Thesis for the degree of Licentiate of Engineering

Development of a hybrid neutron transport solver

Hirepan Palomares Chavez

Division of Subatomic, High Energy and Plasma Physics
Department of Physics
Chalmers University of Technology
Göteborg, Sweden 2024

Development of a hybrid neutron transport solver

Hirepan Palomares Chavez

Division of subatomic, High Energy and Plasma Physics
Department of Physics
Chalmers university of Technology
SE-412 96 Göteborg Sweden
Telephone: +46(0)31-772 1000

Cover:
Hexagonal fuel assembly of a sodium-cooled fast reactor with simplified fuel pins and a mix of triangular and square mesh.

Printed by Chalmers Digital Print,
Gothenburg, Sweden 2024.

Development of a hybrid neutron transport solver

Hirepan Palomares Chavez
Division of Subatomic, High Energy and Plasma Physics
Department of Physics
Chalmers University of Technology

Abstract

One of the most fundamental analyses of a nuclear reactor consists in solving an eigenvalue problem associated with the neutron transport equation and determining the effective multiplication factor and the distribution of neutrons of the system under static conditions. Two categories of methods can be used for this type of static core calculation: Monte Carlo and deterministic. Monte Carlo methods can reproduce near-real-physics characteristics of the problem at hand but with a high computational cost. On the other hand, deterministic methods lead to quicker but less accurate results after several approximations. In this work, a hybrid computational framework for static core calculations relying on the Interface Current Method (ICM) is developed.

The framework consists of three steps: 1) the whole computational domain is divided into subsystems, 2) a set of collision probabilities inside every sub-system is estimated using Monte Carlo, and 3) the set of collision probabilities is used to estimate the neutron scalar flux and the effective multiplication factor using the ICM. The framework is verified against Monte Carlo reference solutions for three cases based on data from a sodium-cooled fast reactor system, i.e.: 1) a hexagonal fuel-pin cell with simplified and detailed geometry, 2) a hexagonal arrangement of seven fuel-pin cells surrounded by coolant, and 3) a full-size hexagonal fuel assembly.

In the first verification case, for both the simplified and the detailed fuel-pin cell, the framework was tested with and without production scattering cross sections, using 3 different coarse meshes, and considering collision probabilities estimated with different number of neutron histories. Good agreement with the reference solution is obtained for both simplified and detailed fuel-pin cells when using scattering production cross sections. The sensitivity analysis shows that increasing the number of neutron histories allows to minimize the uncertainty of the collision probabilities and thus improves the results. A coarse mesh with a combination of triangular and rectangular coarse nodes is used in the second and third verification cases. A relatively good agreement is obtained in terms of the effective multiplication factor and scalar neutron flux in fast systems.

Keywords

Hybrid neutron transport, Interface current method, Deterministic methods, Monte Carlo, Fast reactors

Acknowledgments

I would like to express my gratitude to my supervisors Assoc. Prof. Paolo Vinai and Prof. Christophe Demaziére at Chalmers University of Technology for their guidance, support, and contributions to this project. Also, I would like to thank my examiner Prof. Tünde Fülöp, Prof. Jan Svensson and Prof. Lena Falk for their help during the project. Big thanks to Prof. Tünde Fülöp for her feedback on the thesis.

This project is funded by The Swedish Research Council under grant agreement No. 2019-03868. The computational resources were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. Calculations were performed at Tetralith - National Supercomputer Centre (NSC), Linköping University and at Kebnekaise - High Performance Computing Center North (HPC2N), Umeå University.

Special thanks to the Paul Scherrer Institut (PSI), Switzerland for sharing the data for the modelling of the Superphénix reactor.

A big thank you to my family and my girlfriend for their love and support. I would like to thank my friends at Chalmers University of Technology for their companionship and enriching conversations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A background on nuclear energy is provided in section 1.1. Fast nuclear reactors are introduced in section 1.2. Aspects of computational reactor physics relevant to this work are presented and discussed in section 1.3. The motivation of the project is presented in section 1.4. Finally, the structure of the thesis is described in section 1.5.

1.1 Background

Nuclear power comprises approximately 10% of global electricity production, making it the second-largest source of carbon-neutral energy after hydro-power. As of the 1st of January of 2023, there were 411 operating nuclear reactors distributed across 32 countries, 27 reactors in suspended operation in Japan and India and 58 more under construction in 18 countries. During 2022, 6 new reactors were connected to the electrical grid with a combined electric power of 7,400 MWe and the construction of 8 reactors with 9,125 MWe started. Given its minimal carbon emissions, nuclear energy stands as a pivotal component in facilitating the transition toward cleaner and more sustainable energy sources [1].

Most operating nuclear power reactors are water-cooled thermal reactors and are predominantly of two types: pressurized water reactors (PWR) and boiling water reactors (BWR). In these systems, neutrons are slowed down, i.e. moderated, with light water to thermal energies (< 1 eV) to induce the fissions in fissile material. An isotope that has a large probability of fissioning when struck by a thermal neutron is called fissile. Examples of fissile isotopes are U^{233}, U^{235}, Pu^{239} and Pu^{241}. In Figure 1.1 the fission cross section for the fissile isotopes mentioned above is depicted as a function of energy. For these isotopes, the fission cross section for energies below 1 eV (which can be considered as an upper limit for the thermal range) is higher than 25 barns, with an exponentially increasing trend towards less energetic regions. Naturally occurring uranium only contains 0.711% of U^{235}, then the nuclear fuel needs to be enriched between 3% and 5% of U^{235} to sustain a chain reaction in thermal reactors. Such an arrangement leads to poor utilization of the uranium resources since most of the uranium ore contains U^{238}. Fast nuclear reactors can make use of the remaining 99% of uranium resources in a more efficient way.

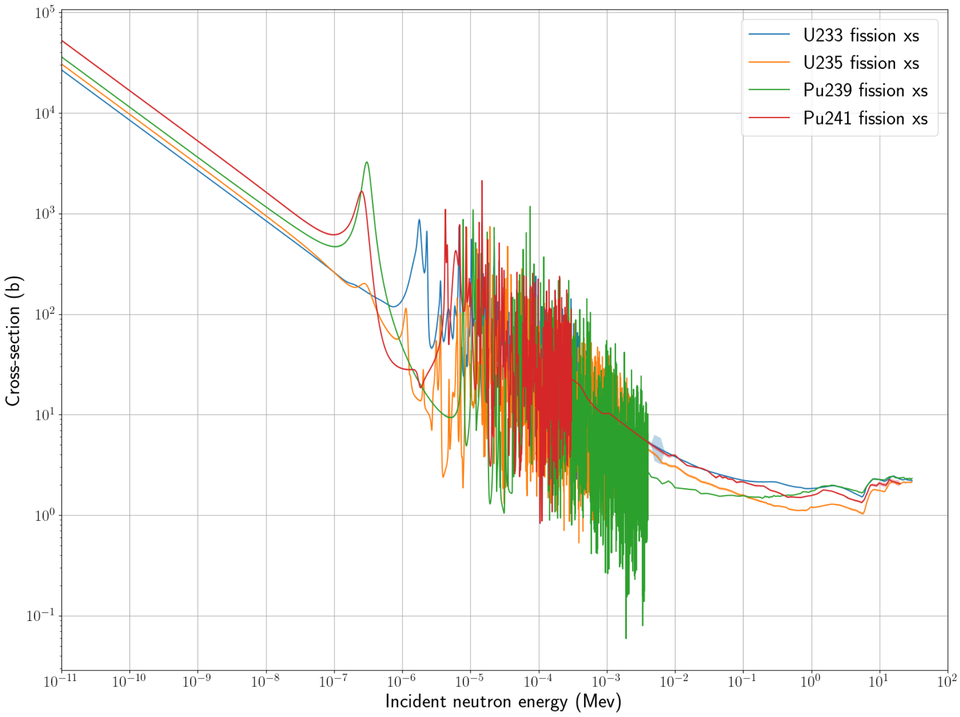


Figure 1.1: Fission cross section as a function of energy for fissile isotopes. Extracted from the JEFF-3.3 library [2], using the JANIS software [3].

1.2 Fast reactors

The first fast reactor ever built was the reactor Clementine in Los Alamos, 1946. After the enthusiasm of Enrico Fermi and Walter Zinn, the fast reactor EBR-1 (Experimental Breeder Reactor), designed by Los Alamos Scientific Laboratory and Argonne National Laboratory, was built in Idaho Falls, 1951, becoming the first-ever nuclear reactor of any type to produce electricity. The power generated by the EBR-1 was 200 kWe, which was enough to provide the needs of the building. In the next three decades 1950s, 1960s, and 1970s, more experimental liquid metal fast breeding reactors (LMFBRs) cooled by sodium or lead were developed in the US, France, UK, Germany, Japan, and the Soviet Union. By the beginning of the 1980s, the first large-scale sodium-cooled fast breeder nuclear power station Superphènix (SPX) was nearing completion with an expected output capacity of 1200 MWe [4]. The SPX reactor achieved criticality for the first time in September 1985, and full power in December 1986 to be later shut down permanently in December 1998.

One of the key components of a fast reactor is the breeding process, where fissile material, such as U^{233}, U^{235}, Pu^{239}, or Pu^{241}, is created from fertile isotopes such as Th^{232}, U^{234}, U^{238}, or Pu^{240}. The two main conversion chains are the conversion of U^{238} into Pu^{239} and the conversion of Th^{232} into U^{233}, being the first one the most efficient. The two chains are as follows:

$$^{238}_{92}\mathrm{U} \xrightarrow[(\mathrm{n},\gamma)]{} {}^{239}_{92}\mathrm{U} \xrightarrow[(23.5\mathrm{m})]{\beta^-} {}^{239}_{93}\mathrm{Np} \xrightarrow[(2.35\mathrm{d})]{\beta^-} {}^{239}_{94}\mathrm{Pu} \tag{1.1}$$

$$^{232}_{90}\mathrm{Th} \xrightarrow[(\mathrm{n},\gamma)]{} {}^{233}_{90}\mathrm{Th} \xrightarrow[(23.4\mathrm{m})]{\beta^-} {}^{233}_{91}\mathrm{Pa} \xrightarrow[(27\mathrm{d})]{\beta^-} {}^{233}_{92}\mathrm{U} \tag{1.2}$$

A characteristic of fertile isotopes is that their probability to fission is very low (< 5 barns) when struck by a neutron with energies in the order of ≤1 eV, see Figure 1.2.

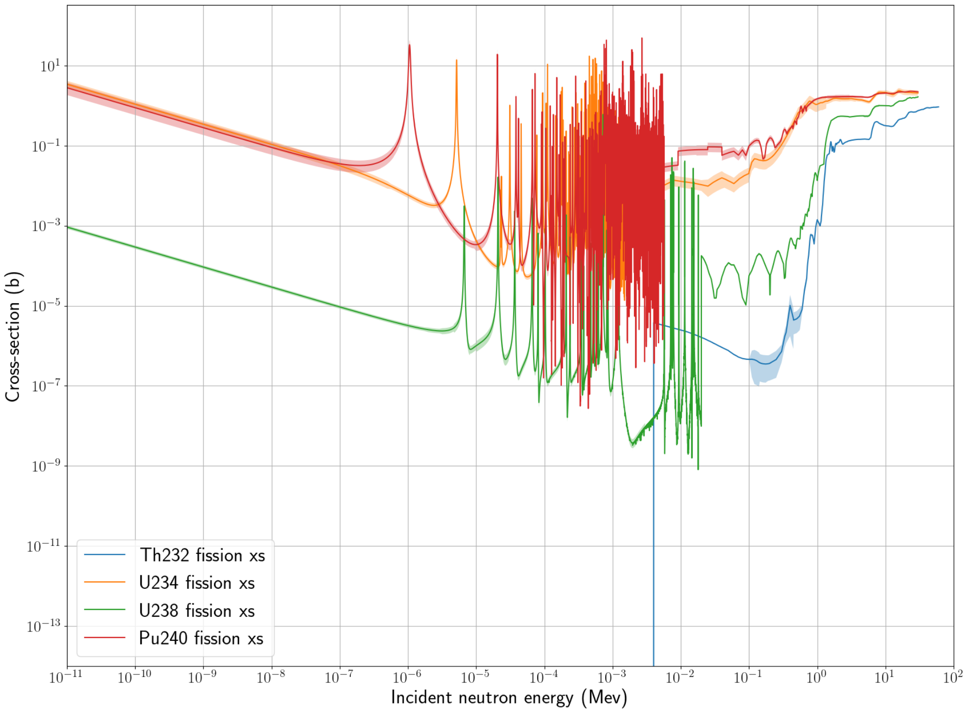


Figure 1.2: Fission cross section as a function of energy for fertile isotopes. Extracted from the JEFF-3.3 library [2], using the JANIS software [3].

For a reactor to be called breeder, the ratio of produced (PF) to consumed (CF) fissile material, also called the breeding ratio (BR), needs to be higher than one, i.e.,:

$$BR = \frac{PF}{CF} > 1 \tag{1.3}$$

The breeding process is influenced by the neutron spectrum of the reactor. Figure 1.3 shows the neutron spectrum for the thermal reactors PWR and Very High-Temperature Reactor (VHTR), and for the fast reactors Lead cooled Fast reactor (LFR) with nitride fuel-type and a Sodium Fast Reactor (SFR) with metal and oxide fuel-type. Fast systems are more suitable for breeding than thermal systems because their neutron spectrum is shifted towards energies larger than 0.1 keV and is not flattened out by neutron thermalization.

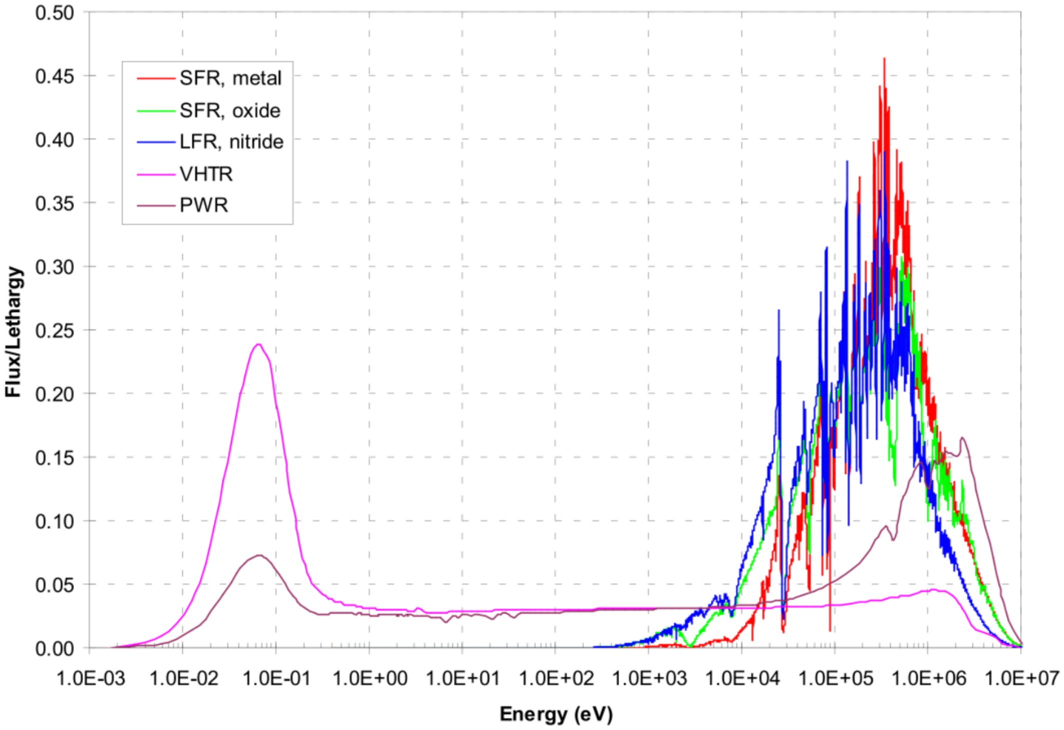


Figure 1.3: Neutron spectrum characteristic of different type of reactors. Taken from Ref. [5].

To sustain the chain reaction and have energy output, the fission reactions are still needed while breeding. Consequently, very high enrichment of fissile isotopes in a fast reactor is required, due to the fast neutron spectrum and the relatively low fission cross sections at high neutron energies. The enrichment can go from 9% to 23% in fissile fraction in the initial fuel mixture. In a nuclear reactor, the amount of energy extracted per unit mass of initial fuel is called burnup, its requirements are higher for a fast system, and this comes directly related to the high fissile fraction. While for a LWR a burnup of 30 MWd/kg can be acceptable, for a fast breeder reactor the requirements for an economically feasible operation are in the order of 100MWd/kg.

Liquid metals and gases are the coolants that comply with the requirements for a fast breeder reactor: 1) no neutron moderation, 2) high heat transfer coefficient for a high power density system, and 3) very low parasitic neutron absorption. The most commonly used are sodium and lead-bismuth eutectic.

The cladding used in a fast breeder reactor differs from the one used in a LWR. The primary requirements are: capable of resisting very high temperatures, good withstanding of neutron fluence, chemical compatibility, and low absorption of neutrons. The 316 stainless steel is a typical choice for the cladding because of its good strength and corrosion performance at high temperatures, and also to the resistance to high neutron fluence characteristic of liquid metal-cooled fast breeder reactor.

The core of a fast system typically includes 3 zones: the fissile core, the blanket, and the shielding. The fissile core is where the majority of the fissile material is placed and the region with highest power density. The blanket, which is where the breeding process takes place, can be internal or external. The internal breeding concept involves mixing fertile material assemblies with fissile material assemblies, while the external consists of having a well-defined blanket section. Then, the reactor core is surrounded by radial and axial shielding that serves as a reflector of neutrons.

