the results of five calculations adopting the polynomial method with different axial meshes and of the SC approach, in terms of memory, time and relative difference of k_{eff} from the reference value. One can deduce that:

- regardless of the mesh adopted for the PX simulations, they always result more accurate than the SC one;
- the accuracy gain due to the mesh refinement is very limited, which justifies the use of the sole M1G1 axial grid for all PX calculations henceforth (assuming that this remains valid for the full assembly case);
- the memory requirements are quite comparable if the coarser mesh is used for PX's, with the traditional approach having a slight advantage; yet, the ratio between the PX and SC demands is lower in the bigger system, which anticipates the reversal occurring with the full assembly (section 7.4);
- the polynomial method entails a huge reduction in computing time that holds true even in most of the refined cases.

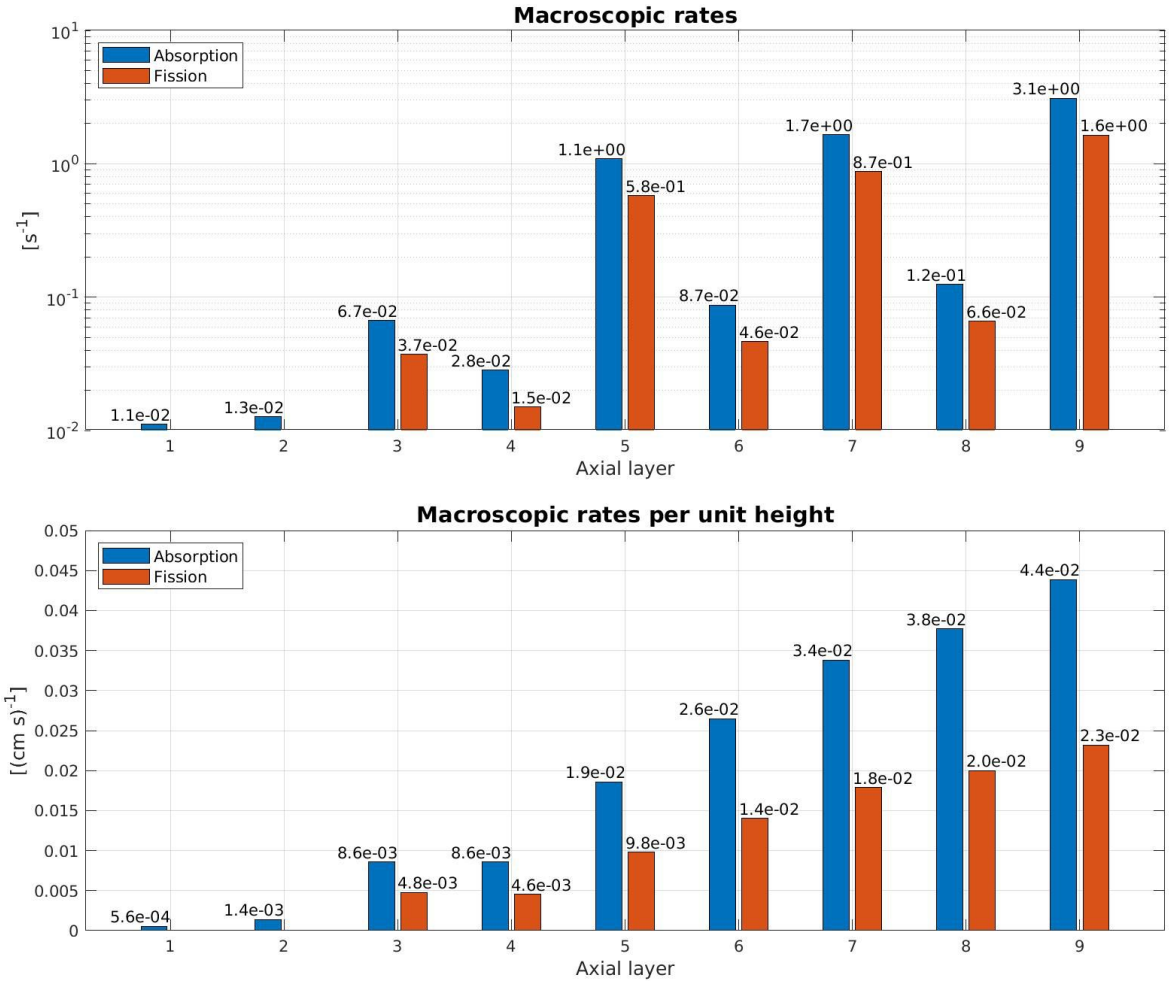


Figure 7.4: Macroscopic rates per axial layer of the FPX-M1G1 case at BOC. The values correspond to a production of one neutron per second within the calculation geometry (1/8 2D domain, half-column assembly).

With regards to the difference in k_{eff} accuracy between SC and PX simulations, it is important to underline that this is exclusively due to the different axial descriptions: a further refinement of the SC axial mesh would inevitably degrade performances, which are already worse than the PX ones as to computing time.

7.2.2 Spatial validation

The discussion dwells now on a more detailed comparison, involving macroscopic reaction rates (Eq. (2.4)): for every square cell containing a fuel rod or a water tube and each of the 9 axial floors of M1G1, the absorption and fission rates of TDT are compared to the ones produced by TRIPOLI-4®. To do this, energy-condensed values are considered, obtained adding up the contributions of all groups. Here, results refer to the FPX case.

In order to get a sense of the orders of magnitude implicated, Fig. 7.4 shows, for each floor, the volume-integrated rates and their values per unit height: concerning the latter, it is worth

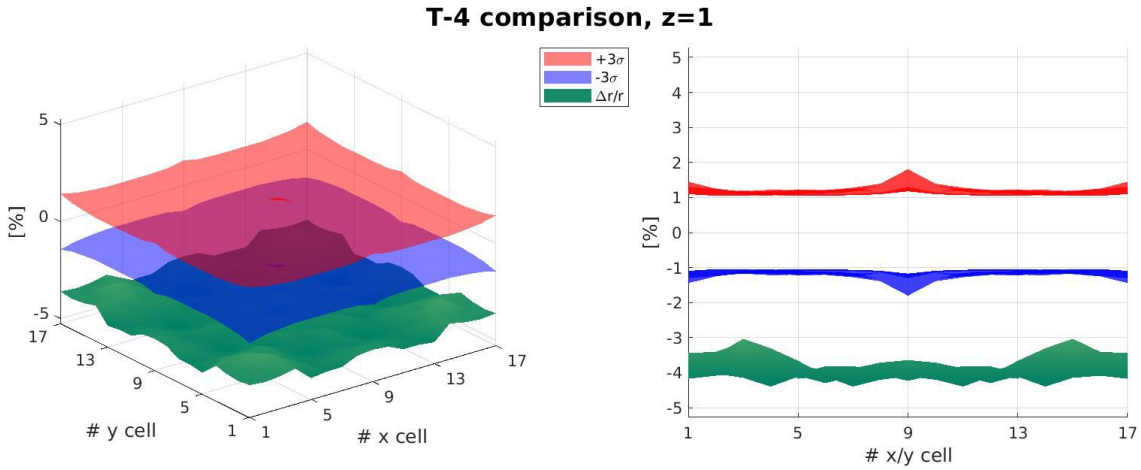


Figure 7.5: Comparison of absorption macro-rates for the 1st axial floor.

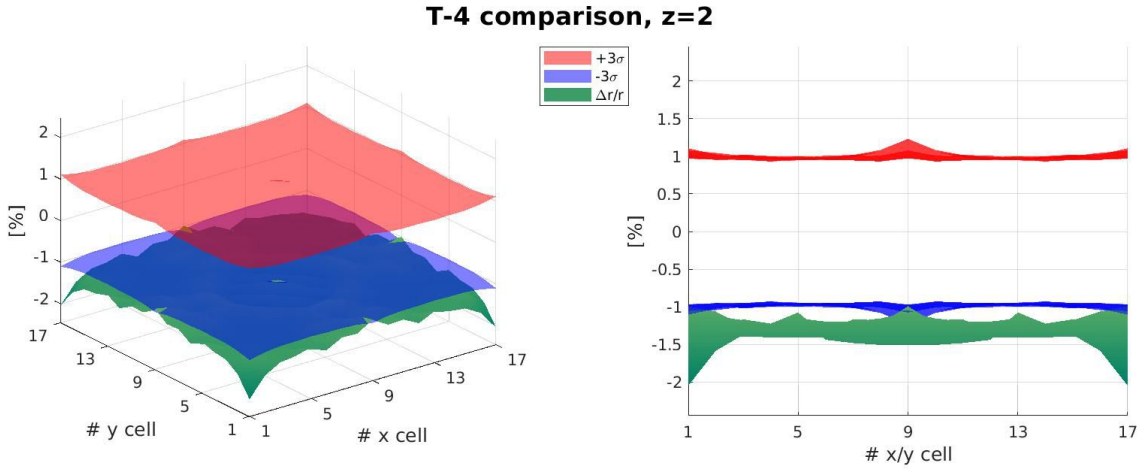


Figure 7.6: Comparison of absorption macro-rates for the 2nd axial floor.

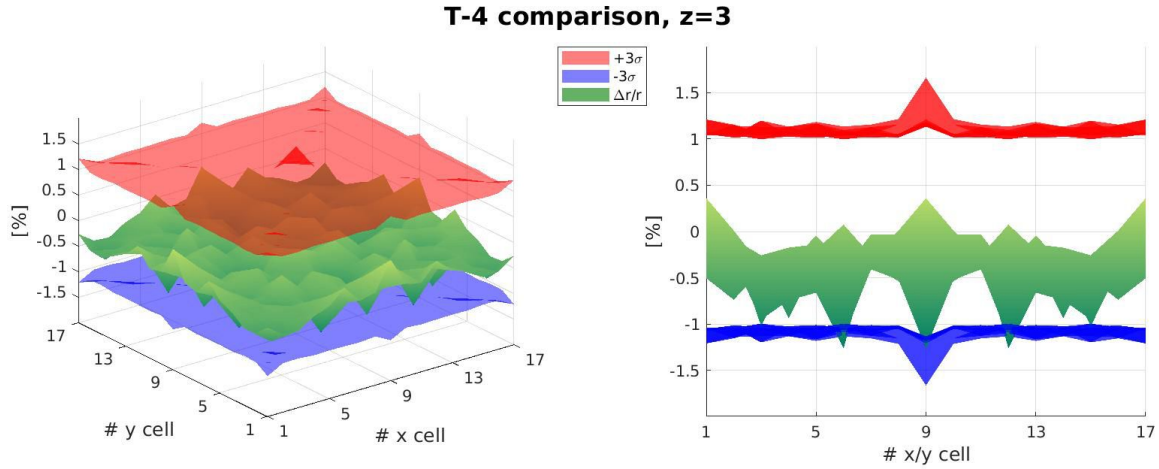


Figure 7.7: Comparison of absorption macro-rates for the 3rd axial floor.

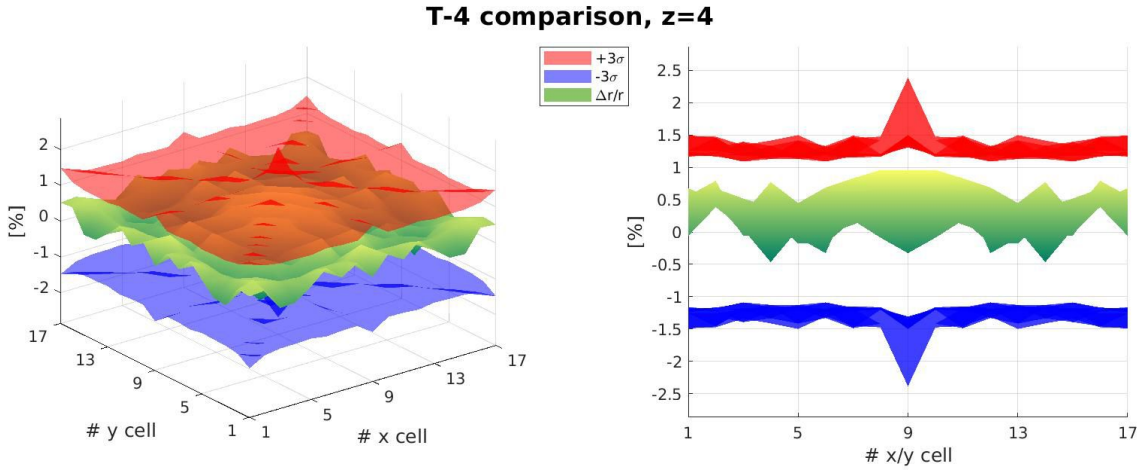


Figure 7.8: Comparison of absorption macro-rates for the 4$^{\text{th}}$ axial floor.

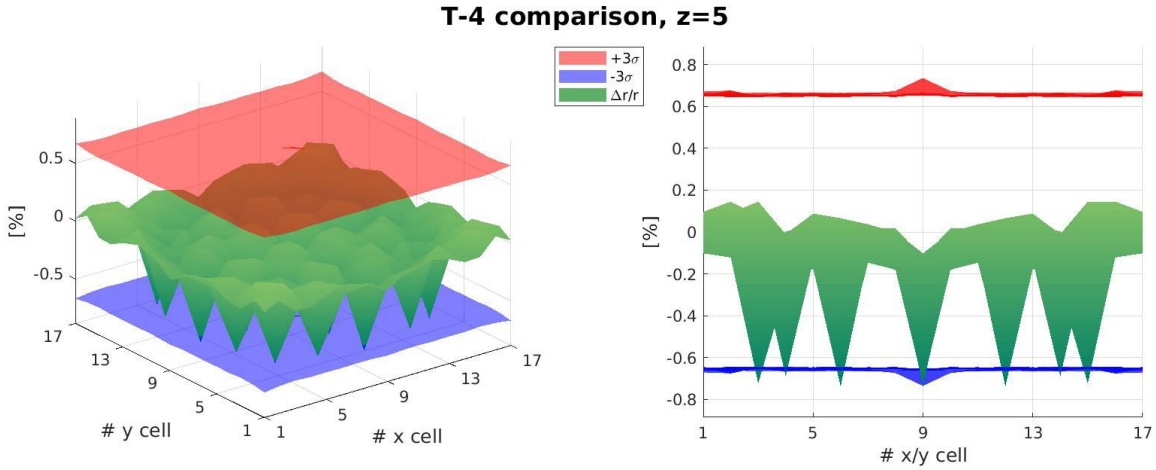


Figure 7.9: Comparison of absorption macro-rates for the 5$^{\text{th}}$ axial floor.

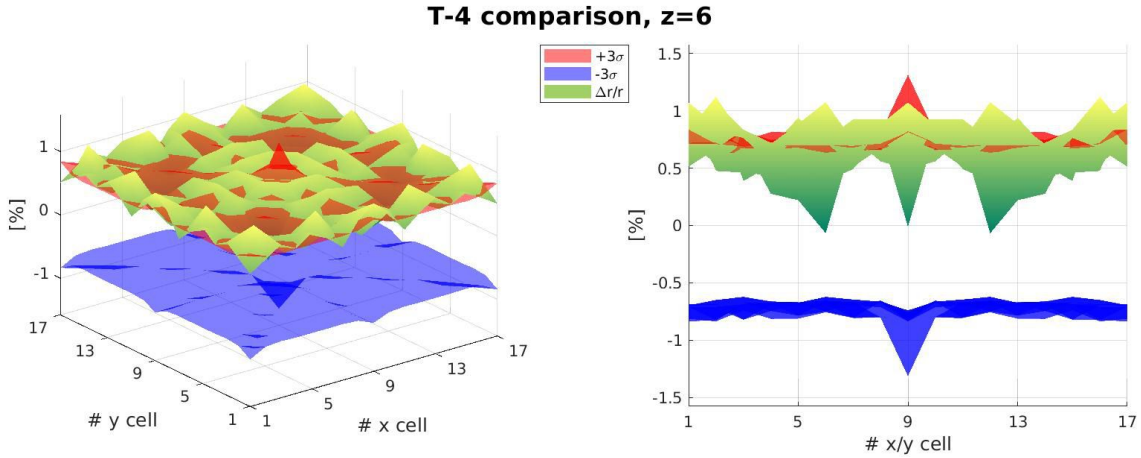


Figure 7.10: Comparison of absorption macro-rates for the 6$^{\text{th}}$ axial floor.

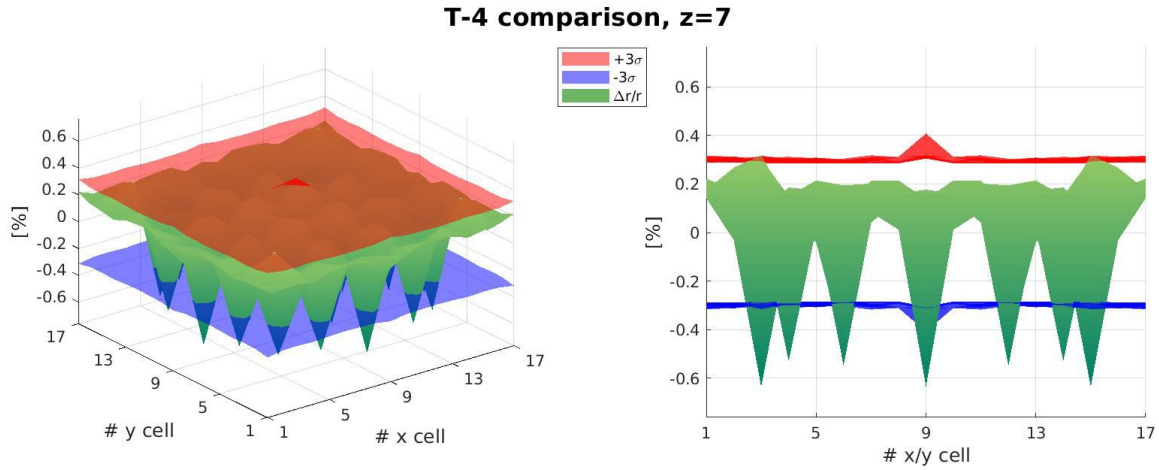


Figure 7.11: Comparison of absorption macro-rates for the 7$^{\text{th}}$ axial floor.

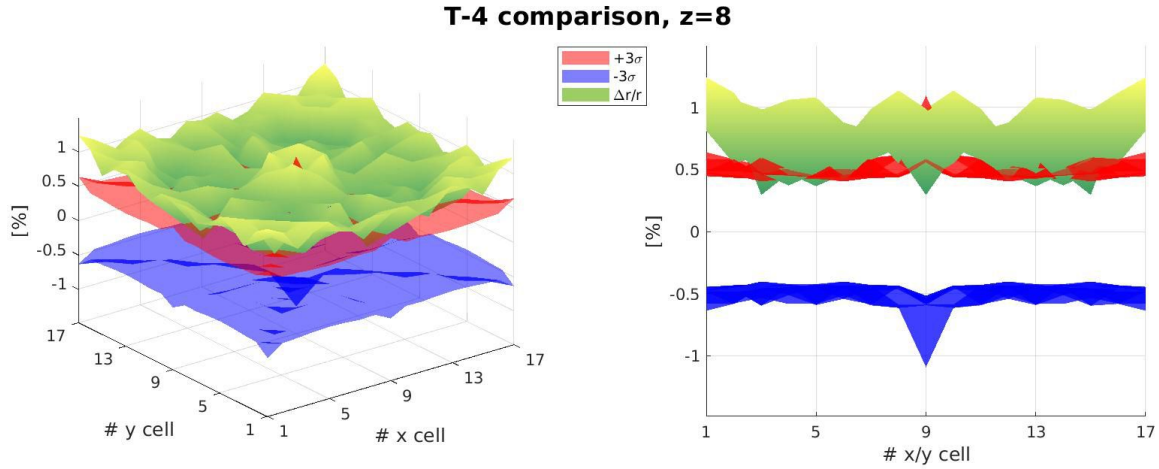


Figure 7.12: Comparison of absorption macro-rates for the 8$^{\text{th}}$ axial floor.

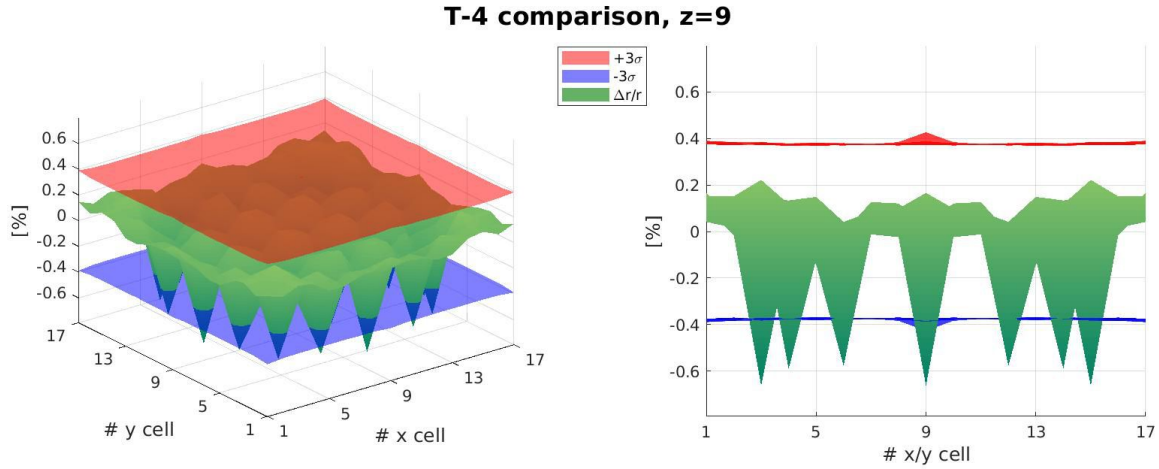


Figure 7.13: Comparison of absorption macro-rates for the 9$^{\text{th}}$ axial floor.

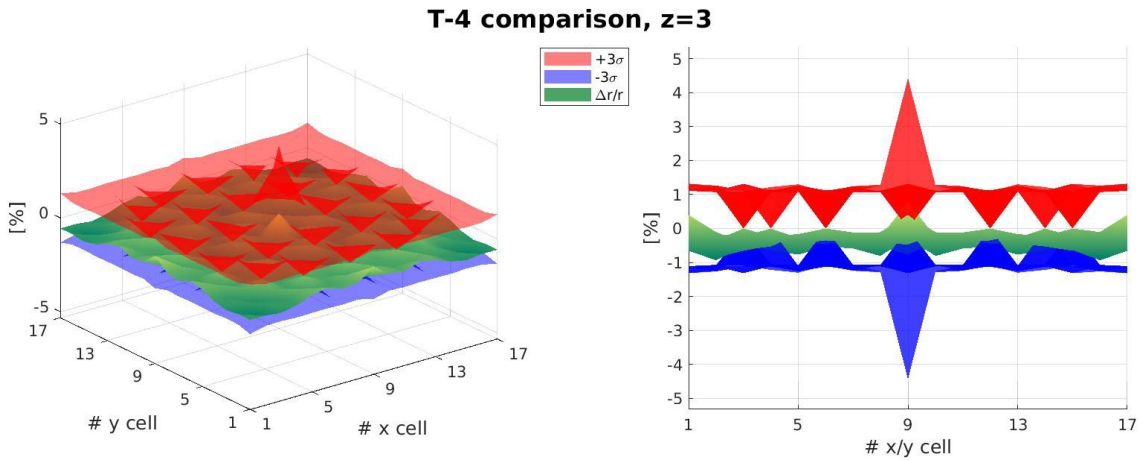


Figure 7.14: Comparison of fission macro-rates for the 3$^{\text{rd}}$ axial floor.

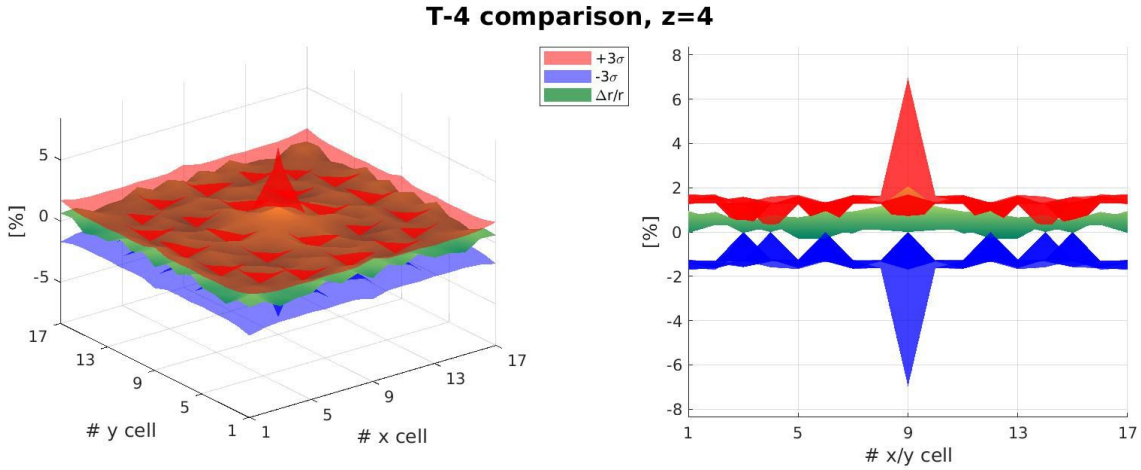


Figure 7.15: Comparison of fission macro-rates for the 4$^{\text{th}}$ axial floor.

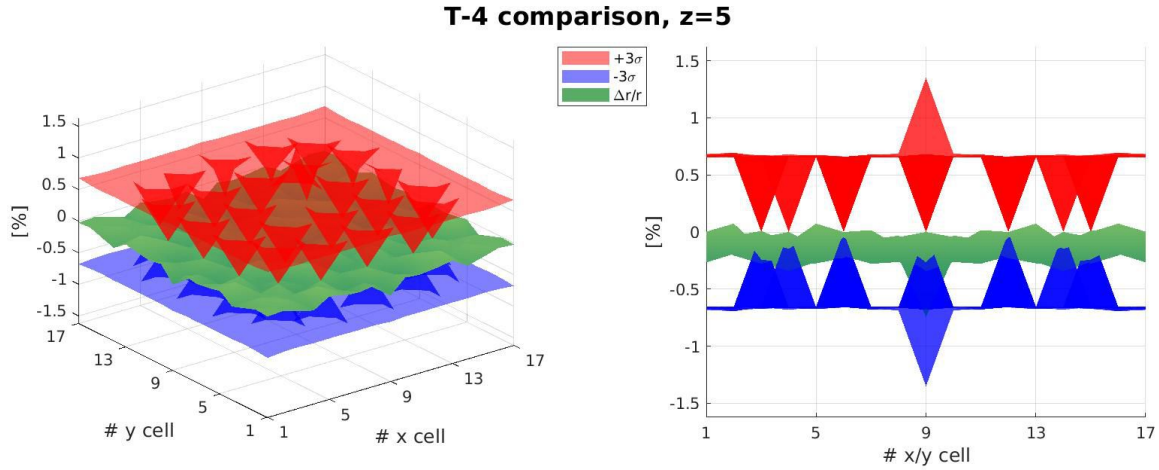


Figure 7.16: Comparison of fission macro-rates for the 5$^{\text{th}}$ axial floor.

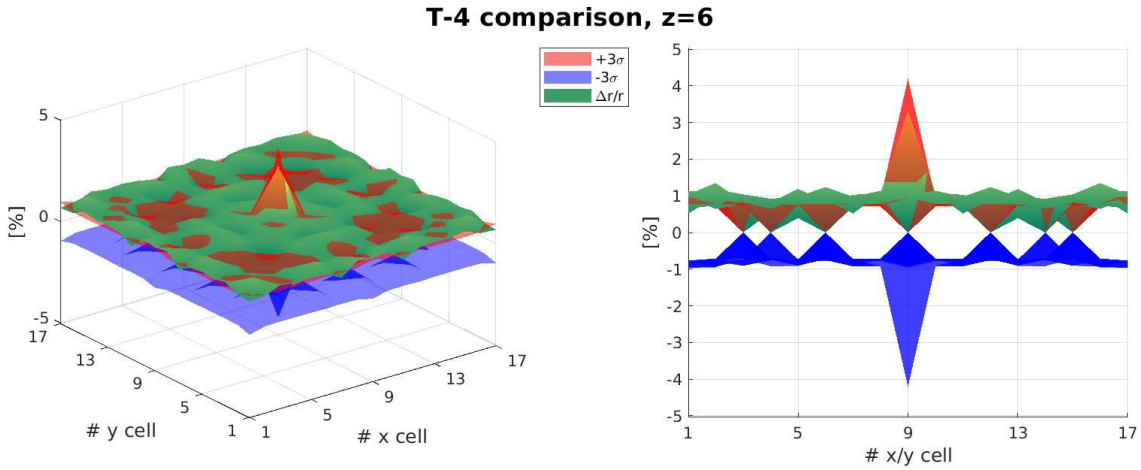


Figure 7.17: Comparison of fission macro-rates for the 6$^{\text{th}}$ axial floor.

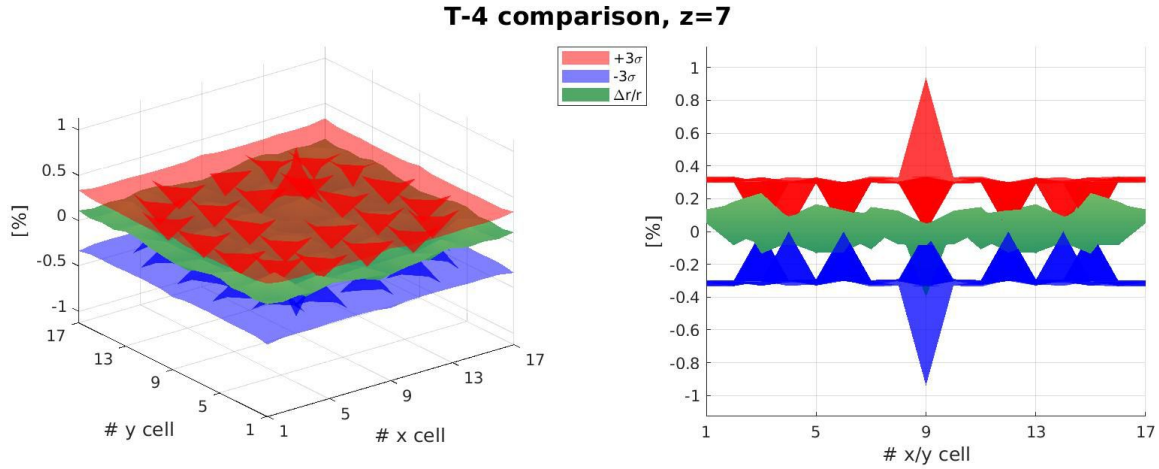


Figure 7.18: Comparison of fission macro-rates for the 7$^{\text{th}}$ axial floor.