A tight fuel arrangement is needed for a fast system, in order to minimize the fissile material loading by mainly reducing the neutron leakage. Thus maximizing the fuel volume fraction is preferred where a hexagonal arrangement is the solution. In contrast, an LWR uses mainly square arrangement to optimize the water-to-fuel ratio which also provides manufacturing advantages. A fuel assembly for a sodium-cooled reactor, also called sub-assembly, consists of a cluster of normally more than 200 fuel pins separated by a spiral wire wrap within a hexagonal channel. A fuel pin generally includes an active region, lower and/or upper axial blanket with an upper and/or lower gas plenum. The dimensions of each zone depend on the design of the reactor.

1.3 Computational reactor physics for fast systems

During the lifetime of a nuclear power plant from the design stages until its final decommissioning, several analyses are needed to ensure safe and efficient operation. One analysis that is important for the reactor core is commonly referred to as neutronics modeling and it leads to the estimation of the neutron multiplication factor and the spatial distribution of the neutron flux in the system. This requires advanced techniques which can be based on a deterministic or a stochastic approach.

According to the deterministic approach, approximations may be introduced to simplify the complexity of the system and/or the modeling of terms in the neutron transport equation, the neutron transport equation may be discretized with respect to the independent variables, i.e., time, energy, position, and solid angle, and then the problem is solved numerically in an iterative manner.

In the probabilistic or Monte Carlo approach, no equation is solved, but the history of a neutron is simulated probabilistically from birth to death where all the variables in the system (energy, angle, and position) are sampled randomly along the life of the neutron. Monte Carlo calculations allow to handle very complex geometries and represent the actual physics of the system. To be able to generate an accurate solution and reduce the uncertainty due to its probabilistic nature, several neutron histories need to be simulated which raises the computational cost.

Hybrid methods have been studied with different objectives, such as variance reduction and the convergence of the fission source in Monte Carlo, as well as the development of Multi-stage Response-function Transport (MRT) methods. The term 'hybrid' was originally coined to incorporate biasing parameters into Monte Carlo simulations, calculated deterministically to reduce uncertainty in source-detector problems, particularly for detectors located far from the source [6]. Efforts to enhance the convergence of the fission source in Monte Carlo include the coarse mesh finite difference (CMFD) method developed by Massachusetts Institute of Technology and the University of Michigan [7], [8]. An alternative, although less common, hybrid methodology is based on a deterministic approach to resolve the energy dependence of the neutron flux in the fast and thermal regions and a Monte Carlo approach for the resonance region. In this case, the solutions are then coupled through the scattering source to ultimately obtain the neutron distribution [9].

The MRT methods, are used to solve neutron transport problems via a two-step procedure. First, the domain is subdivided into multiple subdomains or stages, and a set of coefficients or response functions are pre-calculated with Monte Carlo for each of the subdomains. Second, a linear system of equations is built from the pre-calculated set of coefficients and is solved iteratively in a deterministic fashion to calculate the multiplication factor and the spatial distribution of neutron scalar flux [10]. Previous efforts in the category of MRT methods are those made by Virginia Tech with the code RAPID [11] and Georgia Tech with the software COMET [12]. In RAPID, the fission matrix methodology is used, which involves calculating a set of fission coefficients for a fraction of a square assembly with Monte Carlo, typically through fixed source calculations. This set is then used as a database to solve a linear system of equations for determining the neutron distribution in larger systems. On the other hand, the methodology used in COMET relies on the calculation of a set of response expansion coefficients for a determined set of sub-systems inside a coarse mesh. The coefficients are arranged into a linear system of equations to calculate the effective multiplication factor and the angular neutron flux distribution. Another work developed in Chalmers, which is the basis of this thesis, applies the Interface Current Method (ICM) and was demonstrated over a thermal system [13].

1.4 Motivation and objectives of the project

The current project investigates a hybrid computational framework for static calculations of fast reactor cores. The framework relies on the Interface Current Method and belongs to the category of multi-stage response-function transport. The ICM begins by dividing the domain into a set of subsystems, i.e., coarse mesh. Each subsystem is further refined into a fine mesh. Then, it uses the collision probabilities inside the nodes of the coarse mesh as a set of response functions or coefficients to transmit the neutron information between the fine mesh and the coarse mesh. To calculate the collision probabilities needed in the response matrix method, the use of a Monte Carlo transport code is explored. Given the assumption that the neutron emission rates inside the fine mesh are known, the set of collision probabilities is combined into a linear system of equations and solved iteratively, to calculate the scalar neutron flux in the fine mesh and the neutron multiplication factor.

The hybrid strategy aims to leverage on the computational efficiency of deterministic methods while ensuring a certain level of accuracy and reliability through the probabilistic part.

1.5 Structure of the thesis

This thesis presents a hybrid framework for static calculations of fast reactor core and is structured as follows. Chapter 2 covers the formulation of the interface current method used for the hybrid framework. Chapter 3 describes the design of the hybrid framework, the procedure for the estimation of the collision probabilities via Monte Carlo, and the deterministic algorithm. Chapter 4 discusses the verification and the performance of the hybrid framework using problems obtained from a sodium-cooled fast breeder reactor. Chapter 5 provides conclusions and possible future developments of the framework.

Chapter 2

Theoretical Background

The integro-differential and integral forms of the neutron transport equation are introduced in section 2.1. The interface current method which is the base of the hybrid framework developed in this work is derived in section 2.2.

2.1 Neutron transport equation

2.1.1 Integro-differential formalism

In a nuclear system, under steady-state conditions, the distribution of neutrons with respect to position, solid angle, and energy is given by the static neutron transport equation. The methods reported hereafter follow the derivations of Ref. [14]. Considering a neutron multiplying medium without external sources, the integro-differential form of this equation reads:

$$\mathbf{\Omega} \cdot \nabla\psi(\mathbf{r}, \mathbf{\Omega}, E) + \Sigma_t(\mathbf{r}, E)\psi(\mathbf{r}, \mathbf{\Omega}, E) = q(\mathbf{r}, \mathbf{\Omega}, E) \tag{2.1}$$

with the neutron source defined by:

$$\begin{aligned} q(\mathbf{r}, \mathbf{\Omega}, E) = &\int_{4\pi}\int_0^\infty \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \to \mathbf{\Omega}, E' \to E)\psi(\mathbf{r}, \mathbf{\Omega}', E')dE'd^2\mathbf{\Omega}' \\ &+ \chi(\mathbf{r}, E)\int_{4\pi}\int_0^\infty \nu\Sigma_f(\mathbf{r}, E')\psi(\mathbf{r}, \mathbf{\Omega}', E')dE'd^2\mathbf{\Omega}', \end{aligned} \tag{2.2}$$

where $\psi(\mathbf{r}, \mathbf{\Omega}, E)$ is the angular neutron flux with dependence on position, solid angle, and energy. In Eq. (2.1), the left-hand side represents the total losses of the system and the right-hand side represents the total source of neutrons. The first term of the left-hand-side of the equation is the streaming operator $\mathbf{\Omega} \cdot \nabla\psi(\mathbf{r}, \mathbf{\Omega}, E)$ that accounts for losses of neutrons through a given surface of a volume. The second term $\Sigma_t(\mathbf{r}, E)\psi(\mathbf{r}, \mathbf{\Omega}, E)$ is the total rate at which neutrons are lost due to a collision reaction. In a neutron multiplying system without external neutron sources, the term on the right-hand side includes two contributions: the production due to scattering reactions and the neutrons born from fission. In the case of static problems, the precursors of delayed neutrons are assumed in equilibrium and the total fission production spectrum (χ) takes into account both prompt and delayed neutrons. The first term of the right-hand side of Eq. (2.2):

$$\int_{4\pi}\int_0^\infty \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \to \mathbf{\Omega}, E' \to E)\psi(\mathbf{r}, \mathbf{\Omega}', E')dE'd^2\mathbf{\Omega}',$$

accounts for all the neutrons traveling within all energies E' within all solid angles Ω' that are scattered with speed E and solid angle Ω after the collision. The fission term

$$\chi(\mathbf{r}, E) \int_{4\pi} \int_0^\infty \nu\Sigma_f(\mathbf{r}, E')\psi(\mathbf{r}, \mathbf{\Omega}', E')d^2\mathbf{\Omega}'dE'$$

accounts for the total number of neutrons produced with energy E from fission induced by neutrons traveling with speed E' and having a solid angle Ω' before the collision. Integrating the fission term for all solid angles to remove the angular dependency from the neutron flux gives as a result:

$$\frac{\chi(\mathbf{r}, E)}{4\pi} \int_0^\infty \nu\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E')dE', \tag{2.3}$$

where $\phi(\mathbf{r}, E')$ is the scalar neutron flux. To balance the neutron losses with the production of neutrons, the fission term in the source is simply divided by the multiplication factor k:

$$\begin{aligned} q(\mathbf{r}, \mathbf{\Omega}, E) = &\int_{4\pi} \int_0^\infty \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \to \mathbf{\Omega}, E' \to E)\psi(\mathbf{r}, \mathbf{\Omega}, E')dE'd^2\mathbf{\Omega}' \\ &+ \frac{\chi(\mathbf{r}, E)}{4\pi k} \int_0^\infty \nu\Sigma_f(\mathbf{r}, E')\phi(\mathbf{r}, E')dE' \end{aligned} \tag{2.4}$$

In a critical system where the neutron losses are equal to the production of neutrons, the effective multiplication factor takes a value equal to unity ($k = 1$). In a supercritical system where the production of neutrons is higher than the losses the multiplication factor is greater than unity ($k > 1$), thus balancing the Eq. (2.1) to compensate the excess of neutrons and satisfying the equality. In the opposite case when the system is subcritical ($k < 1$) the equation is balanced in such a manner that the fission neutron production term is increased to compensate for the higher losses.

2.1.2 Multigroup formalism

To solve the neutron transport equation within a deterministic framework, the multigroup approximation is used. Accordingly, all the continuous-energy dependence of the neutrons is handled by using the multigroup approximation where all the continuous-energy dependent variables are discretized splitting the continuous-energy spectrum of the neutrons into a number G of energy bins [14]:

$$[E_{min}; E_{max}] = \bigcup_{g=G}^{1} [E_g; E_{g-1}] \tag{2.5}$$

When introducing the multigroup formalism to the neutron transport equation, the reaction rates and the neutron flux need to be averaged in each of the energy bins. The averaging process is simply done by integrating the angular neutron flux and cross sections in each of the energy intervals $[E_g; E_{g-1}]$ to preserve the reaction rates as follows:

$$\psi_g(\mathbf{r}, \mathbf{\Omega}) = \int_{E_g}^{E_{g-1}} \psi(\mathbf{r}, \mathbf{\Omega}, E)dE \tag{2.6}$$

$$\Sigma_{t,g}(\mathbf{r}) = \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\mathbf{r}, E)\phi_w(E)dE}{\int_{E_g}^{E_{g-1}} \phi_w(E)dE} \tag{2.7}$$

$$\Sigma_{s,g'\to g}(\mathbf{r}, \mathbf{\Omega}' \to \mathbf{\Omega}) = \frac{\int_{E'_g}^{E'_{g-1}} \phi_w(E') \int_{E_g}^{E_{g-1}} \Sigma_s(\mathbf{r}, \mathbf{\Omega}' \to \mathbf{\Omega}, E' \to E)dE'dE}{\int_{E'_g}^{E'_{g-1}} \phi_w(E')dE'} \tag{2.8}$$

$$(\nu\Sigma_f)_{g'}(\mathbf{r}) = \frac{\int_{E'_g}^{E'_{g-1}} \nu\Sigma_f(\mathbf{r}, E')\phi_w(E')dE'}{\int_{E'_g}^{E'_{g-1}} \phi_w(E')dE'} \tag{2.9}$$

$$\phi_{g'}(\mathbf{r}) = \int_{E'_g}^{E'_{g-1}} \phi(\mathbf{r}, E')dE' \tag{2.10}$$

$$\chi_g(\mathbf{r}) = \int_{E_g}^{E_{g-1}} \chi(\mathbf{r}, E)dE \tag{2.11}$$

where $\phi_w(E)$ is a weight function characteristic of the system. Now we can rewrite Eq. (2.1) in its multigroup form:

$$\mathbf{\Omega} \cdot \nabla\psi_g(\mathbf{r}, \mathbf{\Omega}) + \Sigma_{t,g}(\mathbf{r})\psi_g(\mathbf{r}, \mathbf{\Omega}) = q_g(\mathbf{r}, \mathbf{\Omega}) \tag{2.12}$$

with:

$$\begin{aligned} q_g(\mathbf{r}, \mathbf{\Omega}) =& \sum_{g'=1}^{G} \int_{4\pi} \Sigma_{s,g'\to g}(\mathbf{r}, \mathbf{\Omega}' \to \mathbf{\Omega})\psi_{g'}(\mathbf{r}, \mathbf{\Omega}')d^2\mathbf{\Omega}' \\ &+ \frac{\chi_g(\mathbf{r})}{4\pi k} \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'}(\mathbf{r})\phi_{g'}(\mathbf{r}), \end{aligned} \tag{2.13}$$

and

$$g = 1, \ldots, G.$$

2.1.3 Integral formulation

The integral neutron transport equation can be deduced from the integro-differential form. Considering that the gradient $\mathbf{\Omega} \cdot \nabla$ is the directional derivative along the direction of neutron travel between the points B($\mathbf{r}$) and A($\mathbf{r} + u\mathbf{\Omega}$) depicted in Figure 2.1 with u being the traveled distance, the integro-differential form of the neutron transport equation can be rewritten as [15]:

$$\frac{d}{du}\psi_g(\mathbf{r} + u\mathbf{\Omega}, \mathbf{\Omega}) + \Sigma_{t,g}(\mathbf{r} + u\mathbf{\Omega})\psi_g(\mathbf{r} + u\mathbf{\Omega}, \mathbf{\Omega}) = q_g(\mathbf{r} + u\mathbf{\Omega}, \mathbf{\Omega}) \tag{2.14}$$

Given an observation point at A($\mathbf{r}$) and looking at the neutrons that were emitted at the point B with solid angle $\mathbf{\Omega}$, with the notation depicted in Figure 2.2, a more convenient expression can be obtained from Eq. (2.14), i.e.,

$$-\frac{d}{dR}\psi_g(\mathbf{r} - R\mathbf{\Omega}, \mathbf{\Omega}) + \Sigma_{t,g}(\mathbf{r} - R\mathbf{\Omega})\psi_g(\mathbf{r} - R\mathbf{\Omega}, \mathbf{\Omega}) = q_g(\mathbf{r} - R\mathbf{\Omega}, \mathbf{\Omega}) \tag{2.15}$$

After manipulation of the equation above (see details in Ref. [15]), the integral form can be derived and reads:

$$\psi_g(\mathbf{r},\mathbf{\Omega}) = \psi_g(\mathbf{r} - R\mathbf{\Omega},\mathbf{\Omega})e^{-\tau_g(\mathbf{r},\mathbf{r}-R\mathbf{\Omega})} + \int_0^R q_g(\mathbf{r} - R'\mathbf{\Omega},\mathbf{\Omega})e^{-\tau_g({}_g\mathbf{r},\mathbf{r}-R'\mathbf{\Omega})}dR' \tag{2.16}$$

where $\tau_g(\mathbf{r},\mathbf{r} - R'\mathbf{\Omega})$ is the optical path between $\mathbf{r}$ and $\mathbf{r} - R'\mathbf{\Omega}$ defined by:

$$\tau_g(\mathbf{r},\mathbf{r} - R'\mathbf{\Omega}) = \int_0^{R'} \Sigma_{t,g}(\mathbf{r} - R''\mathbf{\Omega})dR'' \tag{2.17}$$

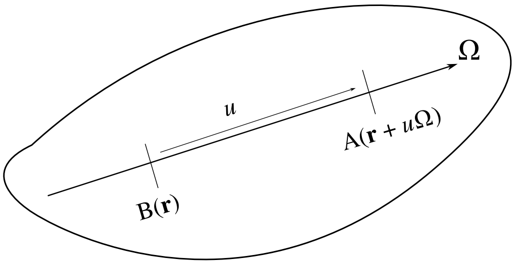


Figure 2.1: Neutron travel along the direction $\mathbf{\Omega}$. The flight distance between B and A is denoted by u. Figure derived from Ref. [14].

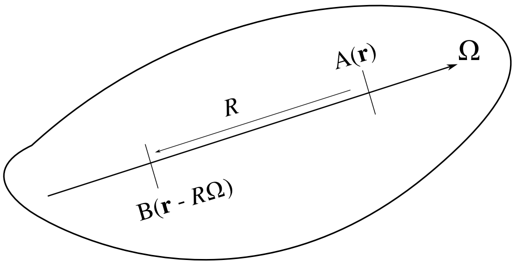


Figure 2.2: Neutron travel along the direction $\mathbf{\Omega}$ between B($\mathbf{r} - R\mathbf{\Omega}$) and A($\mathbf{r}$). The flight distance between B and A is denoted by R. Figure derived from Ref. [14].

2.2 Interface current method

The strategy of the Interface Current Method (ICM) is to divide the overall system into smaller sub-systems according to a spatial coarse mesh. Then, the partial neutron currents are determined assuming that the neutron emission densities inside the subsystem are known; this is also referred to as solving the global problem. Every sub-system is further discretized using a spatial fine mesh to calculate the neutron emission densities assuming that the partial neutron currents are known; this leads to the solution of the so-called local problem. An example of the two computational meshes for a square lattice of 5x5 fuel pins is depicted in Figure 2.3. The global problem and local problem equations are derived from the integral neutron transport equation in the following two subsections.