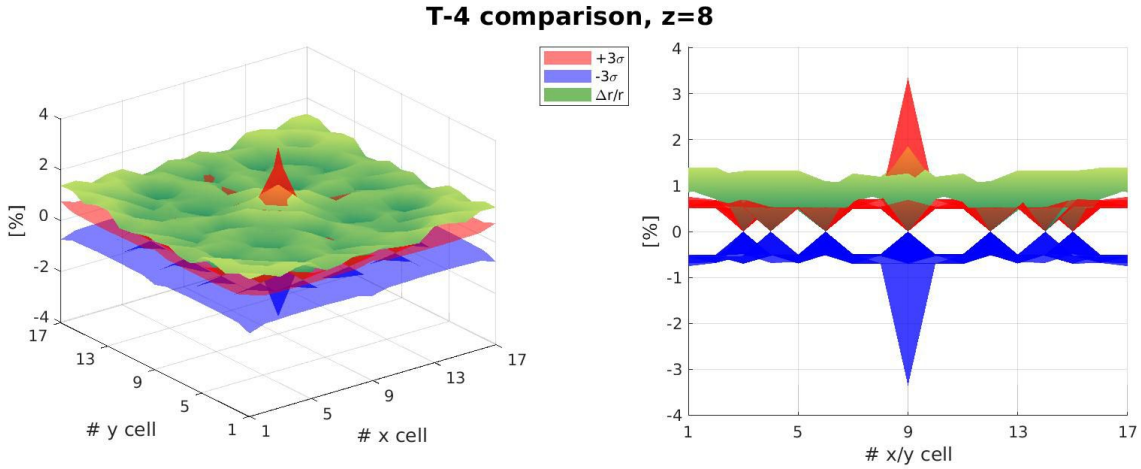


Figure 7.19: Comparison of fission macro-rates for the 8$^{\text{th}}$ axial floor.

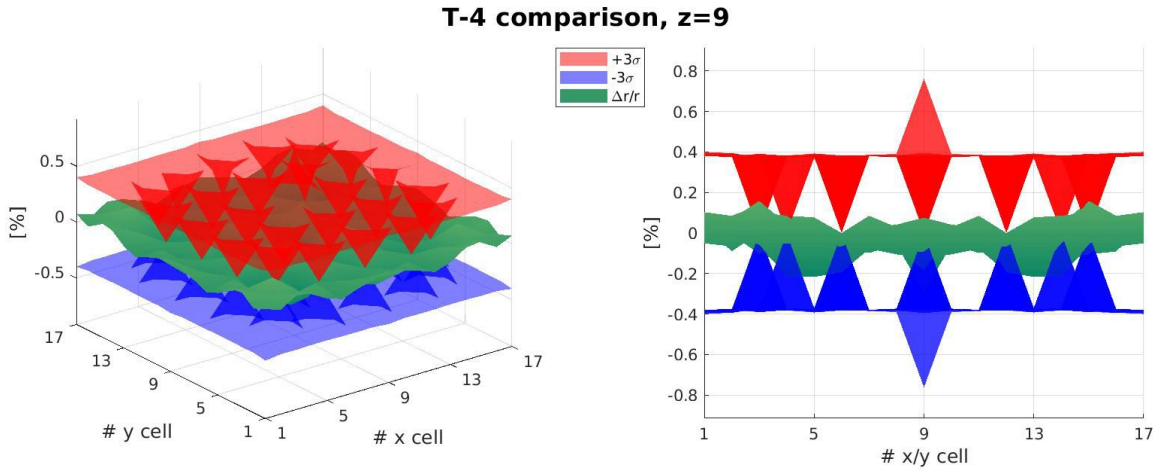


Figure 7.20: Comparison of fission macro-rates for the 9$^{\text{th}}$ axial floor.

noting that the absorption rates in the two reflector layers ($z = 1$ and 2) are respectively 57 and 22 times smaller than the average value in the active height (3.14 $10^{-2}\, cm^{-1}s^{-1}$). Per-cell values can be found in Annex B.

Figs. 7.5 - 7.20 report the rates relative error with respect to the reference ($\Delta r/r = r_{TDT}/r_{T-4} - 1$, r_{TDT} and r_{T-4} being the rates obtained by TDT and TRIPOLI-4®, respectively) and the two 3D surfaces corresponding to ± thrice the relative standard deviation. In the active height, the TDT results prove to be in good agreement with the Monte Carlo ones, with the error never being higher than 1.24% for absorption and 1.40% for fission (3.32% in the central cell, due to a very low concentration of uranium dissolved in water[3], which also makes TRIPOLI-4® farther from convergence). It can be seen that the errors tend to be higher in the axial floors of smaller height, in particular within the grids: this may be due to the self-shielding strategy adopted, which treats independently the different layers as extruded systems of infinite height; for such regions, better errors might be obtained if the currents across horizontal interfaces were considered. As for the reflector, the significantly higher errors (-3.30% ÷ -4.40%) can be tolerated taking into account the much smaller values of the rates in these regions. Such errors may be due to the Livolant-Jeanpierre method adopted for self-shielding, whose main assumption, the fine-structure description of the flux, is hardly verified when significant amounts of intermediate isotopes (Fe, Cr, Zr and Sn) are present. Better results might be obtained using the subgroups approach [66] but, as mentioned in section 3.1.2, the APOLLO3® implementation of this method also adopts the fine-structure hypothesis. Another possibility, which would allow one to eliminate the error due to cross sections condensation, would be to perform the Monte-Carlo simulations with the same multi-group cross sections used in TDT; unfortunately, such intercommunication between APOLLO3® and TRIPOLI-4® is not a functionality that is currently available. The already mentioned batch correlation, occurring when performing power iterations in Monte Carlo [65], leads to a variance under-estimation; to a certain extent, this effect is

[3]This dilution simulates the presence of a fission-chamber detector in the central water tube. A more accurate treatment would define its actual location along z but, unless a non-conforming axial mesh is used, it would also require additional extrusion layers for all 2D regions.

Table 7.4: Performance comparison between free and accelerated power iterations, the latter either in case of direct storage (ACC) or non-linear fitting (ACC_fit) of DP_N coefficients.

	FREE	**ACC**	**ACC_fit**
5PX-M1G1			
k_{eff}	0.90847	0.90854	0.90853
# outer iters	192	11	10
# inner iters	376 031	3091	2810
# DP_N outer iters	–	121	124
Memory [GB]	7.95	14.05	9.48
Time [h]	53.96	1.21	1.56
7PX-M1G1			
k_{eff}	1.07070	1.07077	1.07077
# outer iters	188	9	8
# inner iters	333 919	2529	2248
# DP_N outer iters	–	106	101
Memory [GB]	13.03	26.86	14.87
Time [h]	88.19	1.97	2.27

mitigated by the *Packet_length* option. In order to approximately quantify the difference with respect to the true variance values, a 1/4 assembly pattern has been considered (doubling the original system) and, for each pair of symmetric cells, the resulting mean values and standard deviations have been computed. In doing so, it has been seen that the number of cells for which the new standard deviations are greater than 1% of the mean and that undergo an increase of more than 10% are just 9 for the absorption rates and 4 for the fission ones, out of the 900 total cells (100 per axial plane). It can be inferred that, even though under-estimated, the standard deviations would remain quite small even if the correlation effect was removed.

7.2.3 Acceleration and vectorization performances

This short paragraph compares the results of the free and accelerated iterations and shows the performance gain due to vectorization. Concerning the first aspect, Tab. 7.4 contains the outcomes of the PX method applied to the 5x5 and 7x7 assemblies, distinguishing the cases of direct storage and non-linear fitting of the acceleration matrices. At the cost of roughly double the memory demand, the direct storage allows reducing the computing time by a factor ~ 45. The results are slightly worse when the fitting technique is employed, but the associated memory is much closer to the free-iteration case.

The vectorized construction of the transport and acceleration coefficients (sections 5.2 and 6.3, respectively) contributes quite significantly to limiting the computing time. Tab. 7.5 reports, for the same cases of before, the percentage time reductions achieved through vectorization: as can be seen, the relative advantage for the coefficients evaluation is much higher

Table 7.5: Reductions in computing time due to vectorization. For transport coefficients, the values refer to the reduction of the longest time among the adopted threads, cumulated over the first outer iteration.

	Time reduction [%]
5PX-M1G1	
Global	−27.8
Transport coefs	−46.2
Acceleration coefs	−45.9
7PX-M1G1	
Global	−20.1
Transport coefs	−42.8
Acceleration coefs	−41.4

than the global gain, which can be explained by considering that the involved routines are particularly suited to be vectorized, since they perform the same operations over a generally large number of V-chords.

7.3 Adaptive Gauss-Legendre: required orders

Section 5.3 dealt with the adaptive technique that determines the suitable quadrature orders for the approximation of Eq. (5.7) integrals. Here the aim is to show the obtained order statistics based on the chosen tolerance at BOC and on the burnup level for a given tolerance. Moreover, a comparison among the j indices of the integrals is presented, focusing on the requirement of each of them at 0 GWd/t; this is notably different from the approach followed in the actual simulation, where, to optimize the orders evaluation and to deal with smaller arrays, a single quadrature order is adopted for the set $\{E_j,\, j = 0, 2N_p\}$, equal to the highest order among the ones required by the single integrals.

The case study the results of this paragraph refer to is the FPX-M1G1, which contains only V- and M-chords. Every graph type is thus presented twice, once for each chord class. It is also important to note that, although the percentages shown below account for all energy groups, in the code a dimension of the arrays containing the quadrature orders is reserved to the group dependence.

According to the tolerance - Figs. 7.21

This comparison considers three tolerance levels and plots, for each quadrature order ranging from 2 to the chosen maximum (20), the fraction of chords for which that order is sufficient. All the other results presented in this chapter adopt a tolerance of 10^{-6}. The graphs show that:

- for all tolerances analyzed, 5 Gauss points are sufficient for more than 90% of chords;

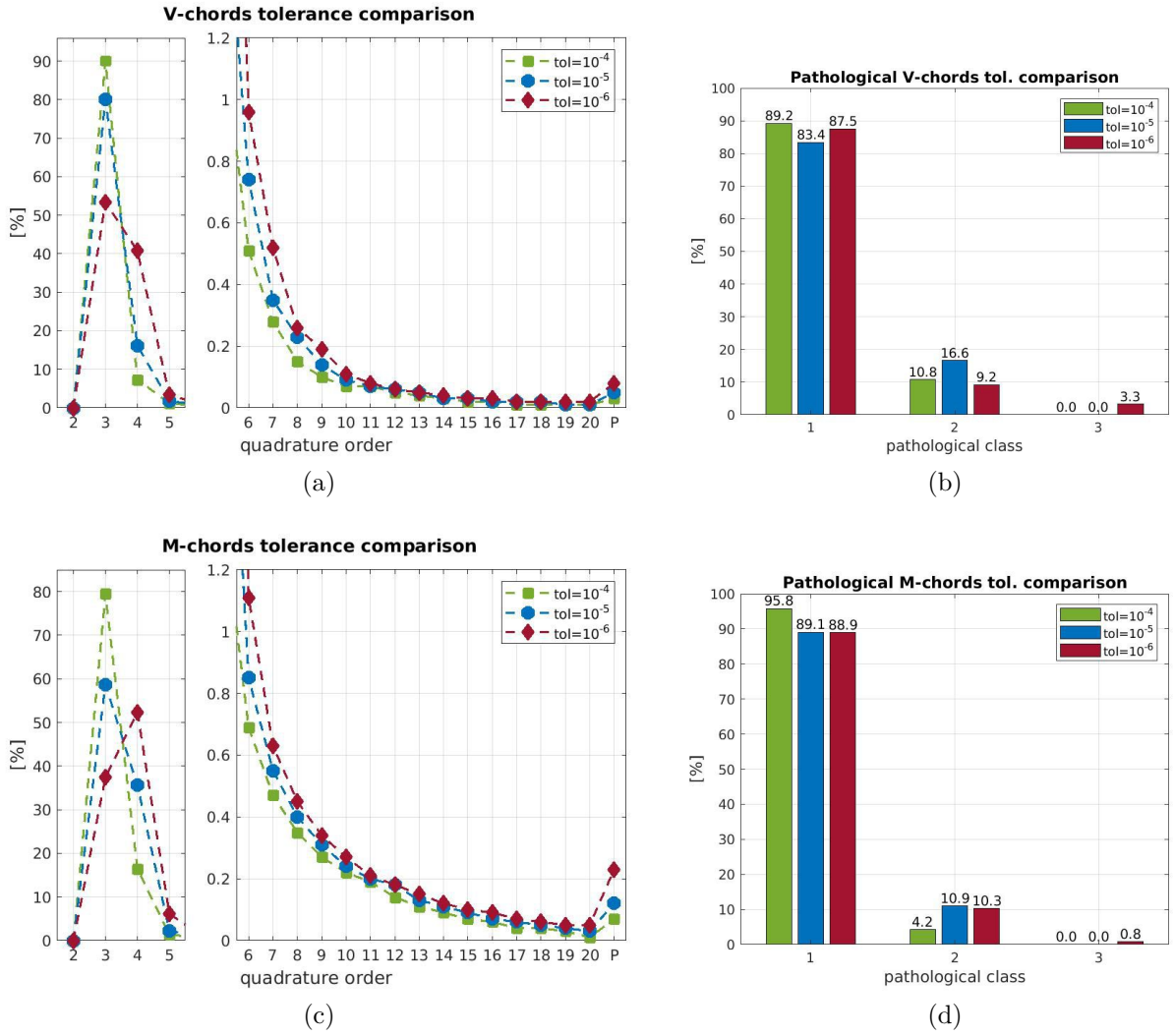


Figure 7.21: Required quadrature orders and pathological classes for V-chords (a, b) and M-chords (c, d) according to the tolerance.

- after reaching the maximum the chord fraction monotonically decreases as the order grows;
- narrowing the tolerance reduces the maximum fraction, shifts it rightward, increases the fraction of pathological chords and, passing from 10^{-5} to 10^{-6}, the number of pathological classes needed.

According to the j index - Figs. 7.22

Analyzing the distributions quadrature orders for the different j's (that is, for the different powers of t in the integrand), two opposed behaviors can be distinguished: on one hand, in the low-order range the chord fractions vary considerably and, as j increases, they shift to the right until coinciding to those of Figs. 7.21a and 7.21c for $j = 2N_p = 4$; on the other hand, the values tend to overlap as higher orders are considered.