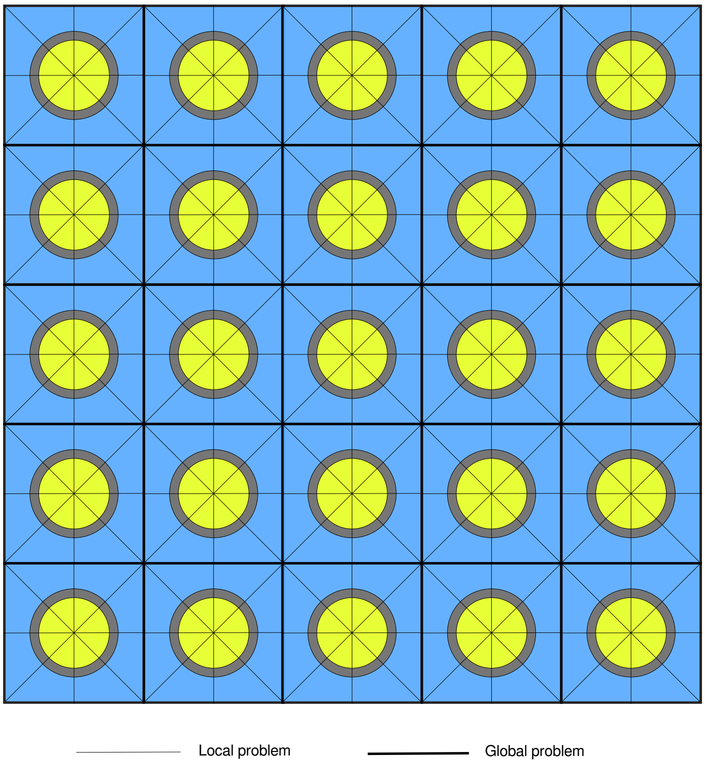


Figure 2.3: Representation of a coarse mesh (bold line) and a fine mesh (thin line). Figure derived from Ref. [14].

2.2.1 Local problem

Recasting Eq. (2.16) and rewriting it for the incoming neutron angular flux as shown in Figure 2.4 gives:

$$\psi_g(\mathbf{r},\mathbf{\Omega}) = \psi_g(\mathbf{r}_{in},\mathbf{\Omega})e^{-\tau_g(\mathbf{r},\mathbf{r}_{in})} + \int_{s_{in}}^{0} q_g(\mathbf{r}+s'\mathbf{\Omega},\mathbf{\Omega})e^{-\tau_g(\mathbf{r},\mathbf{r}+s'\mathbf{\Omega})}ds', \tag{2.18}$$

where $\mathbf{r}_{in} = \mathbf{r} + s_{in}\mathbf{\Omega}$. Integrating the equation above in all directions to remove the angular dependency on the left-hand side of the equation, the balance of neutrons along the path of the direction of travel of the neutron results in:

$$\phi_g(\mathbf{r}) = \int_{4\pi} \psi_g(\mathbf{r}_{in},\mathbf{\Omega})e^{-\tau_g(\mathbf{r},\mathbf{r}_{in})}d^2\mathbf{\Omega} + \int_{4\pi}\int_{s_{in}}^{0} q_g(\mathbf{r}+s'\mathbf{\Omega},\mathbf{\Omega})e^{-\tau_g(\mathbf{r},\mathbf{r}+s'\mathbf{\Omega})}ds'd^2\mathbf{\Omega}, \tag{2.19}$$

If we consider the boundary along the direction $\mathbf{\Omega}$ from the point defined by the position $\mathbf{r}_{in} = \mathbf{r} + s_{in}\mathbf{\Omega}$, the infinitesimal change in surface area ($d^2\mathbf{r}'$) spanned by a change in solid angle $d^2\mathbf{\Omega}$ is given by:

$$d^2\mathbf{r}'|\mathbf{\Omega}\cdot\mathbf{N}| = s_{in}^2 d^2\mathbf{\Omega} = ||\mathbf{r}-\mathbf{r}_{in}||^2 d^2\mathbf{\Omega}, \tag{2.20}$$

at any point defined by:

$$\mathbf{r}' = \mathbf{r} + s'\mathbf{\Omega}. \tag{2.21}$$

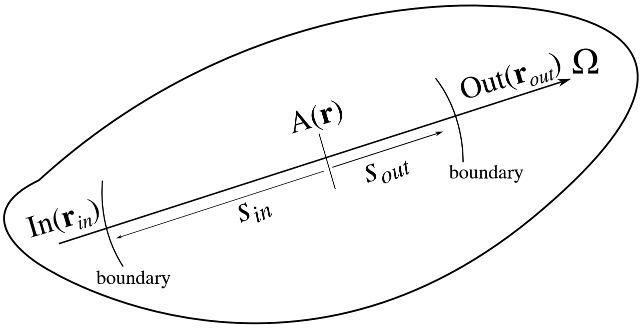


Figure 2.4: Notation used to derive the interface current method. Figure derived from Ref. [14].

Additionally, the infinitesimal change in volume ($d^3\mathbf{r}'$) spanned by change in solid angle $\boldsymbol{\Omega}$ and distance s' is given by:

$$d^3\mathbf{r}' = s_{in}^2 ds' d^2\boldsymbol{\Omega} = ||\mathbf{r} - \mathbf{r}'||^2 ds' d^2\boldsymbol{\Omega}, \tag{2.22}$$

Considering the right hand side of Eq. (2.19), the first integral becomes a surface integral by inserting Eq. (2.20) and the second integral becomes a volume integral by inserting Eq. (2.22). This leads to the following expression:

$$\phi_g(\mathbf{r}) = \int_S \psi_{g,in}(\mathbf{r}', \boldsymbol{\Omega}) \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r} - \mathbf{r}'||^2} |\boldsymbol{\Omega} \cdot \mathbf{N}'| d^2\mathbf{r}' + \int_V q_g(\mathbf{r}', \boldsymbol{\Omega}) \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r} - \mathbf{r}'||^2} d^3\mathbf{r}'. \tag{2.23}$$

In the interface current method formalism, one important step in the derivation of the equations is to partition the total volume V of the system into sub-volumes V_i such that $V = \bigcup_i V_i$ and to partition the total surface S of the system into smaller sub-surfaces S_a such that $S = \bigcup_a S_a$. Then, the previous equation can be rewritten as:

$$\phi_g(\mathbf{r}) = \sum_a \int_{S_a} \psi_{g,in}(\mathbf{r}', \boldsymbol{\Omega}) \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r} - \mathbf{r}'||^2} |\boldsymbol{\Omega} \cdot \mathbf{N}'| d^2\mathbf{r}' + \sum_i \int_{V_i} q_g(\mathbf{r}', \boldsymbol{\Omega}) \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r} - \mathbf{r}'||^2} d^3\mathbf{r}', \tag{2.24}$$

The equation above is multiplied by the g-th transport corrected macroscopic total cross section $\Sigma^0_{t,g}$, integrated on a volume V_j, and assuming isotropic emission density:

$$\begin{aligned}\int_{V_j} \Sigma^0_{t,g}(\mathbf{r})\phi_g(\mathbf{r}) d^3\mathbf{r} = & \\ & \sum_a \int_{S_a} \psi_{g,in}(\mathbf{r}', \boldsymbol{\Omega}) \int_{Vj} \Sigma^0_{t,g}(\mathbf{r}) \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r} - \mathbf{r}'||^2} |\boldsymbol{\Omega} \cdot \mathbf{N}'| d^2\mathbf{r}' d^3\mathbf{r} \\ & + \sum_i \int_{V_i} Q_g(\mathbf{r}') \int_{V_j} \Sigma^0_{t,g}(\mathbf{r}) \ \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{4\pi||\mathbf{r} - \mathbf{r}'||^2} d^3\mathbf{r}' d^3\mathbf{r},\end{aligned} \tag{2.25}$$

where $Q_g(\mathbf{r}')$ represents the emission density under the assumption of isotropic scattering. Eq. (2.25) can be arranged in a compact form, i.e.,

$$\Sigma^0_{t,g,j}\phi_{g,j}V_j = \sum_a S_a J_{in,g,a} P_{g,a\to j} + \sum_i V_i Q_{g,i} P_{g,i\to j} \tag{2.26}$$

The volume averaged quantities to preserve the reaction rates are given by:

$$\phi_{g,j} = \frac{1}{V_j} \int_{V_j} \phi_g(\mathbf{r}) d^3\mathbf{r} \tag{2.27}$$

$$Q_{g,i} = \frac{1}{V_i} \int_{V_i} Q_g(\mathbf{r}) d^3\mathbf{r} \tag{2.28}$$

$$\Sigma^0_{t,g,j} = \frac{1}{V_j \phi_{g,j}} \int_{V_j} \Sigma^0_{t,g}(\mathbf{r}) \phi_g(\mathbf{r}) d^3\mathbf{r} \tag{2.29}$$

with the surface-averaged incoming neutron current defined by:

$$J_{in,g,a} = \frac{1}{S_a} \int_{S_a} \int_{(2\pi),\mathbf{\Omega}\cdot\mathbf{N}<0} \psi_g(\mathbf{r}', \mathbf{\Omega}) |\mathbf{\Omega} \cdot \mathbf{N}| d^2\mathbf{r}' d^2\mathbf{\Omega} \tag{2.30}$$

Since $e^{-\tau_g(\mathbf{r},\mathbf{r}')}$ represents the probability of non-interaction between **r** and **r**' for the energy group g, $P_{g,i\to j}$ is the probability for a neutron emitted isotropically and homogeneously in the region i to have its next interaction in the region j and it is given by:

$$P_{g,i\to j} = \frac{1}{V_i} \int_{V_i} \int_{V_j} \Sigma^0_{t,g}(\mathbf{r}) \; \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{4\pi ||\mathbf{r} - \mathbf{r}'||^2} d^3\mathbf{r}' d^3\mathbf{r}. \tag{2.31}$$

Likewise, $P_{g,a\to j}$ is the probability for a neutron emitted isotropically and homogeneously from surface a to have its first interaction in region j and it is given by:

$$P_{g,a\to j} = \frac{1}{S_a J_{in,g,a}} \int_{S_a} \psi_{g,in}(\mathbf{r}', \mathbf{\Omega}) \int_{Vj} \Sigma^0_{t,g}(\mathbf{r}) \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r} - \mathbf{r}'||^2} |\mathbf{\Omega} \cdot \mathbf{N}'| d^2\mathbf{r}' d^3\mathbf{r} \tag{2.32}$$

2.2.2 Global problem

To obtain the equations for the global problem, the steps are similar to those for the local problem equations. Taking again Eq. (2.16) and rewriting it for the outgoing angular flux, it can be read as:

$$\psi_g(\mathbf{r}_{out}, \mathbf{\Omega}) = \psi_g(\mathbf{r}_{in}, \mathbf{\Omega}) e^{-\tau_g(\mathbf{r}_{out},\mathbf{r}_{in})} + \int_{s_{in}}^{0} q_g(\mathbf{r}_{out} + s'\mathbf{\Omega}, \mathbf{\Omega}) e^{-\tau_g(\mathbf{r}_{out},\mathbf{r}+s'\mathbf{\Omega})} ds'. \tag{2.33}$$

The previous equation is multiplied by $|\mathbf{\Omega} \cdot \mathbf{N}'|$ and integrated over a given surface S and integrated again for all solid angles in a half sphere such that $\mathbf{\Omega} \cdot \mathbf{N} > 0$. The integral with respect to the solid angle is replaced by a surface integral and the integral with respect to the solid angle and the distance s_{in} can be replaced by a volume integral. After partitioning the total volume of the system V into smaller sub-volumes V_i and the total surface S into smaller sub-surfaces S_a and S_b, the equations for the global problem, under the condition that the neutron emission density is isotropic, can be written as:

$$\begin{aligned} &\int_S \int_{(2\pi),\mathbf{\Omega}\cdot\mathbf{N}>0} \psi_{g,out}(\mathbf{r}, \mathbf{\Omega}) |\mathbf{\Omega} \cdot \mathbf{N}'| d^2\mathbf{r} d^2\mathbf{\Omega} \\ &\quad = \sum_b \int_{S_a} |\mathbf{\Omega} \cdot \mathbf{N}| d^2\mathbf{r} \int_{S_b} \psi_{g,in}(\mathbf{r}', \mathbf{\Omega}) \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r} - \mathbf{r}'||^2} |\mathbf{\Omega} \cdot \mathbf{N}'| d^2\mathbf{r}' \\ &\quad + \sum_i \int_{S_a} |\mathbf{\Omega} \cdot \mathbf{N}| d^2\mathbf{r} \int_{V_i} Q_g(\mathbf{r}') \frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{4\pi ||\mathbf{r} - \mathbf{r}'||^2} d^3\mathbf{r}' \end{aligned} \tag{2.34}$$

The compact form of Eq. (2.34) is given as:

$$S_a J_{out,g,a} = \sum_b S_b J_{in,g,b} P_{g,b\to a} + \sum_i V_i Q_{g,i} P_{g,i\to a} \tag{2.35}$$

where the following averaged quantities are defined:

$$J_{out,g,a} = \frac{1}{S_a}\int_{S_a}\int_{(2\pi),\mathbf{\Omega}\cdot\mathbf{N}>0} \psi_{out,g}(\mathbf{r},\mathbf{\Omega})|\mathbf{\Omega}\cdot\mathbf{N}'|d^2\mathbf{r}d^2\mathbf{\Omega} \tag{2.36}$$

Since $e^{-\tau_g(\mathbf{r},\mathbf{r}')}$ represents the probability of non-interaction between **r** and **r**' for the energy group g, $P_{g,i\to a}$ is the probability for a neutron emitted isotropically and homogeneously in the region i to leave the sub-system without interacting crossing the surface a and it is given by:

$$P_{g,i\to a} = \frac{1}{V_i}\int_{S_a}|\mathbf{\Omega}\cdot\mathbf{N}|d^2\mathbf{r}\int_{V_i}\frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{4\pi||\mathbf{r}-\mathbf{r}'||^2}d^3\mathbf{r}' \tag{2.37}$$

Likewise, $P_{g,b\to a}$ is the probability for a neutron emitted isotropically and homogeneously from surface b to exit the sub-system without interacting through surface a being given by:

$$P_{g,b\to a} = \frac{1}{S_b J_{in,g,b}}\int_{S_a}|\mathbf{\Omega}\cdot\mathbf{N}|d^2\mathbf{r}\int_{S_b}\psi_{in,g}(\mathbf{r}',\mathbf{\Omega})\frac{e^{-\tau_g(\mathbf{r},\mathbf{r}')}}{||\mathbf{r}-\mathbf{r}'||^2}|\mathbf{\Omega}\cdot\mathbf{N}'|d^2\mathbf{r}' \tag{2.38}$$

2.2.3 Overall problem

If Eqs. (2.26) and (2.35) are used directly to solve the entire system, the number of probabilities to calculate would be enormous and prohibitively expensive. If we divide the system into sub-systems and define S_I and V_I as subsets containing all the surfaces and the volumes of the system respectively, we can rewrite the equations for the global problem and local problem for each sub-system as:

$$\Sigma^0_{t,g,j}\phi_{g,j}V_j = \sum_{a\in S_I} S_a J_{in,g,a} P_{g,a\to j} + \sum_{i\in V_I} V_i Q_{g,i} P_{g,i\to j} \quad \text{for } j\in V_I \tag{2.39}$$

$$S_a J_{out,g,a} = \sum_{b\in S_I} S_b J_{in,g,b} P_{g,b\to a} + \sum_{i\in V_I} V_i Q_{g,i} P_{g,i\to a} \quad \text{for } a\in S_I \tag{2.40}$$

with the emission density in region i given by:

$$Q_{g,i} = \frac{1}{4\pi}\sum_{g'=1}^{G}\left[\Sigma^0_{s0,g'\to g,i} + \frac{\chi_{g,i}}{k}\nu\Sigma_{f,g',i}\right]\phi_{g',i} \tag{2.41}$$

where $\Sigma^0_{s0,g'\to g,i}$ is the scattering cross section with transport correction applied at the level 0. Using matrix notation, the Eqs. (2.39) and (2.40) can be written as follows:

$$\Phi = \mathbf{S}J_{in} + \Phi_{src} \tag{2.42}$$

$$J_{out} = \mathbf{R}J_{in} + J_{src} \tag{2.43}$$

with:

$$\Phi_{src} = \mathbf{T}Q \tag{2.44}$$

$$J_{src} = \mathbf{U}Q \tag{2.45}$$

In Eqs. (2.42) and (2.43) the matrices **S**, **R**, **U**, **T** are block matrices where each block is associated with a coarse node of the global mesh, i.e.,:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}^1 & 0 & \cdots & 0 \\ 0 & \mathbf{S}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{S}^I \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} \mathbf{R}^1 & 0 & \cdots & 0 \\ 0 & \mathbf{R}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{R}^I \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}^1 & 0 & \cdots & 0 \\ 0 & \mathbf{U}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{U}^I \end{bmatrix} \qquad \mathbf{T} = \begin{bmatrix} \mathbf{T}^1 & 0 & \cdots & 0 \\ 0 & \mathbf{T}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \mathbf{T}^I \end{bmatrix}$$

If we consider a coarse node (or sub-system) with I or J number of regions being enclosed by A or B number of surfaces, with $I = J$ and $A = B$, the matrices $\mathbf{S}^I$, $\mathbf{R}^I$, $\mathbf{U}^I$, and $\mathbf{T}^I$ are given by:

$$\mathbf{S}^I = \begin{bmatrix} S_1 P_{g,1\to1} & S_2 P_{g,2\to1} & \cdots & S_A P_{g,A\to1} \\ S_1 P_{g,1\to2} & S_2 P_{g,2\to2} & \cdots & S_A P_{g,A\to2} \\ \vdots & \vdots & \ddots & \vdots \\ S_1 P_{g,1\to J} & S_2 P_{g,2\to J} & \cdots & S_A P_{g,A\to J} \end{bmatrix} \qquad \mathbf{R}^I = \begin{bmatrix} S_1 P_{g,1\to1} & S_2 P_{g,2\to1} & \cdots & S_B P_{g,B\to1} \\ S_1 P_{g,1\to2} & S_2 P_{g,2\to2} & \cdots & S_B P_{g,B\to2} \\ \vdots & \vdots & \ddots & \vdots \\ S_1 P_{g,1\to A} & S_2 P_{g,2\to A} & \cdots & S_B P_{g,B\to A} \end{bmatrix}$$

$$\mathbf{U}^I = \begin{bmatrix} V_1 P_{g,1\to1} & V_2 P_{g,2\to1} & \cdots & V_I P_{g,I\to1} \\ V_1 P_{g,1\to2} & V_2 P_{g,2\to2} & \cdots & V_I P_{g,I\to2} \\ \vdots & \vdots & \ddots & \vdots \\ V_1 P_{g,1\to A} & V_2 P_{g,2\to A} & \cdots & V_I P_{g,I\to A} \end{bmatrix} \qquad \mathbf{T}^I = \begin{bmatrix} V_1 P_{g,1\to1} & V_2 P_{g,2\to1} & \cdots & V_I P_{g,I\to1} \\ V_1 P_{g,1\to2} & V_2 P_{g,2\to2} & \cdots & V_I P_{g,I\to2} \\ \vdots & \vdots & \ddots & \vdots \\ V_1 P_{g,1\to J} & V_2 P_{g,2\to J} & \cdots & V_I P_{g,I\to J} \end{bmatrix}$$

If we consider a system with G energy groups and J number of regions, the vectors Φ and Q containing the scalar flux and the total source of neutrons would be respectively given by:

$$\mathbf{\Phi} = \begin{bmatrix} \phi_{1,1} \\ \vdots \\ \phi_{1,J} \\ \vdots \\ \phi_{G,1} \\ \vdots \\ \phi_{G,J} \end{bmatrix} \qquad Q = \begin{bmatrix} Q_{1,1} \\ \vdots \\ Q_{1,J} \\ \vdots \\ Q_{G,1} \\ \vdots \\ Q_{G,J} \end{bmatrix} \tag{2.46}$$

The outward currents from one sub-system are inward currents to another. In addition, the inward currents at the boundary of the system depend on the type of chosen condition. The relationships between outward and inward currents of sub-systems and the boundary conditions are described by using a connectivity matrix $\mathbf{M}$ as follows:

$$\mathbf{J}_{in} = \mathbf{M}\mathbf{J}_{out}. \tag{2.47}$$

After rearranging, the resultant equation for the neutron current balance is given by:

$$\mathbf{J}_{in} = \left(\mathbf{I} - \mathbf{M}\mathbf{R}\right)^{-1} \mathbf{M} J_{src}, \tag{2.48}$$

which will be solved simultaneously with Eq. (2.42). As an illustrative example, if a system with 4 coarse nodes and 16 surfaces with reflective boundary conditions in all directions is taken (see Figure 2.5), Eq. (2.47) for each energy group is given by:

$$\begin{bmatrix} J_{in,1} \\ J_{in,2} \\ J_{in,3} \\ J_{in,4} \\ J_{in,5} \\ J_{in,6} \\ J_{in,7} \\ J_{in,8} \\ J_{in,9} \\ J_{in,10} \\ J_{in,11} \\ J_{in,12} \\ J_{in,13} \\ J_{in,14} \\ J_{in,15} \\ J_{in,16} \end{bmatrix} = \begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix} \cdot \begin{bmatrix} J_{out,1} \\ J_{out,2} \\ J_{out,3} \\ J_{out,4} \\ J_{out,5} \\ J_{out,6} \\ J_{out,7} \\ J_{out,8} \\ J_{out,9} \\ J_{out,10} \\ J_{out,11} \\ J_{out,12} \\ J_{out,13} \\ J_{out,14} \\ J_{out,15} \\ J_{out,16} \end{bmatrix}$$

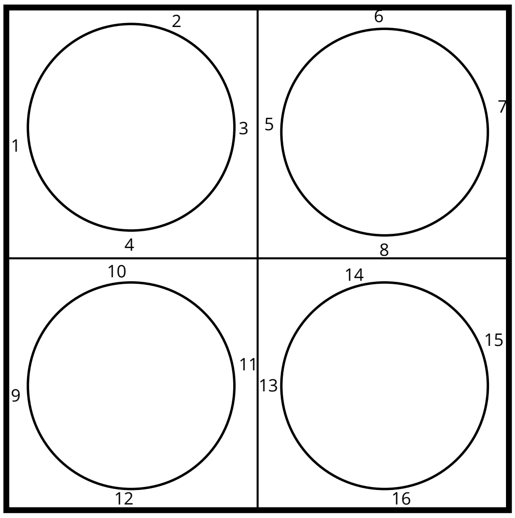


Figure 2.5: 2x2 square lattice with 4 coarse nodes and 4 surfaces by coarse node. The surfaces are numbered clockwise for each coarse node, starting from the west surface.

Chapter 3

Hybrid Solver

The hybrid solution strategy is introduced in section 3.1. The general algorithm of the solver is addressed in section 3.2.

3.1 Hybrid solution strategy for the Interface Current method

In Chapter 2 the Interface Current method (ICM) was derived. It consists of a global problem and a local problem, Eqs. (2.48) and (2.42) respectively. To be able to solve both problems a set of collision probabilities and the cross sections need to be pre-calculated. This is done with the Monte Carlo software Serpent [16]. Then a deterministic calculation is carried out with the ICM to evaluate the multiplication factor and the scalar flux in the entire system. The methodology for the estimation of the collision probabilities through Monte Carlo is based on a previous feasibility study performed at Chalmers University of Technology (see Ref. [13]).

3.1.1 Serpent

The Monte Carlo software Serpent is an open-source continuous-energy 3D neutron and photon transport developed by the Technical Research Centre (VTT) of Finland. It can be used for a wide range of applications, such as reactor modeling, group constant generation, radiation transport, and fusion. The physics model for neutron transport relies on classical collision kinematics and Evaluated Nuclear Data Files (ENDF) which are based on experimental measurements and theoretical nuclear models. In Serpent, universe-based constructive solid geometry (CSG) is used to model the geometry of the systems of interest. Computer Aid Design (CAD)-based geometry capabilities are also available for complex irregular structures. The software is developed in a hybrid multi-core and multi-nodal parallel architecture that can be installed in personal computers or large supercomputer clusters. Within reactor applications, Serpent can calculate neutron flux, power, and reaction rates distribution in different cells or materials through standard detectors [16].

To simulate the neutron transport in a nuclear system, each neutron is tracked individually during its lifetime, i.e., neutron history. Accordingly, the initial position, energy, and fly direction of a neutron in the system are randomly sampled. Then, the distance of flight is sampled according to an exponential distribution. Finally, the neutron is tracked after several collisions until it is absorbed or crosses the boundary of the computational domain. If the neutron causes a fission, the neutron history is terminated and new neutrons are produced which are tracked too. After a fixed number of neutron histories are simulated, a cycle is completed and the process is repeated for a selected number of cycles.

The neutron tracking in Serpent relies on two methods: ray-tracing-based surface tracking and rejection sampling-based delta-tracking. These techniques have their advantages and disadvantages that depend on the complexity of the geometry and the localization of strong neutron absorbers. Serpent switches between these two techniques on the fly which increases the overall computational and probabilistic efficiencies of the simulation [17].

During the Monte Carlo simulation, specific quantities of interest, known as tallies, such as reaction rates and neutron flux, can be recorded. These tallies are derived from the total number of neutron histories in a cycle. After simulating a fixed number of cycles, the final tally is estimated as a statistical average, providing a tally value and the associated standard deviation.

3.1.2 Previous in-house work

The methodology for the calculation of the collision probabilities that is used in this work was first tested within a previous feasibility study performed in the Nuclear Engineering group of Chalmers University Of Technology [13]. Accordingly, the collision probabilities were estimated for a typical square lattice of a light-water system in two energy groups. Two coarse meshes were used, a single-pin and 2x2 fuel pins. Then, the multiplication factor and scalar flux were calculated via a hybrid methodology obtaining good agreement with the reference solution. Additionally, the hybrid solver algorithm was implemented in MATLAB.

In this project, the computational framework is rewritten in Python with C extensions in charge of heavy calculations and built in a more coherent and flexible architecture that can accommodate more easily future developments. In addition, the focus is on the optimization of the methodology for the simulations of fast systems, which are characterized by hexagonal geometry and finer energy group structures.

3.2 Design

The hybrid computational framework developed in this work is shown in Figure 3.1, and it was built in a modular structure focused on scalability in order to facilitate future expansions. The modules are named Materials, Geometry, Mesh, Serpent, CollisionProbabilities, CrossSections, and ICM, each of them written in Python. First, the framework requires the preparation of a Python input file that consists of a geometry description of the system with the related material properties. Information about the types of meshes associated with the global problem and local problems is also required. The calculation of the collision probabilities is performed with Serpent and requires to select a cross section library, the neutron population, skip cycles, and active cycles which will be provided through the input file.

Based on the type of mesh provided in the input file, the second step in the framework is to generate the coarse mesh and fine mesh. Then, the cross sections and the collision probabilities need to be calculated for the coarse nodes. In the next step, the framework generates a set of Serpent input files for each coarse node, that are simulated sequentially to estimate neutron emission and reaction rates. A file is generated for every region within the fine mesh, the surfaces of the coarse node, and one file to generate the macroscopic cross sections. After this step is completed, the hybrid framework reads and interprets the output data from the Monte Carlo simulations to calculate the collision probabilities. Finally, the collision probabilities and the macroscopic cross sections are provided as parameters to a static ICM-based deterministic solver where the effective multiplication factor k_{eff} and the scalar fluxes inside the fine regions are calculated.

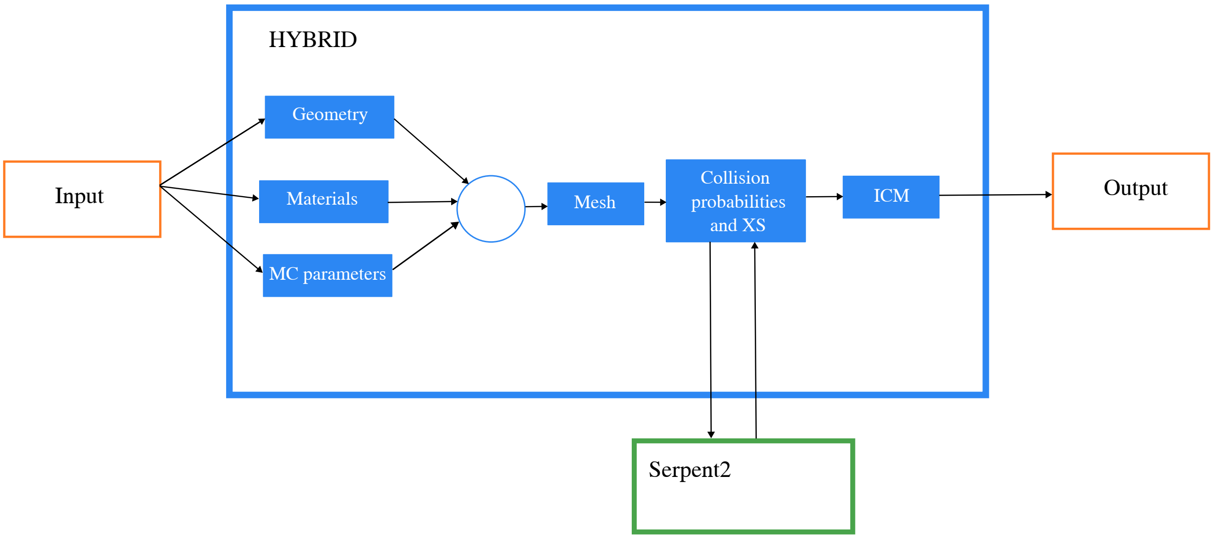


Figure 3.1: Data flow for the hybrid framework

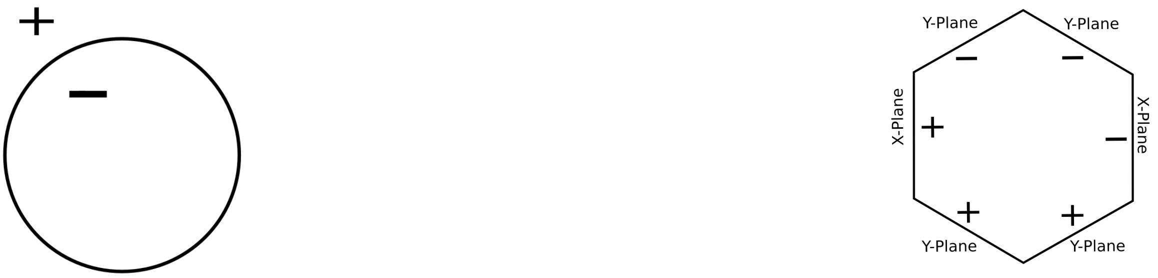
3.3 Geometry representation

The geometric description of the system required in the input is generated using a Constructive Solid Geometry (CSG) approach [18]. The CSG technique is used in computer graphics and computer-aid design (CAD) to represent complex bodies from basic shapes or surfaces such as spheres, cylinders, planes, and prisms. According to the CSG formulation used in the hybrid framework, the system geometry is built with 5 components: surfaces, half spaces, regions, cells, and universes.

A basic surface defined in the Euclidean space is described by a function $f(x, y, z)$ that divides the Euclidean space into three subsets of coordinates, where the first subset is given by $(f(x, y, z) = 0)$ and is related to the surface. A half space is defined by the two other subsets of coordinates. The second subset of coordinates corresponds to $f(x, y, z) < 0$, i.e., to the negative half-space. The third subset is the positive half-space with $f(x, y, z) > 0$. This is represented in Figure 3.2(a) for a circular surface. A region is then the result of a boolean operation – such as intersection and union – between one or more half-spaces. For example, a hexagon can be represented by the intersection of six half-spaces defined by 6 planes, see

Figure 3.2(b).

A cell can then be defined inside a region and be filled with a material. Finally, one or more cells can then be grouped into a universe. The geometry of the system is stored in a data structure often called Binary Space Partitioning (BSP) tree, where each leaf of the tree is characterized by a boolean operation. A nuclear reactor core is typically arranged as a lattice of fuel pins/assemblies. To reproduce this lattice configuration, a hierarchy of components can be used in CSG, as shown in Figure 3.3.



(a) circular surface

(b) hexagonal surface

Figure 3.2: Description of a half space for a surface.

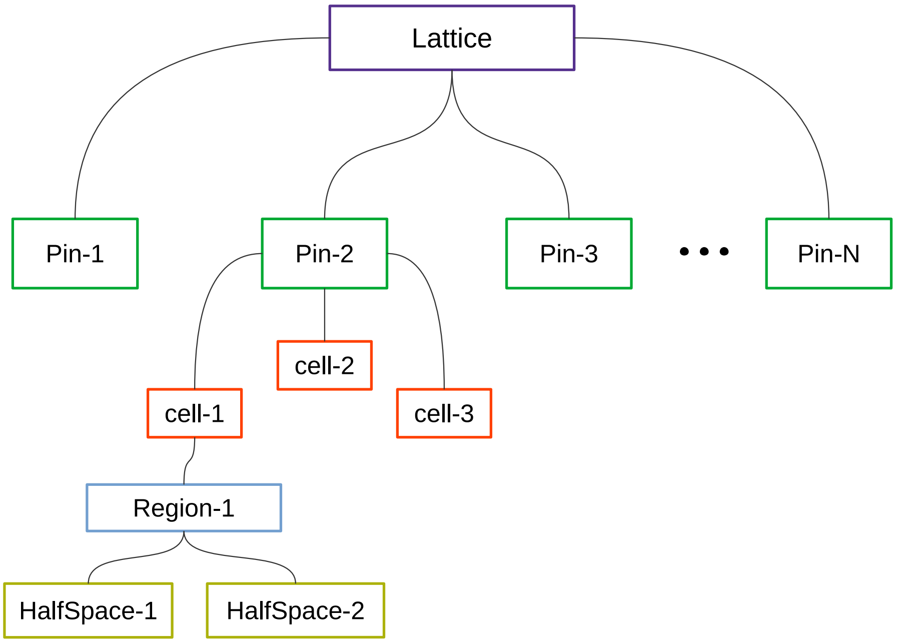


Figure 3.3: Hierarchical representation of a lattice universe.