For low orders, the reason for the fractions distribution is probably related to the smooth

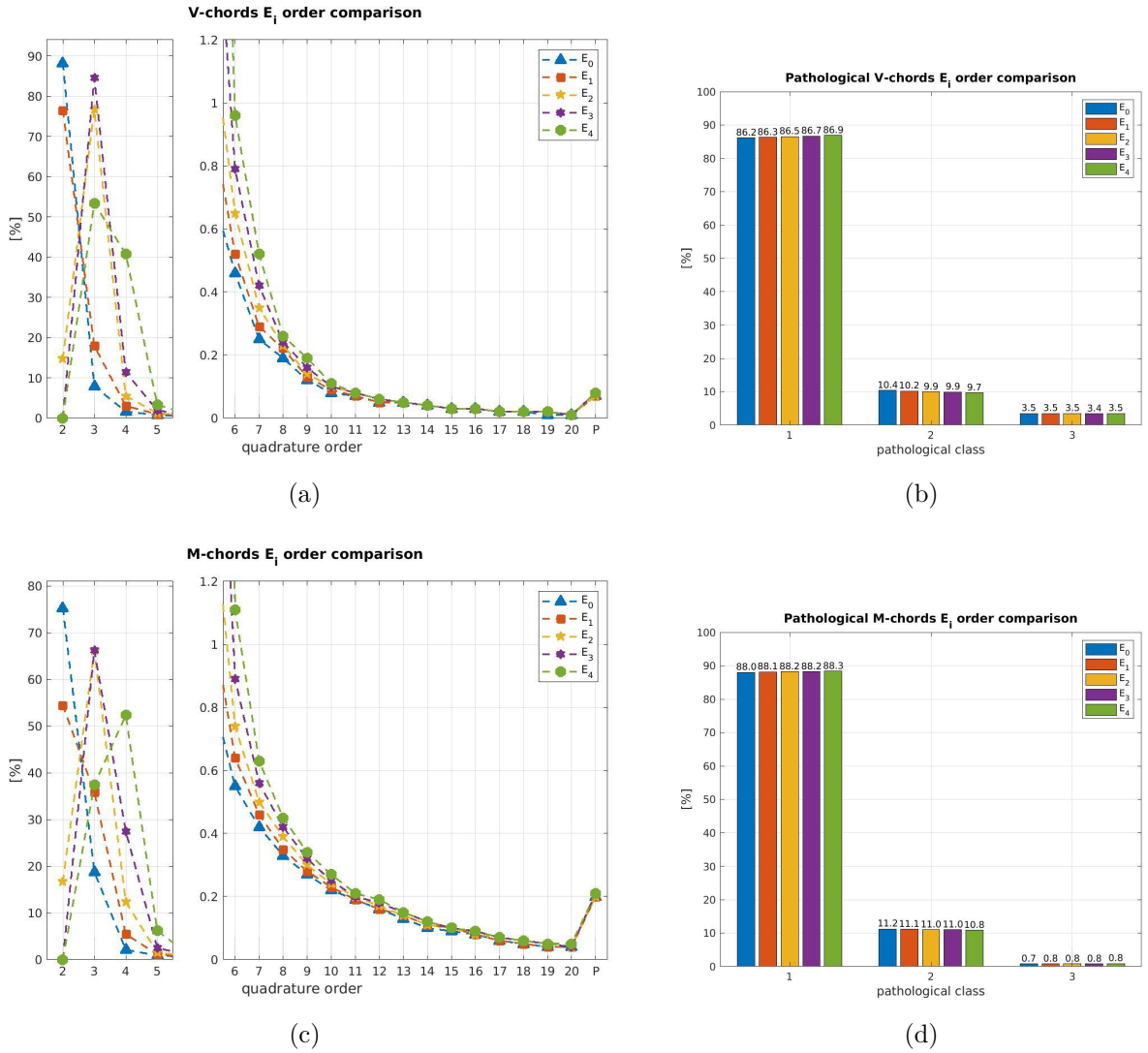
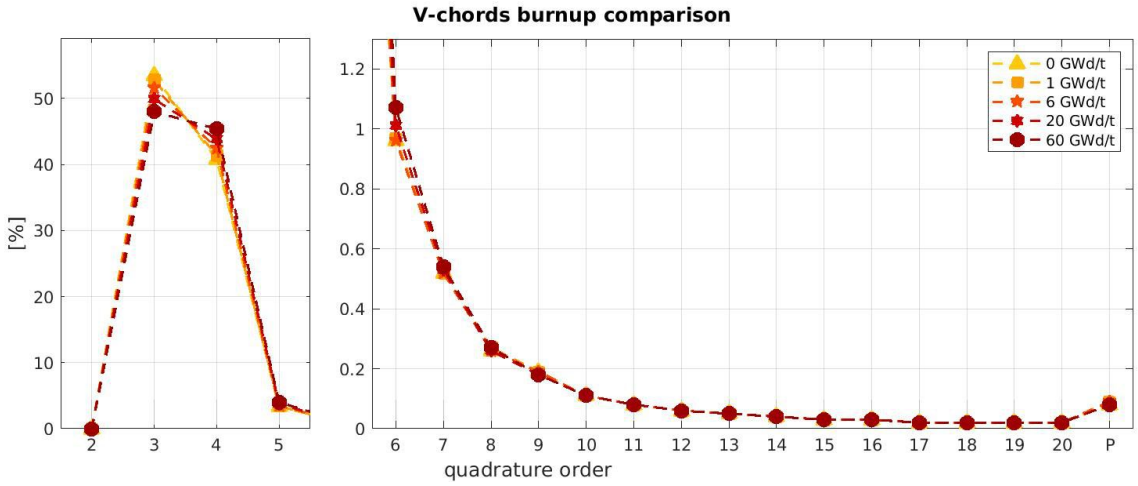


Figure 7.22: Required quadrature orders of V-chords (a, b) and M-chords (c, d) according to the j index.

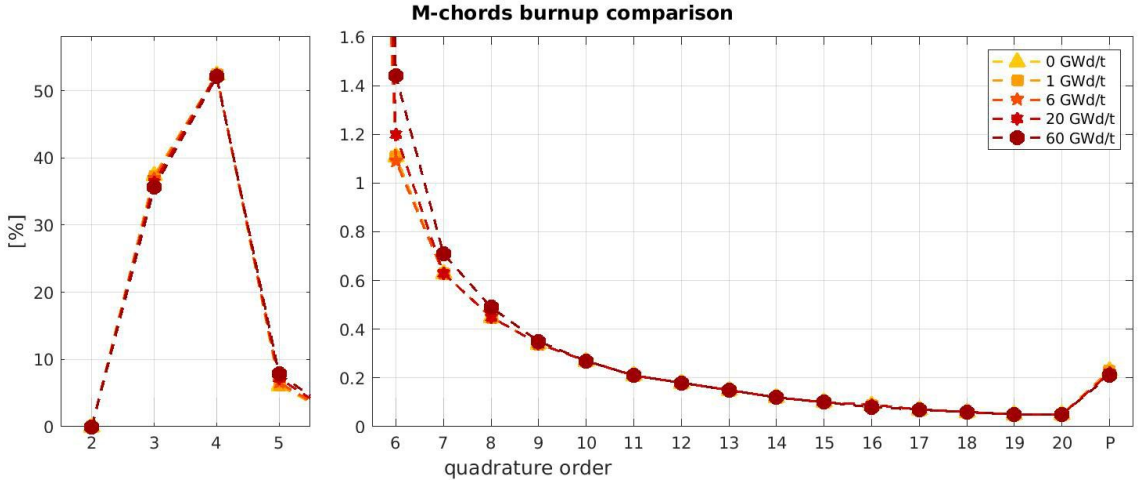
profile of the integrands: in fact, if a number m of Gauss points is enough for the evaluation of E_0, the $e^{\tau(s)-\tau(l)}$ function is equivalent, for the chosen tolerance, to a polynomial of degree $2m-1$ or lower. Multiplying this function by t^j adds j to the degree, so that the relative degree variation, which leads to require an order $m^* \geq m$, is clearly more relevant the lower $2m-1$ is. On the contrary, as the exponential behavior of the integrand becomes more and more dominant, the impact on the quadrature order vanishes quickly and the statistics tend to be the same for all j's.

According to the burnup - Figs. 7.23

The last comparison proposed is maybe the most interesting from a practical point of view, since it shows, for the depletion study described in the next section, how the required quadrature points vary with the burnup. Indirectly, this provides also an estimation of how much



(a)



(b)

Figure 7.23: Required quadrature orders of V-chords (a) and M-chords (b) according to the burnup level.

the isotope depletion modifies cross sections, with regard to the optical path length value for a given chord ($\tau(l)$) and of its polynomial profile along the chord itself ($\tau(s)$), since both these aspects may impact on the quadrature rule needed.

According to the graphs, the increasing burnup determines a slight rightward shift of the distributions, suggesting a likewise limited overall variation in cross sections. However, an energy-condensed description like this is subjected to compensation effects, as nothing prevents the statistics of some groups from shifting in the opposite direction to the global displacement. An example of this behavior is shown in the next section, while presenting the cross sections axial gradients at the end of depletion.

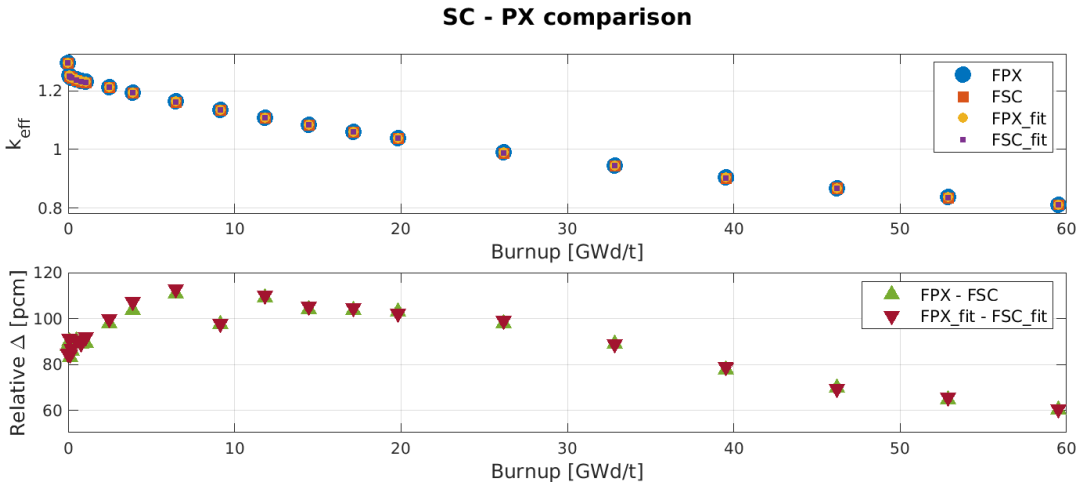


Figure 7.24: Depletion eigenvalue results and relative differences between SC and PX.

7.4 Depletion study

This final section is meant to prove the advantage of the polynomial cross sections approach in the specific type of calculations for which it has been developed. An depletion study from 0 to 60 GWd/t, constituted by 21 flux calculations, has actually been performed using both the traditional SC and the new PX method, with the axial meshes and input parameters described in section 7.1 and considering the full assembly case study.

The resulting performances can be found in Tab. 7.6, which reports, for each approach, the memory and time demanded in case of direct storage and non-linear fitting of the acceleration coefficients. For the first depletion step (burnup = 0 GWd/t), the k_{eff} eigenvalues have been compared with TRIPOLI-4®, confirming the greater accuracy of the polynomial method already observed for the smaller assemblies (Tab. 7.3). The memory occupied by the macrolibrary is also shown, to demonstrate the usefulness of adopting the 5^{th}-order Gaussian mesh (Fig. 7.2b) for depletion and self-shielding: while using the 87-layer mesh (Fig. 7.2c)

Table 7.6: Main input data and compared results of the depletion study of a PWR assembly. The validation against the TRIPOLI-4® reference concerns only the first depletion step.

Depletion comparison				
T-4 zero-burnup k_{eff}	$1.29728 \pm 4\,pcm$			
	FSC	**FSC_fit**	**FPX-M1G1**	**FPX-M1G1_fit**
$\Delta k_{eff}/k_{eff}$ *,** $[pcm]$	-126	-126	-39	-42
Memory $[GB]$	144.72	62.60**	119.30	52.30**
Macrolibrary $[GB]$	12.20	12.20	5.55	5.55
Global time $[h]$	258.87	294.06	105.25	112.80

* *with respect to reference*
** *at BOC*

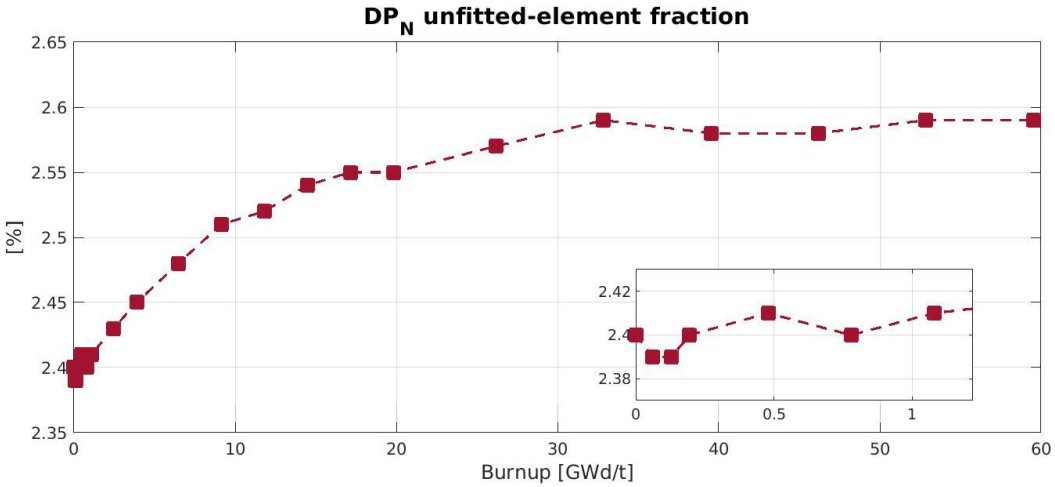


Figure 7.25: DP_N unfitted-element fraction at every flux calculation of the depletion analysis.

would lead to a 14% increase (13.93 GB), the Gaussian one allows a considerable reduction of 55%. This aspect relating to the macrolibrary holds true for the 5PX and 7PX cases, but for the full assembly the global memory becomes lower too, as anticipated in paragraph 7.2.1: respectively, a 18% and a 16% reduction is measured for the direct-storage and non-linear fitting cases. The most relevant result, however, is given by the impressive decrease in the computing time: the higher-order method makes it possible to attain the end of the sequence in 41% or 38% of the time, depending again on the strategy followed for the DP_N coefficients. These results are all the more relevant considering the greater k_{eff} accuracy, at BOC, of the higher-order approximation.

Figure 7.24 provides a verification for the new method in terms of k_{eff} eigenvalues, reporting the results and the relative differences of the FSC and FPX cases: as can be seen, the latter always remain below 120 *pcm*. Furthermore, as shown in Fig. 7.25, when applying the fitting technique to FPX the fraction of unfitted matrix elements undergoes a very slight increase and then tends to saturate starting from about 30 GWd/t: such a behavior legitimates and makes it possible to fit the coefficient matrices in case of polynomial cross sections, because the use of the average cross section in the determination of the fitting laws does not entails a diverging unfitted fraction as the burnup builds up.

Besides performances and goodness of results, one final aspect is worth mentioning: as Fig. 7.26 shows in the active height for a set of energy groups, the axial gradients of cross sections produced by isotope depletion are generally quite smooth, but can locally be very steep, in particular near the reflector. This can be seen in Fig. 7.27a, wherein the maximum gradients among groups and 2D regions are plotted for each axial floor: in the first layer above the reflector the total cross section undergoes an increase of 128.6%. The values remain remarkable in the fifth axial layer, as well as in Fig. 7.27b, for the same two floors, where radially integrated cross sections are considered. On the other hand, a compensation effect among different energy groups appears clearly in the 7.27c plot, where the values refer to the total cross section over all groups: here, the gradients are reduced to a few percentage points. From the point of view of multi-group calculation, however, the only relevant plot of Fig. 7.27 is obviously the first one, as it contains the detailed information used by the

solver. In this respect, the measured gradients impose, to any referential SC approach, the adoption of an extremely refined axial mesh: for instance, while simulating the last step of the discussed depletion sequence, a reference solver would require layers of thickness smaller than 1 mm if a precision of 10^{-2} was targeted for the cross sections profiles in the reflector proximity. This would clearly be computationally unbearable. Instead, one should consider higher-order implementations involving more unknowns but fewer computational regions, such as the one presented in this thesis.