3.4 Collision probabilities

Two sets of collision probabilities need to be estimated prior to the solution of the ICM Eqs. (2.42) and (2.48), i.e., one for the neutrons emitted from the fine regions and one for the neutrons emitted from the surfaces. The total emission of neutrons from region i and from surface a are respectively equal to:

$$N_{g,i} = \sum_j R_{g,i\rightarrow j} + \sum_a R_{g,i\rightarrow a}, \tag{3.1}$$

$$N_{g,a} = \sum_i R_{g,a\rightarrow i} + \sum_b R_{g,a\rightarrow b}, \tag{3.2}$$

where:

- $N_{g,i}$: Total neutron emission from region i in energy group g.
- $N_{g,a}$: Total neutron emission from surface a in energy group g.
- $R_{g,i\rightarrow j}$: Neutrons in energy group g, that are emitted in region i and have the first collision in region j.
- $R_{g,i\rightarrow a}$: Neutrons in energy group g, that are emitted in region i and escape through surface a without having a collision.
- $R_{g,a\rightarrow i}$: Neutrons in energy group g, that are emitted from surface a and have the first collision in region i.
- $R_{g,a\rightarrow b}$: Neutrons in energy group g, that are emitted from surface a and escape through surface b without having a collision.

The collision probabilities can then be as follows:

$$P_{g,i\rightarrow j} = \frac{R_{g,i\rightarrow j}}{N_{g,i}} \tag{3.3}$$

$$P_{g,i\rightarrow a} = \frac{R_{g,i\rightarrow a}}{N_{g,i}} \tag{3.4}$$

$$P_{g,a\rightarrow i} = \frac{R_{g,a\rightarrow i}}{N_{g,a}} \tag{3.5}$$

$$P_{g,a\rightarrow b} = \frac{R_{g,a\rightarrow b}}{N_{g,a}} \tag{3.6}$$

These probabilities are equivalent to those discussed in Section 2.2, see Eqs. (2.31), (2.32), (2.37), and (2.38). To evaluate Eqs. (3.3) - (3.6) for a coarse node, Monte Carlo is used. For this purpose, a model of the fine mesh within the coarse node is required, which includes a set of detectors to tally the quantities of interest $R_{g,i\rightarrow j}$, $R_{g,i\rightarrow a}$, $R_{g,a\rightarrow i}$, and $R_{g,a\rightarrow b}$. The set of detectors varies depending on the type of collision probability to be calculated. Considering a system with nuclear fuel and non-fuel regions, the collision probabilities are computed for three different types of neutron emissions, i.e., neutrons emitted from the fuel regions, neutrons emitted from the non-fuel regions, and neutrons emitted from the surfaces.

The neutrons emitted from one region can have their next collision in one of the regions or escape through one of the surfaces of the coarse node. Labels are used to track the neutrons before their collisions. In Serpent, detectors can be set to count neutrons that interact in a region and assign a flag to them. When the neutrons have a collision or cross a surface, the flag is removed and the corresponding detector is scored. The same methodology applies to the neutrons coming from the surfaces. Once the emission rates are estimated, all the necessary collision probabilities are calculated for every coarse node. The set of detectors used to estimate the neutron emission rates is described in the following sections.

3.4.1 Neutron emission from fuel regions

A hexagonal fuel pin cell is considered and the total emission of neutrons from the fuel for the case in Figure 3.4 is calculated with the following equation:

$$N_{g,fuel} = R_{g,fuel\rightarrow fuel} + R_{g,fuel\rightarrow coolant} + \sum_{a}^{6} R_{g,fuel\rightarrow a} \tag{3.7}$$

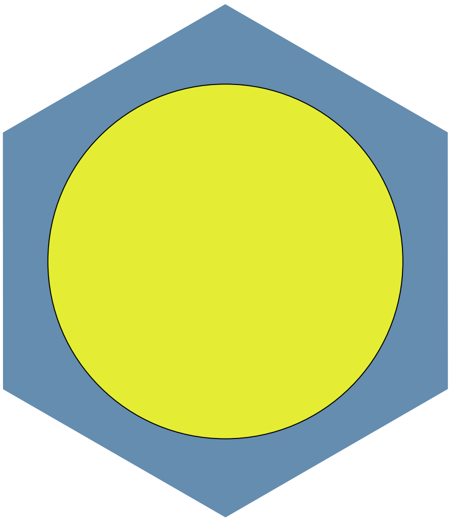


Figure 3.4: Simplified hexagonal fuel cell with fuel (in yellow) and coolant (in blue).

In the last equation eight quantities need to be calculated. The quantity $R_{g,fuel\rightarrow fuel}$ indicates the neutron with first collision in the fuel, $R_{g,fuel\rightarrow coolant}$ the neutrons that have their first collision in the coolant, and $R_{g,fuel\rightarrow a}$ the neutrons that escape through each of the six surfaces of the cell without having a collision. Since the neutrons that are produced in the fuel cannot be flagged in Serpent, the neutron emission rate $R_{g,fuel\rightarrow fuel}$ is calculated indirectly with the following equation:

$$R_{g,fuel\rightarrow fuel} = R_{g,fuel,total} - R_{g,coolant\rightarrow fuel} - \sum_{a}^{6} R_{g,a\rightarrow fuel} \tag{3.8}$$

The total reaction rate $R_{g,fuel,total}$ provides the neutrons that have a collision in the fuel after having a collision in other regions or being emitted from the surfaces.

The process of counting the neutrons is illustrated in Figure 3.5. This is calculated as follows:

1. A detector for the inward current J_{in} is placed on the surface of the fuel and a flag #1 is set.

2. A detector for the outward current J_{out} is placed on the surface of the fuel and it scores only if the flag #1 for J_{in} is not set, then a flag #2 is set. This flagging procedure eliminates the neutrons that might have crossed the fuel region without interacting. Thus, this detector only accounts for the neutrons that are emitted inside the fuel region and exit the region.

3. All the neutrons that interact elsewhere in the cell and then interact in the fuel are scored in another detector only if the flag #1 is set, i.e., second and third terms of the right-hand side of Eq. (3.8). Then the flag #1 is reset.

4. The neutrons that are emitted from the fuel and have a collision in other region or escape through one of the surfaces, i.e., second and third terms in the right-hand side of Eq. (3.7), are estimated with several detectors only if the flag #2 is set. Then the flag #2 is reset.

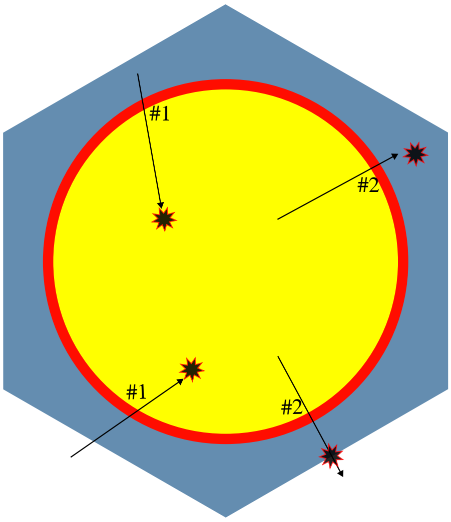


Figure 3.5: Counting of neutrons emitted from a fuel region.

3.4.2 Neutron emission from non-fuel regions

The total emission of neutrons from the coolant is given by:

$$N_{g,coolant} = R_{g,coolant \to fuel} + R_{g,coolant \to coolant} + \sum_{a}^{6} R_{g,coolant \to a} \tag{3.9}$$

The steps to estimate the neutron emission from a non-fuel region differ from that of the fuel region. In this case, the set of detectors uses only a single flag to label the neutrons. First, a detector is defined to score the neutrons that interact in the non-fuel region of interest and the neutrons are flagged with #1. Second, the neutrons that interact later in the same non-fuel region are flagged again with #1 after having the collision. Third, the neutrons that have a collision in other regions or escape through one of the surfaces are scored in the other detectors, finally, the flag #1 is reset. The counting procedure for neutrons emitted from a non-fuel region is illustrated in Figure 3.6.

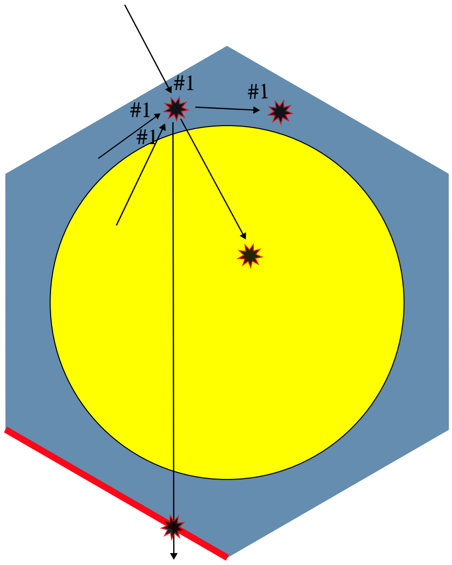


Figure 3.6: Counting of neutrons emitted from a non-fuel region.

3.4.3 Neutron emission from surfaces

The total emission of neutrons from the surfaces is given by:

$$N_{g,a} = R_{g,a\rightarrow fuel} + R_{g,a\rightarrow coolant} + \sum_{b}^{6} R_{g,a\rightarrow b} \tag{3.10}$$

To estimate the neutrons emitted from a generic surface a and having a collision in one of the regions or escaping through the surfaces, the procedure consists of two steps. First, a detector is assigned to score the incoming neutrons in each surface and a different flag for every surface is set. Second, if the neutron has a collision and the flag from a given surface is set, the detector is scored, and the flag is reset. This is depicted in Figure 3.7.

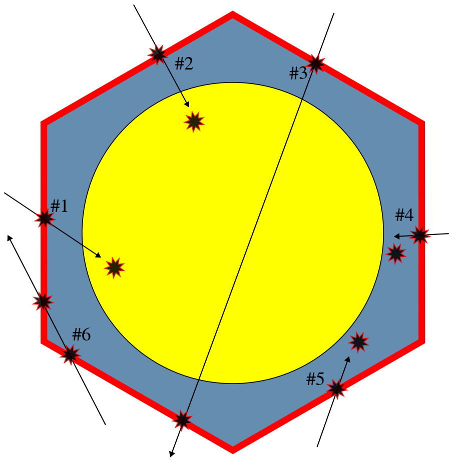


Figure 3.7: Counting of neutrons emitted from the surfaces.

3.5 Solver routines

The hybrid solver includes subroutines to estimate the collision probabilities (CP) for the fine regions inside every subsystem, then the matrices for the interface current method are generated. The eigenvalue problem of the system associated with the static neutron transport equation is solved iteratively using the power iteration method so that the effective multiplication factor and the static fluxes are calculated.

3.5.1 Pre-calculation of the collision probabilities

After the coarse and fine meshes are generated, the coarse nodes that are geometrically equivalent are grouped together into sub-sets. Then, a Serpent model is generated to perform the simulations that lead to the estimation of the cross sections and the collision probabilities for the fine regions of the coarse node and the surfaces. The Serpent model is used to calculate the collision probabilities for the fine regions and surfaces, and the neutron macroscopic cross sections. This process is depicted in Algorithm 1.

3.5.2 Solution of the eigenvalue problem

The algorithm implemented in the Hybrid framework is shown in the diagram of Figure 3.8. Once the input file is provided, the collision probabilities are calculated in a sequential way. The hybrid framework awaits Serpent2 to complete the simulations to then estimate the collision probabilities. The next step is to allocate the matrices for the interface current method, and the initial guess for the multiplication factor and the scalar flux. The transport sweep comprehends an outer iteration also referred to as power iteration (PI) and an inner iteration called energy group iteration. In each outer iteration the vector Q (Eqs. (2.41) and 2.46) is evaluated and used to build the source vectors Φ_{src} (defined by Eq. (2.44)) and $\mathbf{J}_{src}$ (defined by Eq. (2.45)) are calculated for all energy groups, then the neutron current J_{in} and the scalar flux Φ are computed for all energy groups. The effective multiplication factor k_{eff} is updated with Eq. (3.11):

$$k_{eff}^{(n)} = k_{eff}^{(n-1)} \frac{\varphi^{(n-1)} \cdot \varphi^{(n)}}{\varphi^{(n-1)} \cdot \varphi^{(n-1)}}, \tag{3.11}$$

where the superscript n denotes the quantity in the iteration n. The flux values $\varphi^{(n)}$ are obtained by multiplying the vector Φ by the fission source as follows:

$$\varphi^{(n)} = \Phi^{(n)} \times \mathbf{F}, \tag{3.12}$$

with:

$$\mathbf{F} = \begin{bmatrix} f_{j,i,g,g'} & \cdot\cdot & f_{j,i,G,G'} & \cdot\cdot & f_{J,i,g,g'} & \cdot\cdot & f_{J,i,g,G'} \\ \vdots & & \vdots & & \vdots & & \vdots \\ f_{j,i,G,g'} & \cdot\cdot & f_{j,i,G,G'} & \cdot\cdot & f_{J,i,G,g'} & \cdot\cdot & f_{J,i,G,G'} \\ \vdots & & \vdots & & \vdots & & \vdots \\ f_{j,I,g,g'} & \cdot\cdot & f_{j,I,g,G'} & \cdot\cdot & f_{J,I,g,g'} & \cdot\cdot & f_{J,I,g,G'} \\ \vdots & & \vdots & & \vdots & & \vdots \\ f_{j,I,G,g'} & \cdot\cdot & f_{j,I,G,G'} & \cdot\cdot & f_{J,I,G,g'} & \cdot\cdot & f_{J,I,G,G'} \end{bmatrix} \tag{3.13}$$

where every component of the fission source matrix represents the probability for a neutron with energy $g \in G$ born from fission induced by a neutron with energy $g' \in G'$ in region $j \in J$ to have its first interaction in the region $i \in I$ and it is given by:

$$f_{j,i,g,g'} = P_{j,i,g} V_j \chi_{j,g} \nu \Sigma_{f,j,g'} \tag{3.14}$$

Algorithm 1 Collision probabilities generation algorithm.

```
create CoarseMesh()                                   ▷ The coarse mesh is created
for all CoarseNode() ∈ CoarseMesh() do                ▷ Loop over the coarse nodes
    create FineMesh()                                 ▷ Fine regions are created
end for
find coarse_nodes_subsets()                 ▷ Finds geometrically equivalent coarse nodes
for all CoarseNode() ∈ coarse_nodes_subsets() do      ▷ Loop over the sub-sets
    for all region ∈ CoarseNode do                    ▷ Loop over the fine regions
        create SerpentFileRegion          ▷ Serpent model is created for the fine region
        run SerpentFileRegion                         ▷ Serpent model is simulated
        read SerpentFileRegionOutput                  ▷ Serpent model output is read
        calculate ProbabilitiesRegion                 ▷ CP calculation for the fine region
    end for
    create SerpentFileSurfaces              ▷ Serpent model is created for the surfaces
    run SerpentFileSurfaces                           ▷ Serpent model is simulated
    read SerpentFileSurfacesOutput                    ▷ Serpent model output is read
    calculate ProbabilitiesSurfaces                   ▷ CP calculation for the surfaces
end for
for all CoarseNode() ∈ CoarseMesh() do      ▷ Loop over the coarse nodes in the mesh
    replicate ProbabilitiesRegion                     ▷ CP of fine regions are copied
    replicate ProbabilitiesSurfaces                   ▷ CP of surfaces are copied
end for
```

Finally the neutron source vectors Q, $\mathbf{J}_{src}$, and Φ_{src} are updated to be used within the next power iteration. The convergence criterion for the multiplication factor is given by:

$$\epsilon_k = \frac{k_{eff}^{(n)} - k_{eff}^{(n-1)}}{k_{eff}^{(n-1)}}, \tag{3.15}$$

and the convergence criterion for the neutron flux is given by:

$$\epsilon_\Phi = \frac{\max[\Phi^{(n)}] - \max[\Phi^{(n-1)}]}{\max[\Phi^{(n-1)}]}, \tag{3.16}$$

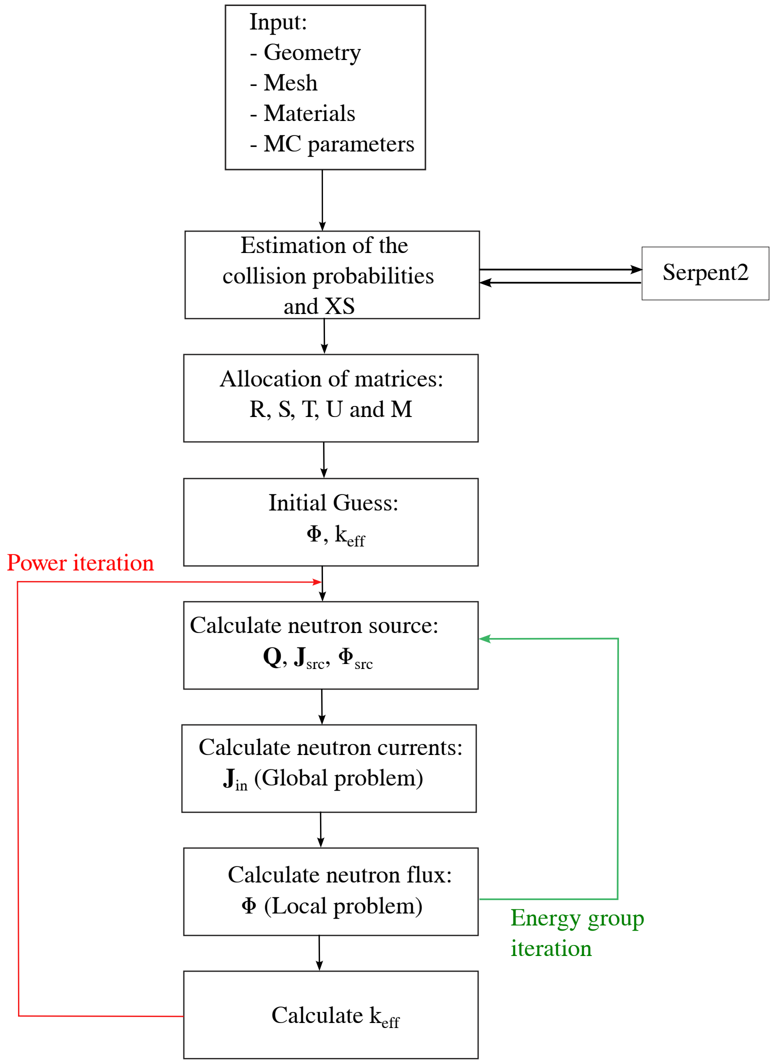

Input:
- Geometry
- Mesh
- Materials
- MC parameters
Estimation of the collision probabilities and XS
Serpent2
Allocation of matrices: R, S, T, U and M
Initial Guess: Φ, k_{eff}
Power iteration
Calculate neutron source: $\mathbf{Q}$, $\mathbf{J}_{src}$, $\mathbf{\Phi}_{src}$
Calculate neutron currents: $\mathbf{J}_{in}$ (Global problem)
Calculate neutron flux: Φ (Local problem)
Energy group iteration
Calculate k_{eff}

Figure 3.8: Chart flow of the algorithm implemented within the hybrid framework.