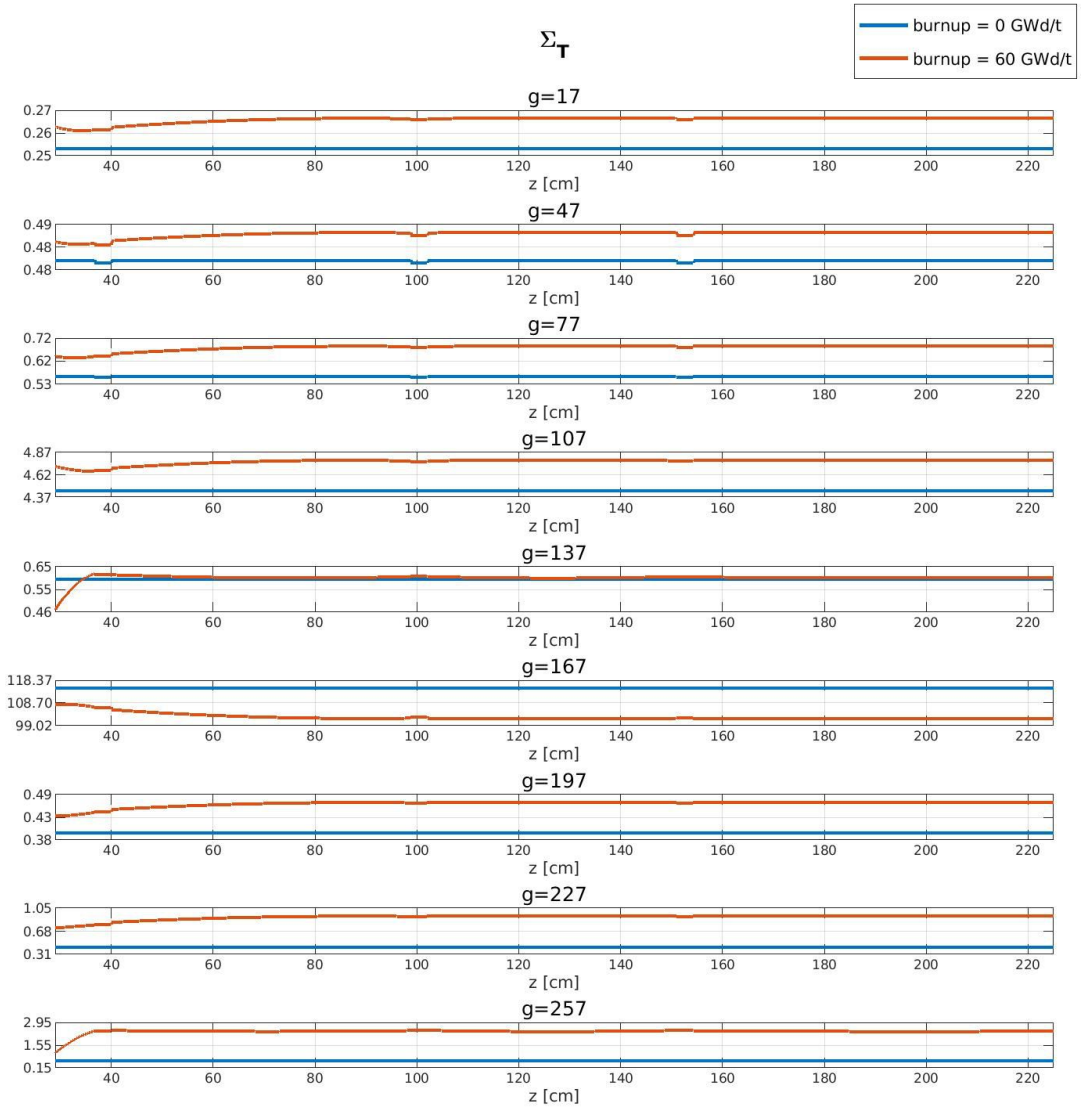
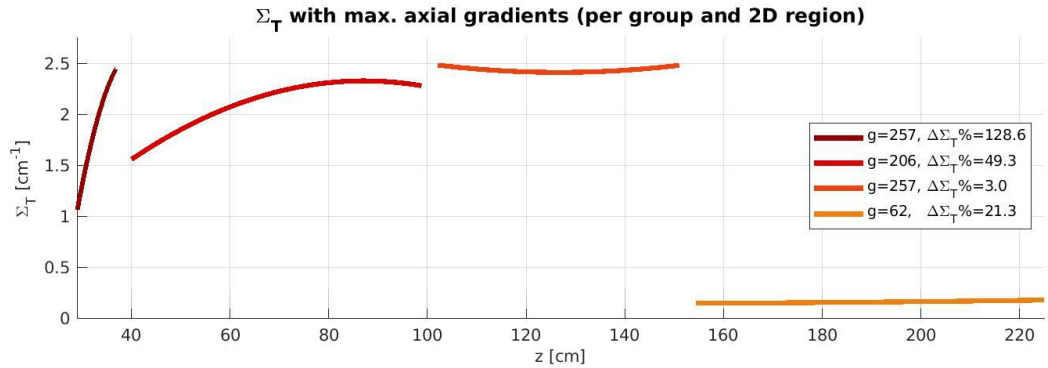
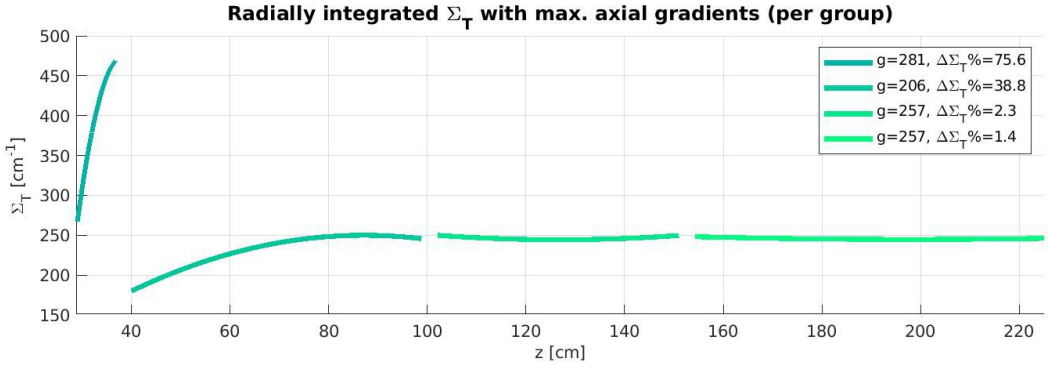


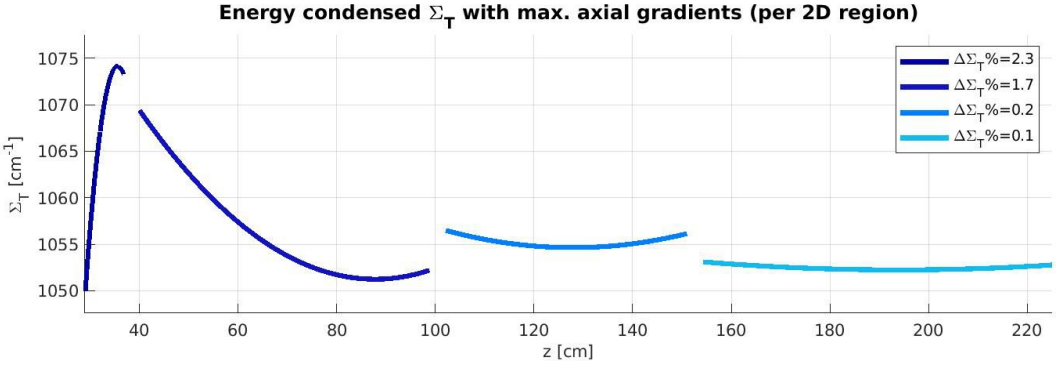
Figure 7.26: BOC - end of depletion comparison of total macroscopic cross section profiles over the active height, for a set of energy groups. The corresponding 2D region is the one presenting the maximum axial gradient of the system, occurring in group 257. The set also includes the group of the lowest-energy resonance of U-238, $g = 167$.



(a)



(b)



(c)

Figure 7.27: For each polynomial layer, the figures show the maximum axial gradients of the total macroscopic cross section among all 2D regions and energy groups (a), integrated over the radial plane, among all groups (b) and energy-condensed, among all 2D regions (c). All plots refer to 60 GWd/t.

Chapter 8

Conclusions and perspectives

In continuity with the previous thesis work carried out on TDT [11], the axial polynomial expansion has been extended to macroscopic cross sections, adopting the same polynomial basis employed for the neutron flux. The scope was to allow the application of this higher-order method to depleted systems, with the hope to benefit from the lower number of regions required even at high burnup levels.

The transmission and balance equations required by MOC solvers to achieve the inner-iteration convergence have been adapted to the new spatial description of cross sections. The implementation concerned their free-iteration versions as well as their DP_N acceleration counterparts; the logic of maximum vectorization has been followed for the evaluation of the transport coefficients, used during the transport sweep, and of the acceleration matrices, with clear benefits on the computing times.

The appearance of integrals that cannot be expressed in closed-form, in the transmission equation and in the DP_N coefficients, as a consequence of the cross sections polynomial development, has been dealt with resorting to an adaptive technique based on Gauss-Legendre quadrature: for all energy groups and every chord contained in polynomial regions, the required quadrature orders are iteratively determined and stored into suitably defined arrays. The Chords Classification Method has been exploited to optimize the orders storage by differentiating the array allocation depending on the chord class.

The non-linear least-square fitting technique, introduced during the previous thesis to limit, if necessary, the memory load of the acceleration matrices, has also been adapted to the new hypotheses: for each energy group, per-region average values of the involved cross sections are used; in this way, nothing else had to be modified of the former implementation, valid for the constant cross sections case.

The new approach has been applied to different PWR subsystems at BOC conditions, so as to test the performances, which proved to be more accurate and faster than the traditional SC. The memory demand, quite comparable in all simulations that have been carried out, became favorable to the higher-order approach in the largest system considered, a full 17x17 assembly. For the latter case, a whole depletion study has been performed, following a strategy that adopts a mesh for the flux calculation and a corresponding Gaussian mesh for depletion and self-shielding. The results of these sequence of calculations, which constitute the real testing ground for the polynomial cross sections method, proved its decisive advantage with respect to the SC approximation, in terms of memory and, above all, computing

time. Moreover, the fraction of DP_N matrices elements that could not be fitted has been shown to undergo a very slight and saturating increase, which legitimates the simple adaptation made. As a final point, even though the cross sections axial gradients, arising as the burnup increased, appeared quite limited on average, some strong and localized effects have been measured, particularly in the proximity of the reflector: this implies the insufficiency, for reference calculations, of a piece-wise constant description of nuclear properties when dealing with high-burnup cases, unless an extremely fine mesh is used.

In the present work the validation comparison with the TRIPOLI-4® reference, involving the k_{eff} eigenvalue and the macroscopic absorption and fission rates, has only concerned the zero-burnup calculations. The higher-burnup results have just been verified by comparing the new and the traditional approach. In the future, it may be convenient to perform all the Monte Carlo simulations corresponding to every depletion step, while using the same depletion solver. Based on the detected axial gradients, careful attention must be paid to choosing the proper reference axial mesh: the convergence study of such mesh throughout the depletion actually represented an untenable amount of work for the available time.

This thesis work has ultimately made the simulation of axially extruded depleted systems much more affordable, taking a further small step towards the feasibility of a one-step full-core calculation. In this respect, one may be interested in applying the polynomial method to study the depletion of larger systems, such as clusters of assemblies. However, unless a compute node with several hundred GB of RAM available, TDT could not be used for that, because it currently relies only on the OpenMP parallel programming. One might think, therefore, to make the solver compatible with MPI, which would allow employing more compute nodes at the same time. Of course, the transport sweep and, in particular, the load balancing algorithm would need be deeply rethought.

One last conceivable development regards the polynomial development itself: while, for the time being, the expansion has been made exclusively along the extruded direction, the potential benefit of a radial expansion on polynomial or exponential functions should be carefully investigated.

Part IV

Annexes

A. Prony's method for transmission integrals

Before developing the adaptive Gauss-Legendre quadrature discussed in section 5.3, an attempt was made to convert the $s^j\, e^{\tau(s)-\tau(l)}$ integrands of Eq. (5.7) into functions having closed-form primitives, by means of the method of Prony [57, 67]. Given a function $f(x)$, this method describes a procedure to approximate it as

$$f(x) \approx \sum_{p=1}^{n} C_p\, e^{b_p x} = \sum_{p=1}^{n} C_p\, \mu_p^x, \tag{A.1}$$

with $N = 2n$ being the number of available sampling points: $x_k = 0, 1, \ldots, N-1$. Here the dicussion considers only an even value of N, but this is not mandatory [67]. Using the notation $f_0 = f(x_k = 0)$, $f_1 = f(x_k = 1)$ and so on, one obtains the following $N \times n$ system:

$$\begin{pmatrix} 1 & 1 & \ldots & 1 \\ \mu_1 & \mu_2 & \ldots & \mu_n \\ \mu_1^2 & \mu_2^2 & \ldots & \mu_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_1^{N-1} & \mu_2^{N-1} & \ldots & \mu_n^{N-1} \end{pmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}. \tag{A.2}$$

The unknowns are given by the n C's and n μ's and the difficulty is due to the N equations being nonlinear in μ. To circumvent this problem the μ's are chosen as the roots of the equation

$$\mu^n + \alpha_1\, \mu^{n-1} + \alpha_2\, \mu^{n-2} + \cdots + \alpha_{n-1}\, \mu + \alpha_n = 0, \tag{A.3}$$

where the α coefficients are solutions of system

$$\begin{pmatrix} f_{n-1} & f_{n-2} & \ldots & f_0 \\ f_n & f_{n-1} & \ldots & f_1 \\ \vdots & \vdots & \ddots & \vdots \\ f_{N-2} & f_{N-3} & \ldots & f_{n-1} \end{pmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = - \begin{bmatrix} f_n \\ f_{n+1} \\ \vdots \\ f_{N-1} \end{bmatrix}. \tag{A.4}$$

The p^{th} equation of the set above is obtained multiplying $n+1$ successive equations of system (A.2), starting from the p^{th} one, by $\alpha_n,\ \alpha_{n-1},\ \ldots,\ \alpha_1$ and 1, respectively, and then adding up the results.

In case of complex μ roots, for each complex conjugate pair $\{\mu_p, \mu_{p+1}\}$ one would consider

$$\mu_{p,p+1} = \mu_R \pm i\, \mu_I = e^{b_R \pm i\, b_I} \tag{A.5}$$

and modify the left-hand side of system (A.2) consequently:

$$\begin{pmatrix} & 1 & 1 & \\ & \mu_p & \mu_{p+1} & \\ \dots & \mu_p^2 & \mu_{p+1}^2 & \dots \\ & \vdots & \vdots & \\ & \mu_p^{N-1} & \mu_{p+1}^{N-1} & \end{pmatrix} \begin{bmatrix} \vdots \\ C_p \\ C_{p+1} \\ \vdots \end{bmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} 1 & 1 \\ e^{b_R}\, cos(b_I) & e^{b_R}\, sin(b_I) \\ \dots \quad e^{2b_R}\, cos(2b_I) & e^{2b_R}\, sin(2b_I) \quad \dots \\ \vdots & \vdots \\ e^{(N-1)b_R}\, cos[(N-1)b_I] & e^{(N-1)b_R}\, sin[(N-1)b_I] \end{pmatrix} \begin{bmatrix} \vdots \\ C_p^* \\ C_{p+1}^* \\ \vdots \end{bmatrix}, \tag{A.6}$$

where

$$C_p^* = \frac{C_p - i\, C_{p+1}}{2}, \qquad C_{p+1}^* = \frac{C_p + i\, C_{p+1}}{2}. \tag{A.7}$$

The corresponding terms of Eq. (A.1) would then be adjusted through the equality

$$C_p\, \mu_p^x + C_{p+1}\, \mu_{p+1}^x = C_p^*\, e^{b_R\, x} cos(b_I\, x) + C_{p+1}^*\, e^{b_R\, x} sin(b_I\, x). \tag{A.8}$$

To summarize, Prony's method consists of four steps:

1. solve system (A.4) to find α's;
2. find μ's, solutions of (A.3);
3. in case of complex roots, convert μ's into e^b's and modify system (A.2) accordingly;
4. solve system (A.2) to obtain C's.

Approximation of transmission integrals

The procedure is therefore applied to the exponential part of the integrand: by sampling $e^{\tau(s)-\tau(l)}$ at points $s(x_k) = \frac{l}{N}(1 + x_k)$, with $k = 1, N$, Prony's method leads to approximate E_j as

$$\begin{aligned} E_j = \int_0^l ds\, s^j\, e^{\tau(s)-\tau(l)} &\approx \frac{l}{N} \sum_{p=1}^{n} C_p \int_{-1}^{N-1} dx\, s(x)^j\, \mu_p^x \\ &= \left(\frac{l}{N}\right)^{1+j} \sum_{p=1}^{n} C_p \int_{-1}^{N-1} dx\, (1+x)^j\, \mu_p^x, \end{aligned} \tag{A.9}$$

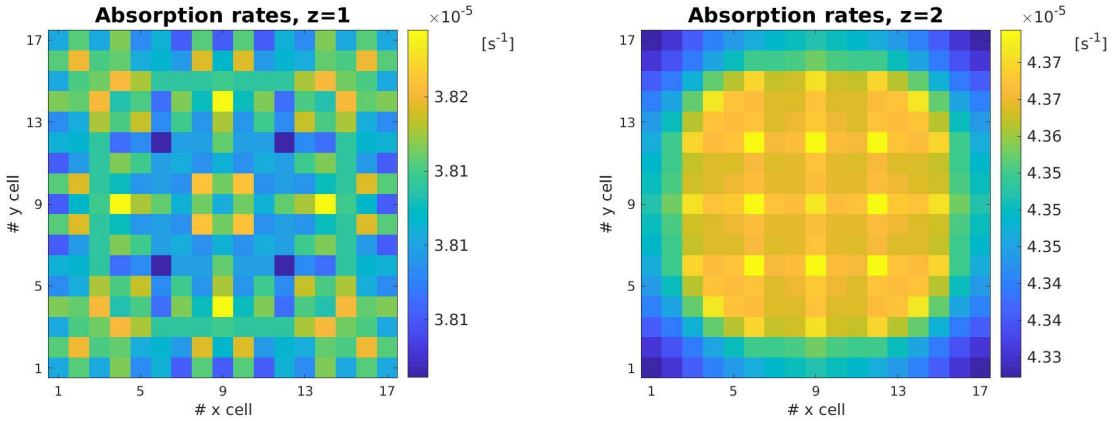
where $\int_{-1}^{N-1} dx\, (1+x)^j\, \mu_p^x$, for $j = 0, 2N_p$, can be expressed in closed-form.

Besides the steps described above, some precautions were taken to avoid the presence of bad conditioned matrices, like limiting the sample points to a significant range for the integrand ($10^{-6} \div 1$). Moreover, the number of samples was limited to 8, so that Eq. A.3 had a maximum degree of 4 and could therefore be solved by algebraic operations, without

the need of iterative approaches [68, 69]. In this way, the method proved to work well for almost every chord. However, the unsustainable amount of floating-point operations required for each chord, with the associated construction time of transport coefficients, prompted to adopt a more efficient strategy.

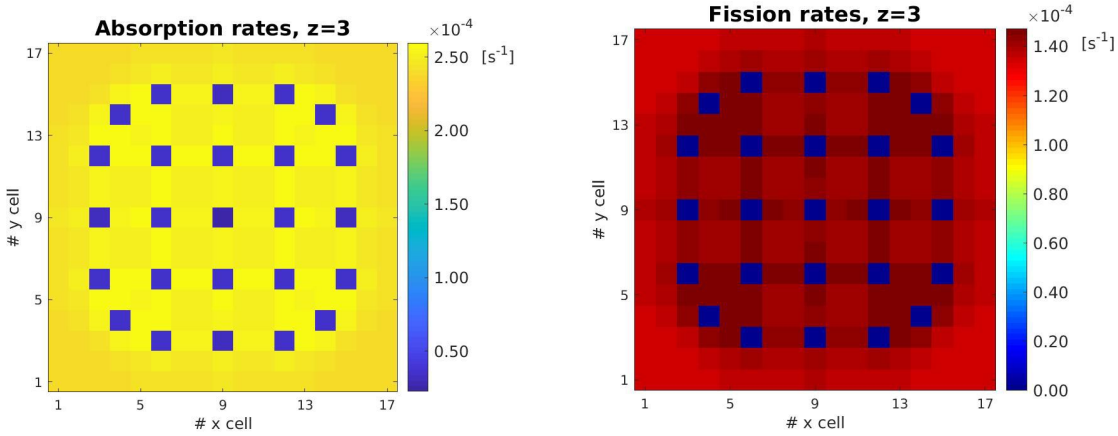
B. Full PWR assembly macro-rates at BOC

Here following are reported the 2D plots of the macroscopic absorption and fission rates of TDT for the FPXM1G1 case, which are compared to the TRIPOLI-4® ones in section 7.2.2.



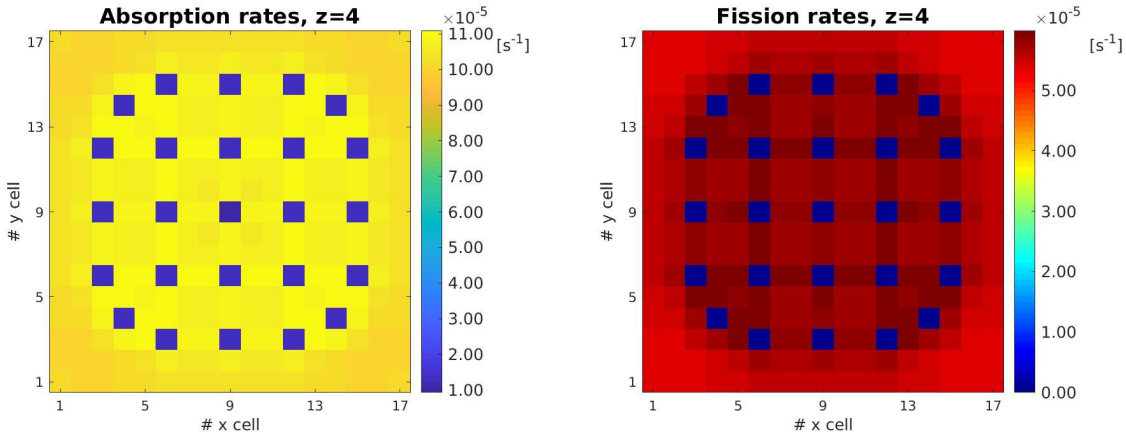
(a) (b)

Figure B.1: Per-cell absorption macro-rates for the 1st and 2nd axial floor.



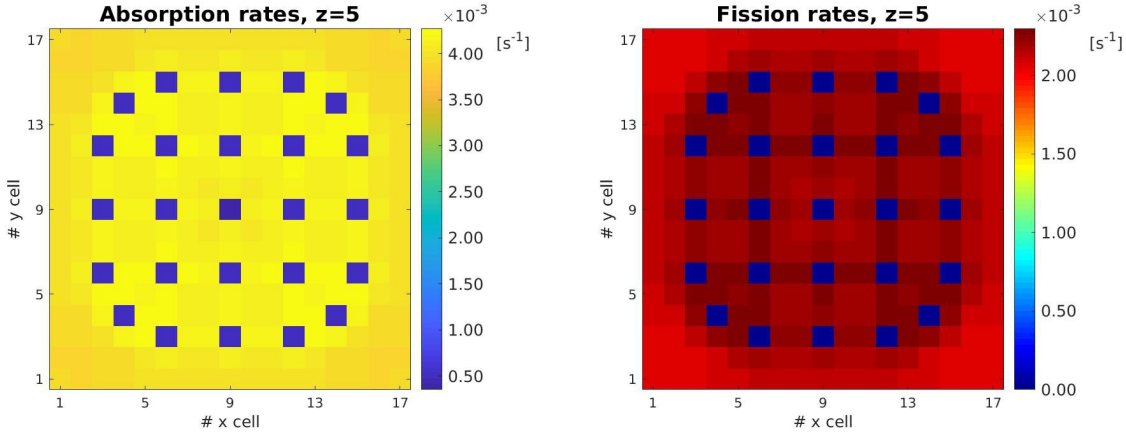
(a) (b)

Figure B.2: Per-cell absorption and fission macro-rates for the 3rd axial floor.



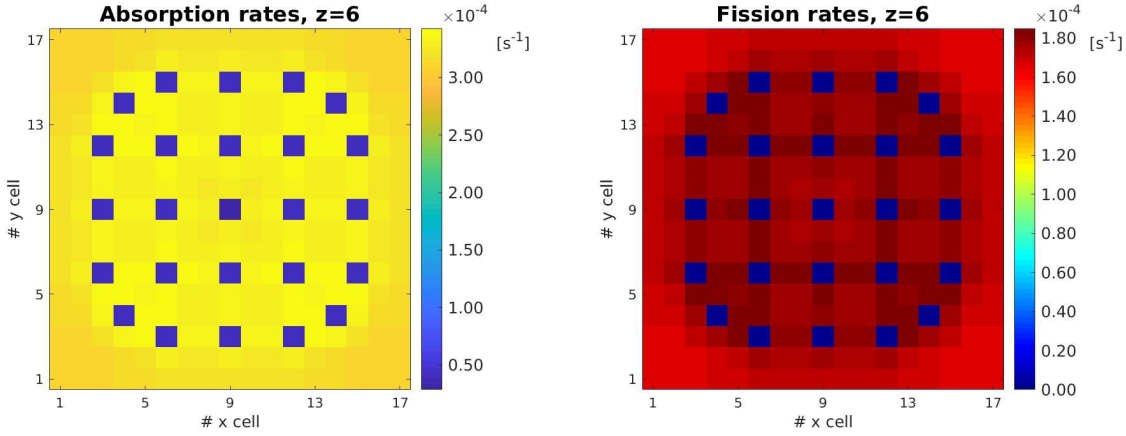
(a) (b)

Figure B.3: Per-cell absorption and fission macro-rates for the 4$^{\text{th}}$ axial floor.



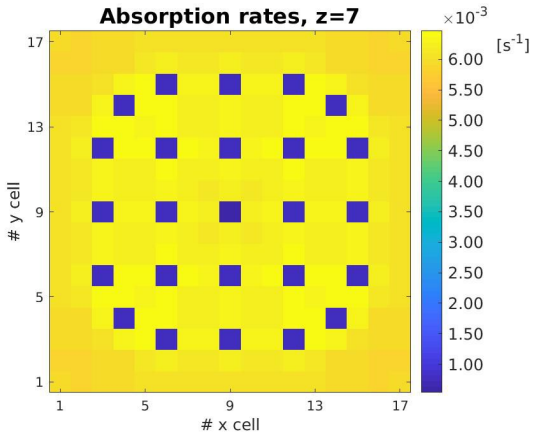
(a) (b)

Figure B.4: Per-cell absorption and fission macro-rates for the 5$^{\text{th}}$ axial floor.

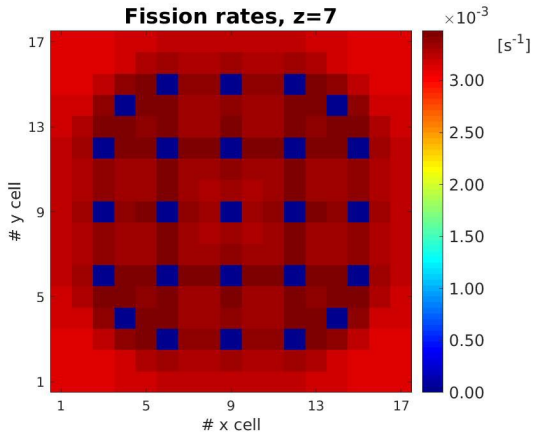


(a) (b)

Figure B.5: Per-cell absorption and fission macro-rates for the 6$^{\text{th}}$ axial floor.

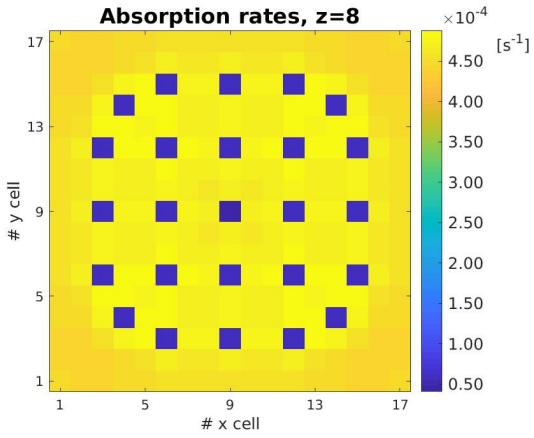


(a)

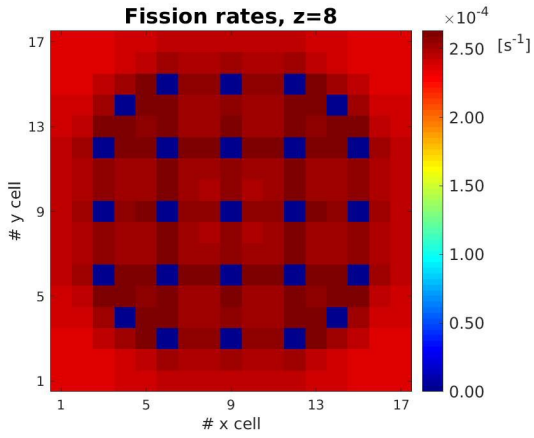


(b)

Figure B.6: Per-cell absorption and fission macro-rates for the 7$^{\text{th}}$ axial floor.

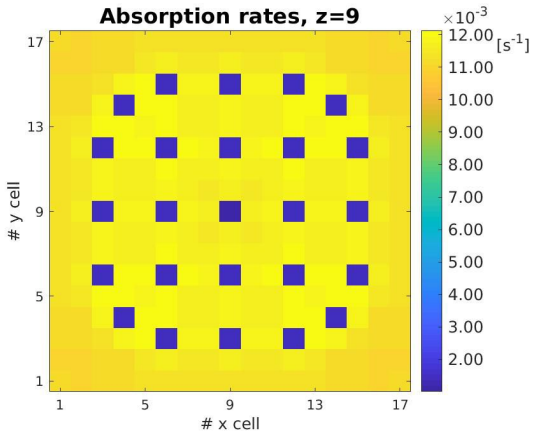


(a)

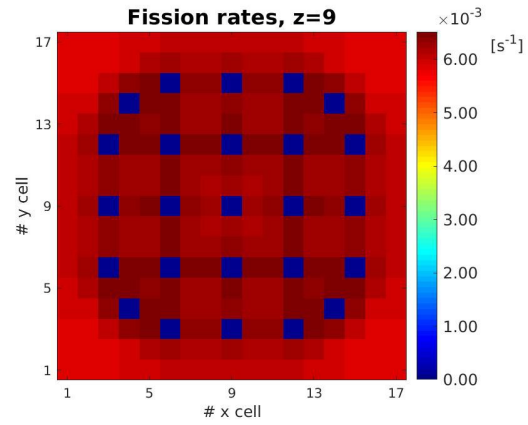


(b)

Figure B.7: Per-cell absorption and fission macro-rates for the 8$^{\text{th}}$ axial floor.