This process is repeated until convergence is reached, which is when ϵ_k and ϵ_Φ are smaller than a selected tolerance. The transport sweep is depicted in Algorithm 2.

Algorithm 2 Transport sweep algorithm.

```
calculate $\mathbf{M},\mathbf{R},\mathbf{S},\mathbf{T},\mathbf{U}$
Initial guess $\mathbf{\Phi}$, $k_{eff} = 1$
calculate initial Source $Q$
while $\epsilon_k <$ tolerance & $\epsilon_\Phi <$ tolerance do
    for all $g \in G$ do
        solve GlobalProblem                    ▷ Eq. (2.48)
        solve LocalProblem                     ▷ Eq. (2.42)
    end for
    $k_{eff}^{(n)} = k_{eff}^{(n-1)} \frac{\varphi^{(n-1)} \cdot \varphi^{(n)}}{\varphi^{(n-1)} \cdot \varphi^{(n-1)}},$
    update Source $Q$
    calculate $\epsilon_k$ & $\epsilon_\Phi$
end while
```

Chapter 4

Verification of the Hybrid framework

The verification cases used for the hybrid framework are introduced in section 4.1. In section 4.2, the results of the verification over the case of a fuel pin cell are presented. Section 4.3 presents the verification using a scaled fuel assembly. Finally, Section 4.4 presents the verification of the hybrid framework with a full-size hexagonal fuel assembly.

4.1 Verification problems

The hybrid neutron transport framework discussed in Chapter 3 is verified over cases based on data that are derived from the sodium-cooled fast reactor Superphénix (SPX). For the verification of the framework, information was taken from the benchmark for validation of calculation tools that include static and transient data related to the sodium-cooled fast reactor Superphènix (SPX) and its operations [19], [20].

The core of the SPX reactor is divided into four radial zones: inner core, outer core, radial breeder blanket, and steel shielding. The inner core zone is composed of 190 fuel assemblies, 3 diluent steel assemblies, 6 control rods, and 3 shutdown rods. In the periphery between the inner core and the outer core 15 control rods are located along with 15 more diluent steel assemblies. The outer core possesses 168 fuel assemblies, the radial breeder blanket has 222 fuel assemblies, and finally 297 radial steel shielding assemblies. The core composition of the SPX benchmark is depicted in Figure 4.1.

Four 2-D verification cases were built upon the SPX data: a hexagonal fuel pin cell with simplified geometry, a hexagonal fuel pin cell with detailed geometry, a scaled fuel assembly with 7 simplified fuel pins, and a full-sized hexagonal fuel assembly with 271 simplified fuel pins (derived from one fuel assembly included in the SPX inner core, see Figure 4.2). The fuel composition used in the four cases is given in Table 4.1, liquid sodium was used as coolant, stainless steel as cladding, and helium gas in the hollow and the gap regions. The temperature for all materials was set to 453K. A typical 8-energy group grid for a fast system is used and is reported in Table 4.2.

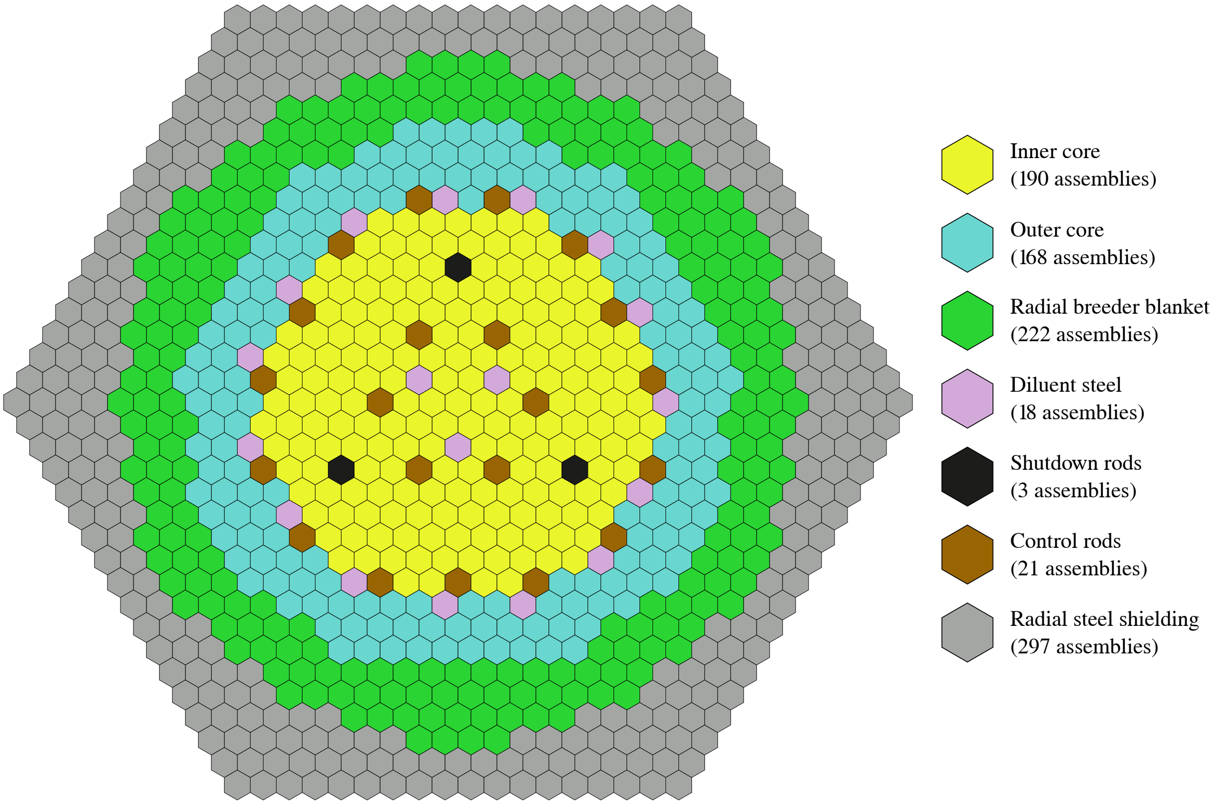
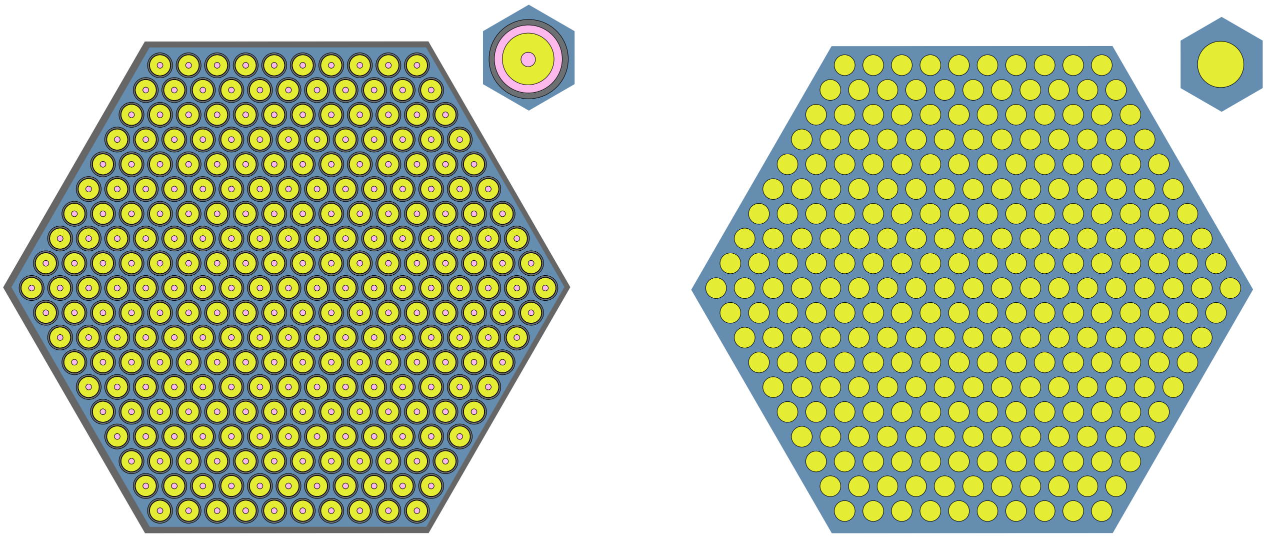


Figure 4.1: Sodium Cooled Fast Reactor SPX core composition.



(a) (b)

Figure 4.2: Fuel assembly with detailed fuel pins and box (a) and simplified fuel assembly (b).

Table 4.1: Isotopic composition of the fuel.

Isotope	Atomic density $[cm^{-3}]$
U^{235}	1.01116E-04
U^{238}	1.98680E-02
Pu^{238}	1.78056E-05
Pu^{239}	2.45592E-03
Pu^{240}	7.32245E-04
Pu^{241}	1.96866E-04
Pu^{242}	6.83261E-05
Am^{241}	4.81830E-05
O^{16}	4.65070E-02

Table 4.2: Energy structure typical of a 3-D core simulation for a SFR.

Energy group	Energy upper limit [eV]
1	2.00E+07
2	2.23E+06
3	8.21E+05
4	3.02E+05
5	1.11E+05
6	4.09E+04
7	1.50E+04
8	7.49E+02

4.2 2-D Hexagonal fuel pin-cell

4.2.1 Simplified geometry

In the first step of the verification, a hexagonal fuel pin cell with simplified geometry is used, see Figure 4.4a. The radius of the fuel pellet is 3.570 mm and the apothem of the hexagon is 4.912 mm. As discussed previously in Chapter 2, the solution of the interface current method requires a coarse and a fine mesh. Three different coarse meshes were tested: a single-node hexagonal mesh that corresponds to the entire hexagonal fuel cell (Figure 4.4a), a 2x2 square grid mesh (Figure 4.4b), and a triangular mesh (Figure 4.4c). In the three cases, the fine mesh of every coarse node consists of two regions: fuel and coolant, respectively. As mentioned in the previous chapter, the ICM requires the precalculation of collision probabilities for the fine mesh.

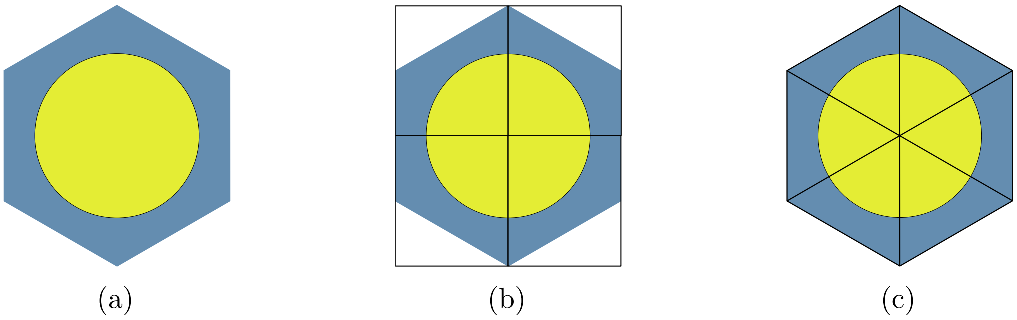


Figure 4.3: Single-node coarse mesh (a), 2x2 square grid coarse mesh (b), and triangular mesh (c) for the simplified fuel pin cell.

Since the collision probabilities are estimated with Monte Carlo, they have an associated uncertainty. The effect of the number of neutron histories selected in the Monte Carlo simulations for the estimation of the collision probabilities is investigated for the single-node, square grid, and triangular coarse meshes. For this purpose, different number of neutron histories (maintaining 200 skip cycles and 2,000 active cycles) are taken: 2E+03, 1E+04, and 1E+05. Serpent is used to calculate the collision probabilities for the eight energy groups for the three different coarse meshes by simulating a coarse node of the corresponding mesh.

For the single-node mesh, the collision probabilities were estimated for the neutrons emitted from the fuel, the coolant, and the six surfaces of the enclosing hexagon (see Figure 4.4a). Reflective boundary conditions are applied to all the faces. For the square-grid mesh, the collision probabilities were estimated within the coarse node illustrated in Figure 4.4b for the neutrons emitted from the fuel, the coolant, and from the surfaces numbered from 1-4 and marked with thicker lines. To model this problem in Serpent, reflective boundary conditions were assumed in the enclosing rectangle and void space was set in the corner (depicted in light gray). 1/4 symmetry is assumed to derive the probabilities in the other coarse nodes. For the triangular mesh, the collision probabilities were estimated within the coarse node illustrated in Figure 4.4c for the neutrons emitted from the fuel, coolant, and the surfaces numbered from 1-3. To model this coarse node, reflective boundary conditions were assumed in the enclosing rectangle and void spaces were set (depicted in light gray). 1/6 symmetry is assumed to derive the probabilities in the other coarse nodes.

The collision probabilities calculated for the square grid, single-node, and triangular coarse meshes are reported in Ref. [21]. The computation time of the estimation of the collision

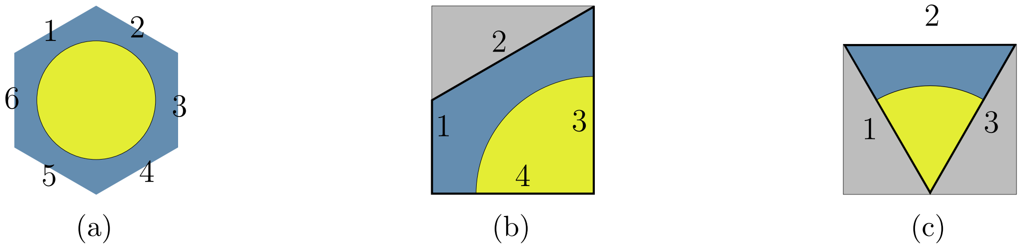


Figure 4.4: Coarse node representation for the single-node, square grid, and triangular coarse meshes respectively.

probabilities with different numbers of neutron histories for the single-node, square grid, and triangular coarse meshes is reported in Table 4.3. In Tables 4.5, 4.4, and 4.6 the relative standard deviation of the probabilities are reported for the three meshes, respectively. Only the highest, intermediate, and lowest energy groups (g=1, 4, and 8) are considered, but they are sufficient for the illustration of the general trend. The values in these tables are related to the calculations with 1E+05 neutron histories. In this problem, the highest uncertainty value corresponds to the estimation of the number of neutrons emitted from the coolant in the thermal energy range (g = 8). The highest standard deviation associated with these Monte Carlo estimations corresponds to the neutrons emitted from the coolant and it ranges from 1.507% to 5.005% for the single-node coarse mesh, from 1.510% to 5.161% for the square grid coarse mesh, and from 1.749% to 5.397% for the triangular coarse mesh.

Table 4.3: Computational time of the estimation of the collision probabilities with different number of neutron histories for the hexagonal fuel pin-cell with simplified geometry.

Neutron histories	CPU-min
Single-node	
2,000	28
10,000	101
100,000	909
Square grid	
2,000	50
10,000	177
100,000	1,635
Triangular mesh	
2,000	72
10,000	279
100,000	2,543

A reference solution value of 1.35600 ± 6.9E-05 was obtained with Serpent2 using 20,000 neutron histories, 500 skip cycles, and 1,500 active cycles that resulted in 19.19 CPU min. The computational effort of the reference solution was smaller than the one required to calculate the collision probabilities even with only 2,000 neutron histories (see Table 4.3). Both for the collision probabilities and the reference solution a supercomputer cluster with a processor Intel® Xeon® Gold 6130 CPU @ 2.10GHz was used.

Table 4.4: Relative standard deviation (%) of the collision probabilities calculated for the single-node hexagonal coarse mesh with 1E+05 neutron histories.

To → **From ↓**	**fuel**	**coolant**	**1**	**2**	**3**	**4**	**5**	**6**
g = 1								
fuel	0.074	0.079	0.046	0.043	0.039	0.034	0.029	0.022
coolant	0.844	0.795	0.482	0.445	0.402	0.354	0.302	0.235
1	0.049	0.051	0.000	0.038	0.032	0.028	0.024	0.019
2	0.049	0.051	0.038	0.000	0.034	0.028	0.023	0.019
3	0.049	0.051	0.035	0.034	0.000	0.029	0.023	0.018
4	0.049	0.051	0.035	0.032	0.030	0.000	0.025	0.019
5	0.049	0.051	0.036	0.032	0.029	0.025	0.000	0.019
6	0.049	0.050	0.038	0.032	0.028	0.024	0.019	0.000
g = 4								
fuel	0.004	0.004	0.003	0.003	0.003	0.002	0.002	0.001
coolant	0.032	0.032	0.021	0.020	0.018	0.015	0.013	0.010
1	0.005	0.005	0.000	0.004	0.003	0.003	0.002	0.002
2	0.005	0.005	0.004	0.000	0.003	0.003	0.002	0.002
3	0.005	0.005	0.004	0.004	0.000	0.003	0.002	0.002
4	0.005	0.005	0.004	0.003	0.003	0.000	0.003	0.002
5	0.005	0.005	0.004	0.003	0.003	0.003	0.000	0.002
6	0.005	0.005	0.004	0.003	0.003	0.002	0.002	0.000
g = 8								
fuel	0.496	0.598	0.411	0.376	0.338	0.295	0.245	0.183
coolant	4.830	5.005	3.261	2.996	2.699	2.364	1.982	1.507
1	0.756	0.866	0.000	0.661	0.565	0.500	0.421	0.324
2	0.756	0.866	0.661	0.000	0.588	0.489	0.413	0.311
3	0.758	0.868	0.633	0.599	0.000	0.518	0.410	0.311
4	0.754	0.864	0.633	0.571	0.523	0.000	0.428	0.308
5	0.761	0.873	0.636	0.580	0.512	0.449	0.000	0.329
6	0.752	0.861	0.659	0.562	0.499	0.420	0.324	0.000

Table 4.5: Relative standard deviation (%) of the collision probabilities calculated for the square grid coarse mesh with 1E+05 neutron histories.

To → **From ↓**	**fuel**	**coolant**	**1**	**2**	**3**	**4**
g = 1						
fuel	0.094	0.097	0.037	0.033	0.034	0.031
coolant	0.970	0.820	0.355	0.376	0.312	0.258
1	0.020	0.018	0.000	0.011	0.010	0.008
2	0.009	0.008	0.006	0.000	0.005	0.004
3	0.009	0.009	0.005	0.005	0.000	0.003
4	0.010	0.010	0.006	0.005	0.004	0.000
g = 4						
fuel	0.0047	0.0049	0.0023	0.0019	0.0018	0.0016
coolant	0.0347	0.0306	0.0148	0.0144	0.0122	0.0099
1	0.0019	0.0018	0.0000	0.0012	0.0009	0.0008
2	0.0009	0.0008	0.0006	0.0000	0.0005	0.0004
3	0.0008	0.0008	0.0005	0.0004	0.0000	0.0003
4	0.0010	0.0010	0.0006	0.0005	0.0004	0.0000
g = 8						
fuel	0.593	0.645	0.319	0.263	0.245	0.215
coolant	5.161	4.843	2.316	2.206	1.874	1.510
1	0.325	0.316	0.000	0.200	0.162	0.141
2	0.147	0.144	0.106	0.000	0.078	0.067
3	0.147	0.150	0.094	0.080	0.000	0.060
4	0.172	0.180	0.108	0.092	0.072	0.000

Table 4.6: Relative standard deviation (%) of the collision probabilities calculated for the triangular coarse mesh with 1E+05 neutron histories.

To → **From ↓**	**fuel**	**coolant**	**1**	**2**	**3**
g = 1					
fuel	0.103	0.105	0.041	0.032	0.039
coolant	1.035	0.841	0.359	0.431	0.324
1	0.005	0.005	0.000	0.003	0.003
2	0.005	0.004	0.003	0.000	0.003
3	0.005	0.005	0.003	0.003	0.000
g = 4					
fuel	0.0050	0.0052	0.0022	0.0017	0.0020
coolant	0.0361	0.0304	0.0134	0.0154	0.0116
1	0.0005	0.0005	0.0000	0.0003	0.0002
2	0.0005	0.0004	0.0003	0.0000	0.0002
3	0.0005	0.0005	0.0003	0.0003	0.0000
g = 8					
fuel	0.629	0.671	0.287	0.238	0.251
coolant	5.397	4.904	2.068	2.345	1.749
1	0.089	0.087	0.000	0.048	0.045
2	0.086	0.080	0.045	0.000	0.044
3	0.089	0.087	0.047	0.046	0.000

After the collision probabilities are estimated with Monte Carlo, the overall problem is solved to determine the effective multiplication factor k_{eff} of the system and the local neutron flux using the deterministic algorithm based on the power iteration method. The selected tolerance for convergence of the multiplication factor and the neutron flux was 1E-10. Two sets of results were generated for every coarse mesh and are denoted as $\mathrm{ICM_{noSP}}$ and $\mathrm{ICM_{SP}}$. The set $\mathrm{ICM_{noSP}}$ corresponds to the case where the scattering production reactions such as (n,2n'), (n,3n'), and (n,4n') were not added to the scattering matrix needed in the deterministic calculations. The second set $\mathrm{ICM_{SP}}$, utilizes scattering matrices including the scattering production neutron reactions mentioned above.

The multiplication factor is shown in Tables 4.7, 4.8, 4.9 for the single-node, square grid, and triangular coarse meshes respectively. The number of iterations to reach convergence in the deterministic calculation is also reported, alongside the wall-clock time of the simulation.

Across all the mesh types, the usage of the scattering production cross sections, corresponding to the results of the set $\mathrm{ICM_{SP}}$, introduces a significant improvement in the prediction of the multiplication factor. The estimation of the collision probabilities with Monte Carlo is the main contributor to the total simulation time, and the deterministic simulation time is negligible. The three different coarse meshes produce very similar results, the discrepancies can be attributed to the geometric representation of the fuel pin and the modeling of each coarse node in Monte Carlo. When the number of neutron histories for the estimation of the collision probabilities is increased, the accuracy and the overall computational effort also does. However, in the case of the single-node and square grid coarse mesh, the accuracy of the estimation of the effective multiplication factor somehow deteriorates, when increasing the neutron population from 1E+04 to 1E+05, from 15pcm to 32 pcm and from 32 pcm to 41 pcm respectively. This behavior is attributed to the variability of Monte Carlo when estimating the collision probabilities.

The scalar neutron flux is calculated for each fine region within the nodes of the coarse meshes using the set of collision probabilities estimated with 1E+05 neutron histories since it has the lowest standard deviation. For the case of the square grid and the triangular coarse mesh, the neutron flux for group g is respectively homogenized over the entire coolant and fuel region using:

$$\phi_g = \frac{\int_V \phi_g(\vec{r})dV}{\int_V dV}. \tag{4.1}$$

The results from the simulations where the scattering production cross sections are added to the scattering matrix, are compared to the reference solution in Figure 4.5 and the relative differences are shown in Figure 4.6. The reference solution is plotted with uncertainty bars having a maximum standard deviation of 0.141% for the coolant and 0.111% for the fuel in energy group 8, while for the rest of the energy groups, the standard deviation is below 0.052%. The scalar flux is plotted in the fuel (F) and coolant (C) regions. The values of the neutron scalar flux are normalized with respect to the maximum value.

In the hexagonal fuel pin the scalar flux is relatively flat, it has very small spatial variations across the system. Relative differences in absolute value with respect to the reference values estimated with Monte Carlo are less than 0.13% except in the energy group g=8 with values of up to 0.31%. The buckling of the neutron flux is reproduced correctly for all energy groups.

Table 4.7: Effective multiplication factor for the simplified 2-D fuel pin (hexagonal coarse mesh).

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
ICM_{noSP}				
2,000	1.35174	426	605	0.191
10,000	1.35223	377		0.184
100,000	1.35207	393		0.182
ICM_{SP}				
2,000	1.35541	59	606	0.187
10,000	1.35585	15		0.188
100,000	1.35568	32		0.182

Table 4.8: Effective multiplication factor for the simplified 2-D fuel pin (square grid coarse mesh).

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
ICM_{noSP}				
2,000	1.35123	477	602	0.369
10,000	1.35207	393		0.366
100,000	1.35198	402		0.367
ICM_{SP}				
2,000	1.35483	117	603	0.366
10,000	1.35568	32		0.374
100,000	1.35559	41		0.369

Table 4.9: Effective multiplication factor for the simplified 2-D fuel pin (triangular coarse mesh).

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
ICM$_{\text{noSP}}$				
2,000	1.35139	461	605	0.480
10,000	1.35168	432		0.476
100,000	1.35187	413		0.466
ICM$_{\text{SP}}$				
2,000	1.35501	99	606	0.459
10,000	1.35531	69		0.470
100,000	1.35548	52		0.481

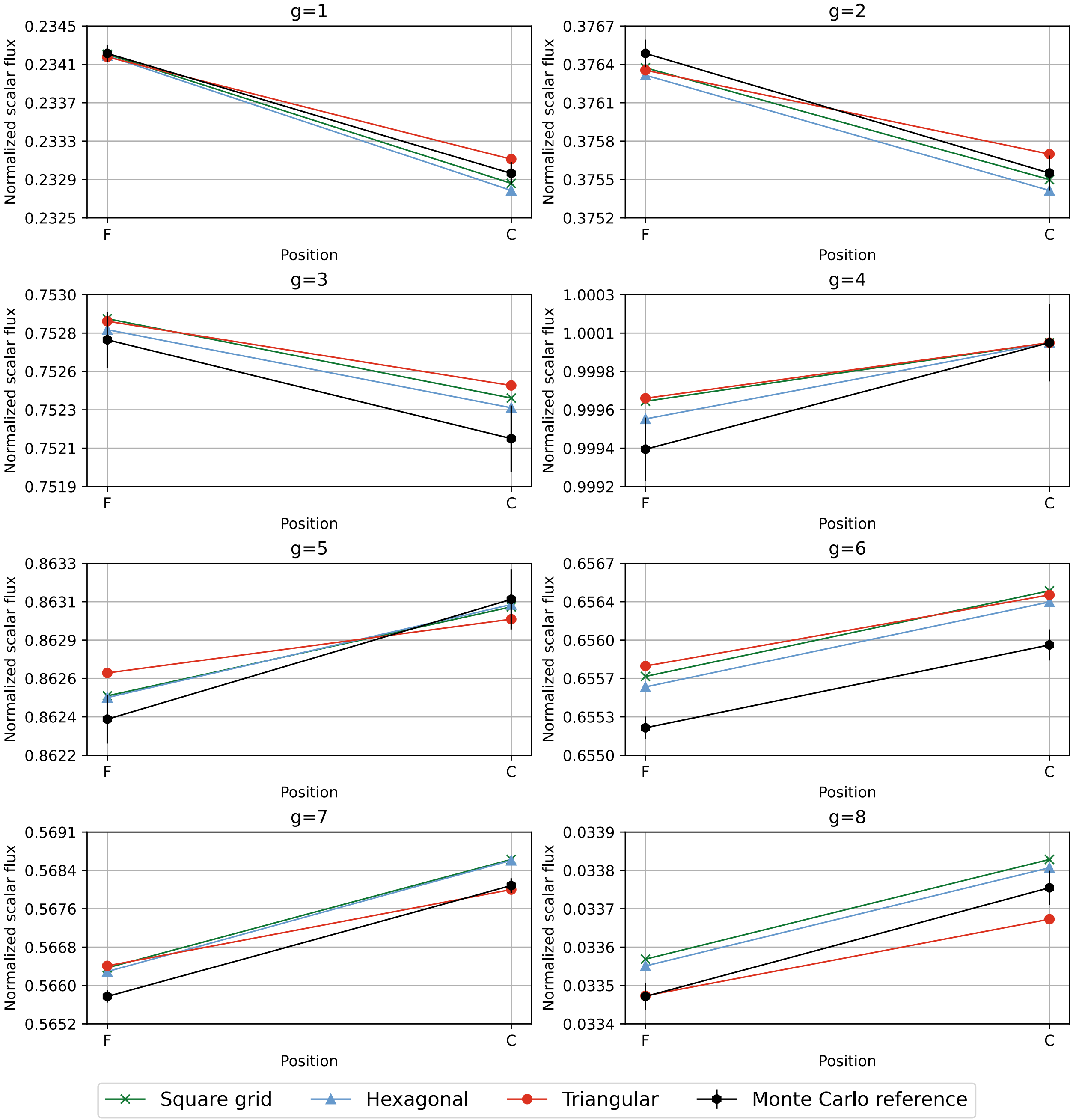


Figure 4.5: Scalar flux for the simplified hexagonal fuel pin cell.

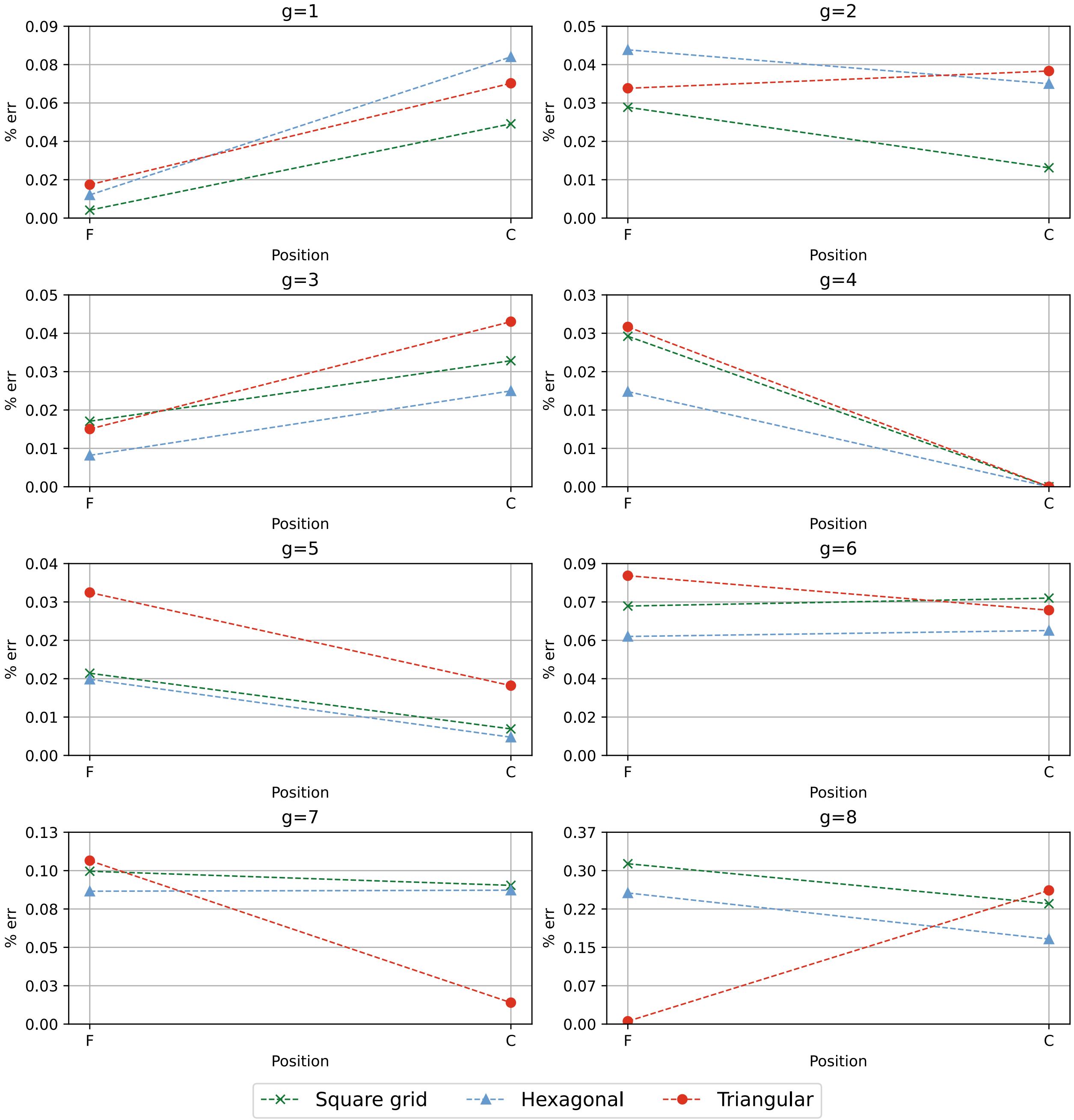


Figure 4.6: Relative difference with respect to the reference neutron flux estimated with Monte Carlo for the simplified fuel pin cell.

4.2.2 Detailed geometry

In the second step of the verification, a hexagonal fuel pin cell with detailed geometry was simulated. It consists of a fuel pin (F) with a helium hollow (H), helium gap (HG), and cladding (Z) surrounded by coolant (C). Three different coarse meshes were tested: a single-node hexagonal mesh that corresponds to the entire hexagonal fuel cell (Figure 4.7a), a 2x2 square grid mesh (Figure 4.7b), and a triangular mesh (Figure 4.7c). The fine mesh consists of the regions characterized by the different materials in each section of the cell, i.e., helium hollow, fuel, gap, cladding, and coolant.