(a)



(b)

Figure B.8: Per-cell absorption and fission macro-rates for the 9$^{\text{th}}$ axial floor.

C. Résumé en français

Dans le cadre des calculs déterministes des réacteurs nucléaires et, en particulier, dans le solveur Two-Three Dimensional Transport (TDT) du code APOLLO3®, le présent travail introduit un développement polynomial des profils spatiaux de sections efficaces macroscopiques, dans le but d'appliquer cette schéma d'ordre supérieur aux systèmes en évolution. Cette thèse représente une amélioration supplémentaire de la méthode des caractéristiques (MOC) 3D implémentée dans TDT, utilisée pour résoudre l'équation de transport des neutrons en régime stationnaire et ainsi déterminer la valeur propre, le facteur de multiplication effectif.

Contexte

Lorsqu'il s'agit de systèmes nucléaires, les simulations neutroniques sont en général subdivisées en deux étapes successives, du fait de la complexité de l'objet d'étude et de la limitation des ressources de calcul. La première phase est responsable de l'évaluation des taux de réaction des neutrons sur un sous-domaine relativement petit et détaillé, qui, répété à l'infini, constitue un réseau le plus représentatif possible du système global. Ensuite, les sections efficaces sont définies sur les plus grandes régions homogénéisées de la géométrie réelle de manière à conserver les taux précédemment calculés. En principe, cette distinction est vouée à être remplacée par une procédure en une seule étape, une fois que les outils numériques le permettront. TDT est, à tous égards, un solveur de réseau 3D, permettant des simulations de première étape plus précises que son implémentation original 2D, mais il aspire également à devenir, dans les années à venir, un solveur à une seule étape. Tout cela grâce à ses développements continus, visant à améliorer les performances du code et ainsi rendre possible l'analyse de cas de plus en plus grands et réalistes.

Tous les solveurs neutroniques déterministes introduisent des représentations approximatives des dépendances sur les variables de l'espace des phases (position, direction et énergie) ; les solveurs de réseau ne font pas exception. Un équivalent numérique du système original est donc considéré, donné par un ensemble de régions de calcul. Ceci permet de distinguer les comportements spatiaux du flux neutronique et des sections efficaces d'une région de ceux d'une autre et éventuellement d'adopter des bases d'expansion différentes. L'approche la plus courante est l'approximation Step-Constant (SC), qui consiste à adopter des valeurs constantes par région. La dépendance angulaire est traitée en introduisant une expansion sur les harmoniques sphériques dans le terme de transfert de l'équation du transport, jusqu'à un certain ordre d'anisotropie, et en reformulant l'ensemble de l'équation de manière à réduire le nombre d'inconnues angulaires à un ensemble de moments du flux. De plus, certaines

méthodes (comme le MOC) considèrent un ensemble discret de directions possibles pour les neutrons (approximation S_N) et adoptent par conséquent une règle de quadrature appropriée pour évaluer les intégrales sur l'angle. Quant à l'énergie, la dépendance continue est typiquement discrétisée en groupes (approximation multigroupe), encore une fois pour avoir un nombre fini d'inconnues ; ceci implique la nécessité de définir avec précision les sections efficaces multigroupes résultantes, afin d'évaluer correctement les taux de réaction par groupe (formalisme d'autoprotection).

Étant donné la possibilité d'être appliquée en principe à n'importe quelle géométrie, la méthode des caractéristiques fait partie des approches dominantes adoptées en neutronique. Elle permet de reformuler l'équation de Boltzmann linéaire aux dérivées partielles comme une équation différentielle ordinaire en considérant son expression le long de la direction de propagation des neutrons. On obtient ainsi une relation intégrale, pour une direction et un groupe énergétique donnés, exprimant le flux angulaire en un point de la ligne caractéristique en fonction de sa valeur en un point précédent et de l'émission des neutrons entre les deux. L'expansion angulaire et l'approximation spatiale choisies sont donc introduites dans le terme d'émission, conduisant à la forme finale de l'équation de transmission qui est utilisée pour le balayage des trajectoires, c'est-à-dire l'évaluation du flux angulaire sortant pour chaque intersection des caractéristiques avec le maillage (dite "corde"). Ensuite, une relation de bilan est nécessaire pour fermer le système et mettre à jour les moments angulaires spatialement constants des densités d'émission (et, éventuellement, leurs coefficients spatiaux d'ordre supérieur) pour le balayage successif. Cette procédure constitue le niveau itératif le plus interne de la méthode des puissances, qui est appliquée pour déterminer la plus grande valeur propre du système, k_{eff} : à partir du groupe énergétique le plus élevé et en procédant selon une logique de Gauss-Seidel, le MOC est chargé d'atteindre la convergence au sein de chaque groupe en appliquant à plusieurs reprises les relations de transmission et de bilan. Lorsque le critère est satisfait dans tous les groupes, une itération dite externe se termine et la valeur propre est mise à jour au moyen des intégrales de fission la plus récente et de la précédente.

Afin d'améliorer la représentativité des résultats de TDT, l'applicabilité du solveur a d'abord été étendue aux géométries 3D extrudées axialement, caractérisées par une section 2D constante pour chaque portion axiale différente présente dans le système. Bien qu'il s'agisse d'un énorme pas en avant, un inconvénient du MOC 3D était le coût de calcul important associé au balayage des trajectoires sur un très grand nombre de régions. Ce fait était une conséquence directe de l'approximation spatiale SC utilisée pour les moments du flux angulaire et pour les sections efficaces, nécessitant un maillage spatial assez raffiné. Par conséquent, pour améliorer les performances, un développement polynomial a été introduit qui décrit les profils axiaux des moments du flux. Cette approche d'ordre supérieur a permis de réduire le nombre de régions le long de la direction axiale d'un facteur 10 environ et, avec le même niveau de précision, la mémoire et le temps de calcul de plus de 50%. Cependant, comme les sections efficaces n'étaient pas impliquées dans le traitement polynomial, son application était limitée aux configurations de début de cycle (Beginning Of Cycle - BOC), c'est-à-dire aux systèmes n'ayant pas encore subi l'évolution isotopique et présentant donc des profils axiaux à sections efficaces constantes.

Nouveaux développements

Si on veut appliquer le MOC d'ordre supérieur à des systèmes en évolution, ce qui est l'objet de cette thèse, il suffit d'adopter le développement polynomial également pour les sections efficaces, afin de représenter correctement les gradients axiaux qui apparaissent au fur et à mesure que le burnup augmente. De cette façon, on peut s'attendre à bénéficier des avantages de la méthode polynomiale également à des niveaux de burnup plus élevés.

L'introduction de la description spatiale d'ordre supérieur des sections efficaces a nécessité l'adaptation des équations de transmission et de bilan du MOC. La forme dérivée de la deuxième envisage la décomposition d'un système global, comprenant les équations pour tous les coefficients polynomiaux des moments angulaires du flux, en sous-systèmes associés à l'ordre des harmoniques sphériques. Cela permet de limiter fortement le nombre d'opérations à virgule flottante nécessaires aux inversions de matrices. Quant à l'équation de transmission, la version obtenue contient des intégrales qui n'ont pas de solution de forme fermée dans le cas général, puisque les chemins optiques, qui sont décrits par des polynômes de degré supérieur à 1 si l'ordre de l'expansion est supérieur à 0, apparaissent comme exposants. Le problème est résolu en recourant à une quadrature de Gauss-Legendre dont l'ordre nécessaire est déterminé de manière adaptative et stocké, pour chaque intégrale à calculer, avant de commencer les itérations de puissance. Chaque fois qu'un nouveau groupe d'énergie est considéré, les coefficients de transport correspondants qui apparaissent dans l'équation de transmission sont calculés et stockés pour le balayage. Cette évaluation profite du très grand nombre de cordes ayant leurs extrémités sur deux surfaces verticales du maillage, appelées cordes V : pour un ensemble de cordes V appartenant à la même région et partageant la même projection sur le plan radial, il est en effet possible de vectoriser le calcul des coefficients, entraînant ainsi une réduction significative du temps de calcul.

Lorsque l'approche des sections efficaces polynomiales est utilisée, une stratégie spécifique a été développée pour permettre à TDT de s'interfacer avec le module d'autoprotection et le solveur de depletion, du fait que ces derniers sont actuellement tous deux incompatibles avec des ordres spatiaux supérieurs à 0. Deux types de maillages axiaux sont donc adoptés : un maillage fin SC, utilisé pour l'autoprotection et la depletion, et un polynomial plus grossier, utilisé en TDT pour le calcul du flux. Le maillage SC contient les valeurs autoprotégées des sections efficaces macroscopiques, à partir desquelles les coefficients polynomiaux sur le maillage le plus grossier sont récupérés par conservation des moments et utilisés par le MOC. Ensuite, une fois la convergence atteinte, le flux calculé est converti dans sa version SC sur le maillage fin et donné en entrée au solveur de depletion. En actualisant les concentrations isotopiques, ce dernier fournit de nouvelles valeurs des sections efficaces macroscopiques qui, après la phase d'autoprotection, servent à répéter le cycle. Celle-ci est la procédure développée pour réaliser une étude d'évolution avec la méthode polynomiale de TDT. Afin de traiter un plus petit nombre de régions SC, contenant des propriétés inutilisées lors des itérations de puissance, un maillage SC gaussien est utilisé, ayant les couches axiales correspondant aux abscisses de Gauss-Legendre : cela permet de réduire la charge mémoire tout en garantissant la conservation des moments des sections efficaces.

La méthode des puissances nécessite notoirement un grand nombre d'itérations pour atteindre la convergence ; pour les réduire et rendre les schémas numériques plus abordables,

plusieurs techniques d'accélération ont été développées. La méthode d'ordre supérieur proposée pour les sections efficaces a été accélérée au moyen de l'approche synthétique DPN, déjà implémentée dans TDT, qui exprime le flux angulaire sur les limites des régions en termes de moments spatialement constants (et de coefficients polynomiaux, si la méthode d'ordre supérieur est employée). L'accélération est utilisée pour corriger la dernière itération de transport, à la fois dans la boucle itérative interne et externe. Pour rendre la méthode cohérente avec l'hypothèse de sections efficaces polynomiales, des équations DPN adaptées ont été dérivées de leurs homologues utilisées dans le MOC. Comme pour ceux de transport, l'évaluation des coefficients d'accélération, reposant sur le traçage des caractéristiques, a exploité les propriétés des cordes V : chaque fois que cela était possible, des opérations vectorisées ont été effectuées en cumulant la contribution de chaque corde à ces coefficients. En outre, l'adaptation a également impliqué la technique de fitting non linéaire aux moindres carrés, introduite avec l'expansion polynomiale pour le flux, qui est utile pour réduire la charge mémoire assez importante due au stockage des coefficients.

Résultats, conclusions et perspectives

Les applications de la nouvelle méthode ont concerné plusieurs sous-systèmes de Réacteurs à Eau Pressurisée (REP). Pour chacun d'eux, une comparaison avec l'approche traditionnelle SC et la référence Monte-Carlo (TRIPOLI-4®) a été fournie, relative aux conditions BOC. Même si les gradients axiaux des sections efficaces ne sont pas présents, de telles configurations permettent de prédire le coût de calcul d'une étude d'évolution complète réalisée avec les routines mises en œuvre. Pour les deux systèmes plus petits les efficacités d'accélération et de vectorisation ont été montrées en termes de réduction du temps de calcul. Concernant les ordres de quadrature requis pour les intégrales de transmission, leurs valeurs en fonction de la tolérance choisie, du type d'intégral et du burnup ont été obtenues et distinguées selon le type de cordes présentes dans le système ; une série de graphiques montre les résultats de cette étude.

Pour le plus grand des systèmes considérés, un assemblage REP entier, une validation à la fois radiale et axiale des taux de réaction en conditions BOC a été faite par comparaison avec TRIPOLI-4®, et une étude de l'évolution a été conduite jusqu'à $60\,GWd/t$, montrant l'avantage décisif de la méthode polynomiale en termes de mémoire et de temps de calcul. Ceci est d'autant plus vrai que la comparaison en conditions BOC avec la référence Monte-Carlo a mis en évidence la bien plus grande précision de l'approximation d'ordre supérieur par rapport à celle SC, malgré que cette dernière adopte une grille axiale avec environ 10 fois plus de mailles. La possibilité d'utiliser la technique de fitting non linéaire des coefficients d'accélération également en cas de sections polynomiales a été démontrée en considérant l'évolution, à mesure que le burnup augmente, de la fraction d'éléments pour lesquels l'erreur de fitting est hors tolérance : les résultats ont montré que, après une légère augmentation, cette fraction tend rapidement à se saturer. Par ailleurs, les gradients axiaux des sections efficaces obtenus ont mis en évidence un aspect supplémentaire : compte ténu de la valeur très remarquable qu'ils peuvent atteindre localement, pour certains groupes énergétiques, tout solveur de référence doit prêter une attention extrême au maillage axial adopté aux niveaux élevés de burnup.

Grâce à ce travail, une avancée significative a été franchie pour rendre plus abordables les simulations neutroniques de systèmes 3D extrudés axialement en évolution. Parmi les développements futurs, des applications à cas d'évolution plus grands pourraient être envisagées, parmi lesquels, notamment, des grappes d'assemblages. Cela nécessitera une grande disponibilité de mémoire sur le nœud utilisé et/ou des optimisations plus poussées dans la méthode. Dans cette dernière perspective, la possibilité d'étendre la description d'ordre supérieur au plan radial et d'ajouter un troisième degré de programmation parallèle dans TDT, qui actuellement n'est pas compatible avec MPI, sera examinée en détail.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] IEA, World gross electricity production, by source, 2018, IEA, Paris https://www.iea.org/data-and-statistics/charts.

[2] D. Sciannandrone, "Acceleration and higher order schemes of a characteristic solver for the solution of the neutron transport equation in 3D axial geometries," Ph.D. dissertation, Université Paris-Sud (2015).

[3] D. Sciannandrone, S. Santandrea, R. Sanchez, L. Lei-Mao, J. Vidal, J. Palau, P. Archier, "Coupled fine-group three-dimensional flux calculation and subgroups method for a FBR hexagonal assembly with the APOLLO3® core physics analysis code," *Mathematics and Computations, Supercomputing in Nuclear Applications and Monte Carlo International Conference, M&C+SNA+MC 2015*, Nashville, Tennessee, 19-23 April 2015.

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

[5] S. Santandrea, D. Sciannandrone, R. Sanchez, L. Mao, L. Graziano, "A Neutron Transport Characteristics Method for 3D Axially Extruded Geometries Coupled with a Fine Group Self-Shielding Environment," *Nuclear Science and Engineering*, **186**, pp. 239-276 (2017).

[6] B. Kochunas, T.J. Dovvnar, Z. Liu, "Parallel 3D Method of Characteristics in MPACT," *Proceedings of the 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering - M&C 2013*, Sun Valley, ID, USA, 5-9 May 2013.

[7] G. Gunow, B. Forget, K.S. Smith, "Full core 3D simulation of the BEAVRS benchmark with OpenMOC," *Annals of Nuclear Energy*, **134**, pp. 299-304 (2019).

[8] G. Giudicelli, K. Smith, B. Forget, "Generalized Equivalence Theory Used with Spatially Linear Sources in the Method of Characteristics for Neutron Transport," *Nuclear Science and Engineering*, **194**, pp. 1044-1055 (2020).

[9] J.R. Tramm, K.S. Smith, B. Forget, A.R. Siegel, "ARRC: A random ray neutron transport code for nuclear reactor simulation," *Annals of Nuclear Energy*, **112**, pp. 693-714 (2018).

[10] D.R. Gaston, B. Forget, K.S. Smith, L.H. Harbour, G.K. Ridley, G. Giudicelli, "Method of Characteristics for 3D, Full-Core Neutron Transport on Unstructured Mesh," *Nuclear Technology*, **207**, pp. 931-953 (2021).

[11] L. Graziano, "An axial polynomial expansion and acceleration of the characteristics method for the solution of the Neutron Transport Equation," Ph.D. dissertation, Université Paris-Sud (2018).

[12] S. Santandrea, L. Graziano, D. Sciannandrone, "Accelerated polynomial axial expansions for full 3D neutron transport MOC in the APOLLO3® code system as applied to the ASTRID fast breeder reactor," *Annals of Nuclear Energy*, **113**, pp. 194-236 (2017).

[13] D. Sciannandrone, L. Graziano, "Memory footprint reduction of a multi-group DP_N transport operator for 3D MOC acceleration," *Proceedings of the 2019 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering - M&C 2019*, Portland, Oregon, USA, 25-29 August 2019, Vol. 4.

[14] B. Fontaine, N. Devictor, P. Le Coz, A. Zaetta, D. Verwaerde, J.-M. Hamy, "The french R&D on SFR core design and ASTRID project," In: Global. Makuhari, Japan (2011).

[15] OpenStax University Physics, *University Physics Volume 3*. OpenStax CNX Jul 16. 2019 http://cnx.org/contents/af275420-6050-4707-995c-57b9cc13c358@12.2.

[16] Nuclear Energy Agency, *Joint Evaluated Fission and Fusion (JEFF) Nuclear Data Library*, http://www.oecd-nea.org/dbdata/jeff/

[17] G.I. Bell, S. Glasstone, *Nuclear Reactor Theory*, Van Nostrand Reinhold, New York USA (1970).

[18] M. Coste-Delclaux, "Modélisation du phénomène d'autoprotection dans le code de transport multigroupe APOLLO2, Rapport CEA, ISSN 0429-3460," CEA Saclay, Direction des systèmes d'information (2006).

[19] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, Society for Industrial and Applied Mathematics, USA (2011).

[20] S. Nakamura, *Computational Methods in Engineering and Science with applications to Fluid Dynamics and Nuclear Systems*, John Wiley & Sons Inc, USA (1977).

[21] J.H. Won, N.Z. Cho, "Discrete ordinates method-like transport computation with equivalent group condensation and angle-collapsing for local/global iteration," *Annals of Nuclear Energy*, **38**, pp. 846-852 (2011).

[22] Members of the Cross Sections Evaluation Working Group, *ENDF-6 Formats Manual: Data Formats and Procedures for the Evaluated Nuclear Data Files ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII*, National Nuclear Data Center - Brookhaven National Laboratory, Upton, NY, USA (2018).

[23] R.J. Labauve, M.E. Battat, D.R. Harris, M.M. Klein, R.E. Seamon, C.R. Weisbin, P.G. Young, *PENDF - a library of nuclear data for Monte Carlo calculations derived from data in the ENDF/B format*, Los Alamos Scientific Laboratory, NM, USA (1974).

[24] R.E. Macfarlane, D.W. Muir, D.C. George, "NJOY99.0: Code System for Producing Pointwise and Multigroup Neutron and Photon Cross Section from ENDF/B Data," PSR-480/NJOY99.0 - RSICC Peripheral Shielding Routine Collection - Contributed by Los Alamos National Laboratory (2000).

[25] M. Coste-Delclaux, C. Diop, A. Nicolas, B. Bonin, *La neutronique, une monographie de la Direction de l'énergie nucléaire*, CEA Saclay and Groupe Moniteur, Paris, France (2013).

[26] P. Leconte, J.-F. Vidal, D. Bernard, A. Santamarina, R. Eschbach, J.-P. Hudelot, "Qualification of the APOLLO2.8 code package for the calculation of the fuel inventory and reactivity loss of UOx spent fuels in BWRs," *Annals of Nuclear Energy*, **36**, pp. 362-367 (2009).

[27] R. Sanchez, I. Zmijarevic, M. Coste-Delclaux, E. Masiello, S. Santandrea, E. Martinolli, L. Villate, N. Schwartz, N. Guler, "APOLLO2 YEAR 2010," *Nuclear Engineering and Technology*, **42**, pp.474-499 (2010).

[28] J. Cetnar, "General solution of Bateman equations for nuclear transmutations," *Annals of Nuclear Energy*, **33**, pp. 640-645 (2006).

[29] S. Lahaye, A. Tsilanizara, P. Bellier, T. Bittar, "Implementation of a CRAM solver in MENDEL Depletion Code System," *Proceedings of M&C 2017*, Jeju, Korea, 16-20 April 2017, Vol. 1.

[30] C. Moler, C. Van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later," *SIAM review*, **45**, pp. 3-49 (2003).

[31] M. Pusa, "Higher-Order Chebyshev Rational Approximation Method and Application to Burnup Equations," *Nuclear Science and Engineering*, **182**, pp. 297–318 (2016).

[32] A. Isotalo, "Computational Methods for Burnup Calculations with Monte Carlo Neutronics," Doctoral dissertation, Aalto University (2013).

[33] J.R. Askew, "A characteristics formulation of the neutron transport equation in complicated geometries," Atomic Energy Authority - Reactor Group, Winfrith, UK (1972).

[34] M. Halsall, "CACTUS, a characteristics solution to the neutron transport equations in complicated geometries," UKAEA Atomic Energy Establishment, Winfrith, UK (1980).

[35] I.R. Suslov, "Solution of transport in 2- and 3-dimensional irregular geometry by the method of characteristics," *Proceedings of Mathematical methods and supercomputing in nuclear applications*, Vol. 1, p. 878, Karlsruhe, Germany (1993).

[36] D. Knott, M. Edenius, "Validation of the CASMO-4 Transport Solution," *Proceedings of Mathematical methods and supercomputing in nuclear applications*, Vol. 2, p. 822, Karlsruhe, Germany (1993).

[37] N.Z. Cho, S.G. Hong, "CRX: a transport theory code for cell and assembly calculations based on characteristic method," *Proceedings of International conference on the physics of reactors PHYSOR96*, Vol. 1, p. 615, Taejeon, Japan (1996).

[38] W. Boyd, S. Shaner, L. Li, B. Forget, K.S. Smith, "The OpenMOC method of characteristics neutral particle transport code," *Annals of Nuclear Energy*, **68**, pp. 43-52 (2014).

[39] R. Ferrer, J. Rhodes, "A Linear Source Approximation Scheme for the Method of Characteristics," *Nuclear Science and Engineering*, **182**, pp. 151-165 (2016).

[40] M. Hursin, B. Collins, Y. Xu, T. Downar, "The Development and Implementation of a One-Dimensional S_n Method in the 2D-1D Integral Transport Solution," *Nuclear Science and Engineering*, **176**, pp. 186-200 (2014).

[41] A. Yamamoto, A. Giho, Y. Kato, T. Endo, "GENESIS : A Three-Dimensional Heterogeneous Transport Solver Based on the Legendre Polynomial Expansion of Angular Flux Method," *Nuclear Science and Engineering*, **186**, pp. 1-22 (2017).

[42] S.G. Stimpson, A.M. Graham, B.S. Collins, "Development of a Coupled Subplane Capability in MPACT," *Proceedings of PHYSOR2020 – International Conference on Physics of Reactors: Transition to a Scalable Nuclear Future*, Cambridge, UK, 29 March - 2 April 2020.

[43] G.S. Lee, N.Z. Cho, "2D/1D fusion method solutions of the three-dimensional transport OECD benchmark problem C5G7 MOX," *Progress in Nuclear Energy*, **48**, pp. 410-423 (2006).

[44] I.R. Suslov, "MCCG3D - 3D discrete-ordinates transport code for unstructured grid. State of the art and future development," *Proceedings of Seminar "Neutronics-96"*, Obninsk, Russia, p. 162 (1998).

[45] G. Wu, R. Roy, "A new characteristics algorithm for 3D transport calculations," *Annals of Nuclear Energy*, **30**, pp. 1-16 (2003).

[46] A. Hébert, "Acceleration of step anf linear discontinuous schemes for the method of characteristics in DRAGON5," *Nuclear Engineering and Technology*, **49**, pp. 1135-1142 (2017).

[47] OpenMP Architecture Review Board, *OpenMP Application Programming Interface, Version 5.1*, November 2020.

[48] Message Passing Interface Forum, *MPI: A Message-Passing Interface Standard, Version 3.1*, High Performance Computing Center Stuttgart (2015).

[49] R. Lenain, E. Masiello, F. Damian, R. Sanchez, "Domain-decomposition method for 2D and 3D transport calculations using hybrid MPI/OpenMP parallelism," *Mathematics and Computations, Supercomputing in Nuclear Applications and Monte Carlo International Conference, M&C+SNA+MC 2015*, Nashville, Tennessee (2015).

[50] E. Masiello, R. Sanchez, I. Zmijarevic, "New Numerical Solution with the Method of Short Characteristics for 2-D Heterogeneous Cartesian Cells in the APOLLO2 Code: Numerical Analysis and Tests," *Nuclear Science and Engineering*, **161**, pp. 257-278 (2009).

[51] A. Chetaine, R. Sanchez, L. Erradi, "The use of the characteristics method to solve the transport equation in unstructured geometries," *Radiation Physics and Chemistry*, **61**, pp. 763-765 (2001).

[52] R. Sanchez, L. Mao, S. Santandrea, "Treatment of boundary conditions in trajectory-based deterministic transport methods," *Nuclear Science and Engineering*, **140**, pp. 23-50 (2002).

[53] A. Kavenoky, J. Lautard, "A Finite Element Depletion Diffusion Calculation Method with Space-Dependent Cross Sections," *Nuclear Science and Engineering*, **64**, pp. 563-575 (1977).

[54] S. Santandrea, P. Bellier, "An unstructured characteristics scheme with a linear expansion of both fluxes and cross sections," In: *Proceedings of the Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)*, Monterey, California, April 15-19, 2007.

[55] P.G. Maginot, J.C. Ragusa, J.E. Morel, "Discontinuous finite element discretizations for the SN neutron transport equation in problems with spatially varying cross sections," *Annals of Nuclear Energy*, **73**, pp. 506-526 (2014).

[56] C. Cohen-Tannoudji, B. Diu, F. Laloë, *Mécanique quantique*, Hermann ed. (1997).

[57] F.B. Hildebrand, *Introduction to numerical analysis, second edition*, Dover Publications Inc., USA (1987).

[58] R. Sanchez, A. Chetaine, "A Synthetic Acceleration for a Two-Dimensional Characteristic Method in Unstructured Meshes," *Nuclear Science and Engineering*, **136**, pp. 122-139 (2000).

[59] I. Suslov, "An Algebraic Collapsing Acceleration Method for the Acceleration of the Inner (Scattering) Iterations in Long Characteristics Transport Theory," *Proceedings of the International Conference on Supercomputing in Nuclear Applications - SNA 2003*, Paris, France, 22-24 September 2003.

[60] S. Santandrea, "An Integral Multidomain DP_N Operator as Acceleration Tool for the Method of Characteristics in Unstructured Meshes," *Nuclear Science and Engineering*, **155**, pp. 223-235 (2007).

[61] K. Madsen, H.B. Nielsen, O. Tingle, *Methods for Non-Linear Least Squares Problems*, 2nd edition (2004).

[62] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares," *Quarterly of Applied Mathematics*, **2**, pp. 164-168 (1944).

[63] L. Graziano, personal communication, June 17, 2020.

[64] E. Brun, F. Damian, C.M. Diop, E. Dumonteil, F.X. Hugot, C. Jouanne, Y.K. Lee, F. Malvagi, A. Mazzolo, O. Petit, J.C. Trama, T. Visonneau, A. Zoia, "TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code," *Annals of Nuclear Energy*, 82, pp. 151-160 (2015).

[65] E. Dumonteil, F. Malvagi, A. Zoia, A. Mazzolo, D. Artusio, C. Dieudonné, C. De Mulatier, "Particle clustering in Monte Carlo criticality simulations," *Annals of Nuclear Energy*, 63, pp. 612-618 (2014).

[66] J.-F. Vidal, P. Blaise, R. Le Tellier, G. Guyot, N. Huot, O. Litaize, A. Santamarina, C. Vaglio-Gaudard, "Analysis of the FLUOLE experiment for the APOLLO2 validation of PWR core reflectors," *Proceedings of PHYSOR2008 – International Conference on the Physics of Reactors "Nuclear Power: A Sustainable Resource"*, Interlaken, Switzerland, 14-19 September 2008.

[67] G.-C.-F.-M. R. de Prony, "Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques, et sur celles de la force expansive de la vapeur de l'alkool, à différentes températures," *Journal de l'École polytechnique*, **1**, pp. 24–76 (1795).

[68] G. Cardano, *Artis Magnae, sive de regulis algebraicis, liber unus*, In J. A. Huguetan, M. A. Ravaud, *Hieronimi Cardani Opera Omnia, vol. IV*, Lyon, France (1663).

[69] N.H. Abel, *Démonstration de l'impossibilité de la résolution algébrique des équations générales qui passent le quatrième degré*, In L. Sylow, S. Lie, *Oeuvres complètes de Niels Henrik Abel: Nouvelle édition*, Cambridge Library Collection - Mathematics, Cambridge University Press, UK (2012).

université PARIS-SACLAY
ÉCOLE DOCTORALE
Physique et ingénierie : électrons, photons, sciences du vivant (EOBE)

Titre: Développement et accélération d'une méthode des caractéristiques 3D incluant une expansion polynomiale axiale des sections efficaces

Mots clés: Théorie du transport des neutrons, Méthode des caractéristiques 3D, Méthodes d'ordre supérieur, Sections efficaces polynomiales

Résumé: Cette thèse introduit un développement polynomial pour les profils axiaux de sections efficaces macroscopiques dans le solveur TDT du code APOLLO3®, utilisant la méthode des caractéristiques (MOC) en 3D, dans le but d'appliquer ce schéma d'ordre supérieur aux systèmes en évolution. La description polynomiale permet une représentation appropriée des gradients axiaux des sections efficaces, qui apparaissent à mesure que le burnup augmente, tout en utilisant un maillage axial plus grossier que celui requis par l'approximation traditionnelle Step-Constant (SC). Les équations du MOC et l'accélération synthétique DP_N ont été adaptées aux nouvelles hypothèses et une stratégie spécifique a été développée pour permettre à TDT de s'interfacer avec le module d'autoprotection et le solveur de depletion. La méthode a été appliquée à différents sous-systèmes de réacteurs à eau pressurisée, en comparant les résultats avec ceux de l'approche SC : l'étude de l'évolution a montré l'avantage décisif de la méthode polynomiale en termes de mémoire et de temps de calcul.

Title: Development and acceleration of a 3D characteristics method including an axial polynomial expansion of cross sections

Keywords: Neutron transport theory, 3D Method of Characteristics, Higher-order methods, Polynomial cross sections

Abstract: This thesis introduces a polynomial expansion for the axial profiles of macroscopic cross sections in TDT, the 3D Method Of Characteristics (MOC) solver of the APOLLO3® code, with the aim to apply this higher-order scheme to depleted systems. The polynomial description allows for a proper representation of the cross sections axial gradients, which appear as the burnup increases, while using a coarser axial mesh than that required by the traditional Step-Constant (SC) approximation. The MOC equations and the DP_N synthetic acceleration have been adapted to the new hypotheses and a specific strategy has been developed to allow TDT to interface with the self-shielding module and the depletion solver. The method has been applied to different pressurized water reactors sub-systems, comparing the results with those of the SC approach: the depletion study has shown the decisive advantage of the polynomial method in terms of memory and computing time.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France