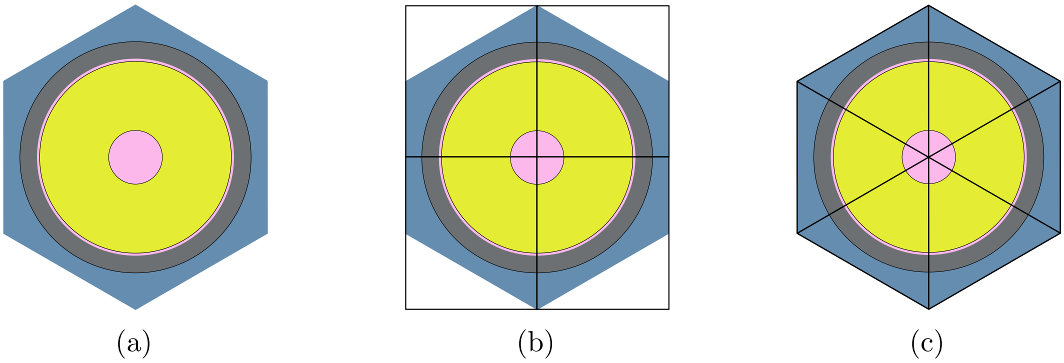


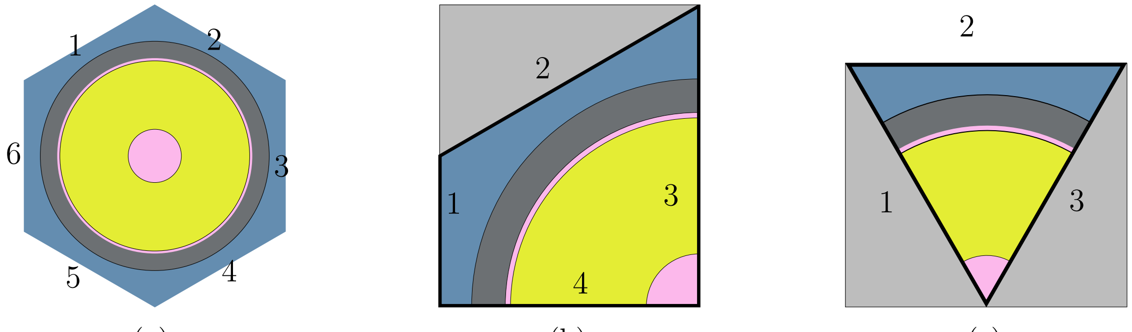
Figure 4.7: Hexagonal fuel pin with detailed geometry.

The effect of the number of neutron histories on the accuracy of the Monte Carlo calculations of the collision probabilities is investigated for the single-node, square grid, and triangular coarse meshes. As in the previous section, the number of neutron histories was selected to be 2E+03, 1E+04, and 1E+05 and the active and skip cycles were set to 2,000 and 200 respectively. Serpent is used to calculate the collision probabilities for the eight energy groups for the neutrons emitted from every region and the surfaces.

For the single-node coarse mesh the collision probabilities were estimated for the neutrons emitted from the helium hollow, fuel, gap, cladding, coolant, and the six surfaces of the enclosing hexagon (see Figure 4.8a). Reflective boundary conditions were applied to all the faces. Symmetry of 1/4 is assumed to derive the probabilities in the other coarse nodes of the mesh.

For the square-grid mesh, the collision probabilities were estimated within the coarse node illustrated in Figure 4.8b. To model this problem in Serpent, reflective boundary conditions were assumed on the enclosing rectangle and void space was set in the corner (depicted in light gray). The probabilities were calculated for the neutrons emitted from the hollow, fuel, gap, cladding, and coolant, and from the surfaces marked with thicker lines and numbered from 1 to 4.

For the triangular mesh, the collision probabilities were estimated within the coarse node illustrated in Figure 4.8c for the neutrons emitted from the fuel, coolant, and the surfaces numbered from 1 to 3. Reflective boundary conditions were assumed in the enclosing rectangle and void spaces were set (depicted in light gray) to model this coarse node. Symmetry of 1/6 is assumed to derive the probabilities in the other coarse nodes. The collision probabilities were estimated for the neutrons emitted from the hollow, fuel, gap, cladding, coolant, and the three surfaces.



(a) (b) (c)

Figure 4.8: Hexagonal fuel pin with detailed geometry.

The estimated values of the collision probabilities for the single-node, square grid, and triangular coarse meshes are reported in Ref. [21]. The computation time of the estimation of the collision probabilities with different numbers of neutron histories for the single-node, square grid, and triangular coarse meshes is reported in Table 4.10. The relative standard deviation of the probabilities are calculated and reported for the energy groups 1, 4, and 8 in Tables 4.11, 4.12, and 4.13 for the single-node hexagonal coarse mesh and in Tables 4.14 and 4.15 for the square grid and triangular coarse meshes respectively. These results were generated with 1E+05 neutron histories.

The probabilities with the highest statistical uncertainty are related to the number of neutrons emitted from the helium regions (both hollow and gap) in the three meshes and all energy groups. This is expected because neutrons have very few collisions in these regions (the reaction rates are in the order of 10^{-6}) and the probability for neutrons emitted from anywhere in the cell and then interacting in a helium region is very low (in the order of 10^{-5}). Then, a very large number of neutron histories is required to obtain good accuracy. The values for the standard deviation of these probabilities range from 32.326% to higher than 100% for group 1, from 2.505% up to 39.396% for group 4, and higher than 100% for the energy group 8.

The reference value of the effective multiplication factor is 1.26450 ± 6.9E-05 and it was obtained with Serpent2 using 100,000 neutron histories, 500 skip cycles, and 1,500 active cycles that resulted in 115 CPU min, which needed less computational effort than the ones required to estimate the collision probabilities with 2E+03, 1E+04, and 1E+05 neutron histories. Both for the collision probabilities and the reference solution, a supercomputer cluster with a processor Intel® Xeon® Gold 6130 CPU @ 2.10GHz was used.

Table 4.10: Computational time of the estimation of the collision probabilities with different number of neutron histories for the hexagonal fuel pin-cell with detailed geometry.

Neutron histories	CPU-min
Single-node	
2,000	193
10,000	586
100,000	5,391
Square grid	
2,000	99
10,000	375
100,000	3,142
Triangular mesh	
2,000	150
10,000	560
100,000	5,121

Table 4.11: Relative standard deviation (%) of the collision probabilities calculated for the detailed pin with single-node hexagonal coarse mesh using 1E+05 neutron histories (Energy group g = 1).

To → **From ↓**	**H**	**F**	**HG**	**Z**	**C**	**1**	**2**	**3**	**4**	**5**	**6**
g = 1											
H	>100	>100	>100	77.704	71.679	41.173	39.572	37.927	36.195	34.355	32.326
F	0.385	0.280	0.308	0.136	0.126	0.079	0.076	0.072	0.068	0.064	0.060
HG	>100	>100	>100	80.365	76.221	50.339	48.545	46.712	44.912	42.801	40.729
Z	2.680	1.707	1.787	0.726	0.689	0.513	0.496	0.478	0.459	0.440	0.420
C	13.973	9.452	9.781	3.097	2.741	2.529	2.471	2.408	2.343	2.275	2.207
1	0.237	0.163	0.178	0.075	0.071	0.000	0.055	0.059	0.059	0.053	0.036
2	0.238	0.164	0.179	0.076	0.071	0.055	0.000	0.051	0.056	0.056	0.049
3	0.236	0.163	0.178	0.075	0.071	0.062	0.051	0.000	0.046	0.052	0.053
4	0.238	0.164	0.179	0.076	0.072	0.065	0.059	0.048	0.000	0.042	0.049
5	0.237	0.165	0.180	0.075	0.071	0.062	0.062	0.056	0.042	0.000	0.036
6	0.238	0.164	0.179	0.075	0.072	0.055	0.059	0.059	0.052	0.037	0.000

Table 4.12: Relative standard deviation (%) of the collision probabilities calculated for the detailed pin with single-node hexagonal coarse mesh using 1E+05 neutron histories (Energy group g = 4).

To → **From ↓**	**H**	**F**	**HG**	**Z**	**C**	**1**	**2**	**3**	**4**	**5**	**6**
g = 4											
H	17.066	13.792	16.244	5.684	5.139	3.299	3.153	3.003	2.847	2.677	2.505
F	0.018	0.013	0.015	0.006	0.006	0.004	0.004	0.004	0.003	0.003	0.003
HG	20.729	11.259	12.358	5.666	5.291	3.920	3.772	3.613	3.451	3.277	3.103
Z	0.107	0.065	0.070	0.026	0.025	0.020	0.019	0.019	0.018	0.017	0.016
C	0.409	0.265	0.281	0.082	0.073	0.073	0.071	0.069	0.067	0.065	0.063
1	0.019	0.012	0.014	0.006	0.005	0.000	0.004	0.004	0.004	0.004	0.003
2	0.019	0.012	0.014	0.006	0.005	0.004	0.000	0.004	0.004	0.004	0.004
3	0.019	0.012	0.014	0.006	0.005	0.005	0.004	0.000	0.004	0.004	0.004
4	0.019	0.012	0.014	0.006	0.005	0.005	0.004	0.004	0.000	0.003	0.004
5	0.019	0.012	0.014	0.006	0.005	0.005	0.005	0.004	0.003	0.000	0.003
6	0.019	0.012	0.014	0.006	0.005	0.004	0.004	0.004	0.004	0.003	0.000

Table 4.13: Relative standard deviation (%) of the collision probabilities calculated for the detailed pin with single-node hexagonal coarse mesh using 1E+05 neutron histories (Energy group g = 8).

To → **From ↓**	**H**	**F**	**HG**	**Z**	**C**	**1**	**2**	**3**	**4**	**5**	**6**
g = 8											
H	>100	>100	>100	>100	>100	>100	>100	>100	>100	>100	>100
F	1.739	1.129	1.401	0.577	0.579	0.388	0.364	0.337	0.310	0.280	0.244
HG	>100	>100	>100	>100	>100	>100	>100	>100	>100	>100	>100
Z	2.829	1.690	1.866	0.779	0.753	0.569	0.539	0.507	0.474	0.437	0.397
C	54.994	35.110	37.863	10.425	9.720	9.314	9.060	8.789	8.513	8.226	7.933
1	2.577	1.624	1.919	0.735	0.709	0.000	0.568	0.552	0.529	0.469	0.376
2	2.574	1.620	1.915	0.735	0.709	0.568	0.000	0.525	0.511	0.486	0.419
3	2.584	1.629	1.925	0.733	0.706	0.588	0.524	0.000	0.480	0.467	0.442
4	2.567	1.617	1.910	0.733	0.709	0.602	0.553	0.482	0.000	0.431	0.419
5	2.568	1.623	1.919	0.735	0.709	0.590	0.566	0.513	0.433	0.000	0.375
6	2.573	1.622	1.918	0.734	0.707	0.566	0.550	0.527	0.469	0.375	0.000

Table 4.14: Relative standard deviation (%) of the collision probabilities calculated for the detailed pin with square grid coarse mesh using 1E+05 neutron histories.

To → **From ↓**	**H**	**F**	**HG**	**Z**	**C**	**1**	**2**	**3**	**4**
g = 1									
H	>100	>100	>100	>100	>100	37.911	61.179	>100	>100
F	0.580	0.410	0.432	0.186	0.171	0.081	0.134	0.174	0.168
HG	>100	>100	>100	99.744	93.779	68.508	>100	>100	>100
Z	4.330	2.193	2.223	0.840	0.785	0.734	1.305	1.089	0.988
C	22.826	13.030	13.132	3.471	2.867	4.681	7.874	5.247	4.207
1	0.134	0.083	0.085	0.030	0.026	0.000	0.026	0.054	0.047
2	0.049	0.031	0.032	0.013	0.011	0.008	0.000	0.023	0.018
3	0.042	0.032	0.033	0.014	0.013	0.010	0.020	0.000	0.011
4	0.049	0.038	0.040	0.017	0.015	0.012	0.021	0.014	0.000
g = 4									
H	30.293	27.887	29.367	9.743	8.464	2.988	4.697	10.828	10.770
F	0.027	0.018	0.020	0.008	0.007	0.004	0.006	0.008	0.007
HG	33.849	13.530	14.090	6.806	6.261	5.219	9.344	8.651	8.027
Z	0.171	0.082	0.084	0.029	0.026	0.029	0.051	0.041	0.038
C	0.668	0.367	0.375	0.088	0.072	0.136	0.230	0.149	0.118
1	0.011	0.006	0.007	0.002	0.002	0.000	0.002	0.004	0.004
2	0.004	0.002	0.002	0.001	0.001	0.001	0.000	0.002	0.001
3	0.003	0.002	0.003	0.001	0.001	0.001	0.002	0.000	0.001
4	0.004	0.003	0.003	0.001	0.001	0.001	0.002	0.001	0.000
g = 8									
H	>100	>100	>100	>100	>100	>100	>100	>100	>100
F	2.536	1.648	1.868	0.738	0.725	0.357	0.528	0.699	0.672
HG	>100	>100	>100	>100	>100	>100	>100	>100	>100
Z	4.483	2.119	2.219	0.862	0.779	0.729	1.258	0.986	0.891
C	91.471	48.437	50.035	10.987	9.685	18.464	31.385	18.785	14.691
1	1.447	0.848	0.905	0.285	0.258	0.000	0.280	0.490	0.480
2	0.526	0.314	0.334	0.121	0.113	0.086	0.000	0.232	0.170
3	0.434	0.319	0.350	0.132	0.127	0.097	0.193	0.000	0.104
4	0.510	0.373	0.415	0.160	0.155	0.120	0.192	0.138	0.000

Table 4.15: Relative standard deviation (%) of the collision probabilities calculated for the detailed pin with triangular coarse mesh using 1E+05 neutron histories.

To → **From ↓**	**H**	**F**	**HG**	**Z**	**C**	**1**	**2**	**3**
g = 1								
H	>100	>100	>100	>100	>100	>100	73.807	>100
F	0.659	0.458	0.478	0.211	0.193	0.230	0.165	0.230
HG	>100	>100	>100	>100	>100	>100	>100	>100
Z	5.090	2.360	2.377	0.897	0.838	1.513	1.810	1.510
C	26.735	14.470	14.500	3.677	2.989	7.249	11.377	7.230
1	0.026	0.020	0.020	0.008	0.007	0.000	0.012	0.012
2	0.031	0.019	0.020	0.007	0.006	0.015	0.000	0.015
3	0.026	0.020	0.020	0.008	0.007	0.012	0.012	0.000
g = 4								
H	36.257	34.029	35.287	11.635	10.044	14.596	5.611	14.573
F	0.030	0.020	0.022	0.008	0.008	0.010	0.007	0.010
HG	39.396	14.257	14.689	7.363	6.794	12.007	12.491	11.981
Z	0.201	0.089	0.091	0.030	0.028	0.058	0.071	0.058
C	0.784	0.407	0.413	0.092	0.073	0.207	0.332	0.207
1	0.002	0.001	0.002	0.001	0.001	0.000	0.001	0.001
2	0.002	0.001	0.002	0.001	0.000	0.001	0.000	0.001
3	0.002	0.001	0.002	0.001	0.001	0.001	0.001	0.000
g = 8								
H	>100	>100	>100	>100	>100	>100	>100	>100
F	2.896	1.851	2.059	0.830	0.813	0.939	0.641	0.932
HG	>100	>100	>100	>100	>100	>100	>100	>100
Z	5.341	2.306	2.390	0.923	0.820	1.411	1.786	1.400
C	>100	53.744	55.021	11.361	9.902	26.851	45.694	26.793
1	0.270	0.199	0.213	0.078	0.073	0.000	0.119	0.117
2	0.343	0.200	0.208	0.070	0.063	0.151	0.000	0.151
3	0.269	0.198	0.212	0.078	0.073	0.116	0.118	0.000

The probabilities generated with Monte Carlo are used for the deterministic solution of the overall problem (given by Eqs. (2.39) and (2.40)). As a result, k_{eff} and the spatial distribution of the neutron flux are calculated. The convergence tolerance for the multiplication factor and the neutron flux was selected equal to 1E-10. Two sets of results were generated for every coarse mesh and are denoted as ICM$_{\text{noSP}}$ (only scattering reactions (n,n')) and ICM$_{\text{SP}}$ (production scattering reactions included). The multiplication factor is shown for the two sets of results in Tables 4.16, 4.17, and 4.18 for the single-node, square grid, and triangular coarse meshes, respectively. Additionally, the number of iterations to reach convergence and the wall-clock time of the deterministic simulation are reported.

Table 4.16: Effective multiplication factor calculation for the detailed 2-D hexagonal fuel pin-cell (single-node hexagonal coarse mesh).

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
ICM$_{\text{noSP}}$				
2,000	1.26112	338	746	0.333
10,000	1.26116	334		0.341
100,000	1.26178	272		0.330
ICM$_{\text{SP}}$				
2,000	1.26376	74	748	0.327
10,000	1.26375	75		0.370
100,000	1.26380	70		0.336

Table 4.17: Effective multiplication factor calculation for the detailed 2-D hexagonal fuel pin-cell (square grid coarse mesh).

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
ICM$_{\text{noSP}}$				
2,000	1.26162	288	735	0.762
10,000	1.26186	264		0.794
100,000	1.26175	275		0.771
ICM$_{\text{SP}}$				
2,000	1.26422	28	736	0.745
10,000	1.26445	5		0.852
100,000	1.26435	15		0.792

Table 4.18: Effective multiplication factor calculation for the detailed 2-D hexagonal fuel pin-cell (triangular coarse mesh).

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
		ICM_{noSP}		
2,000	1.26156	294	745	1.093
10,000	1.26122	328		1.095
100,000	1.26139	311		1.088
		ICM_{SP}		
2,000	1.26413	37	746	1.097
10,000	1.26381	69		1.060
100,000	1.26399	51		1.061

As seen in the case of the hexagonal pin with simplified geometry, the usage of the scattering production cross sections (ICM_{SP}) significantly improves the prediction of the effective multiplication factor of the hexagonal pin with the detailed geometry. The wall-clock time duration of the deterministic simulation is negligible compared with the computational time for the collision probabilities. The three different coarse meshes produce similar results. However, the estimation from the triangular mesh is slightly less accurate. This can be due to the geometric representation and modeling of the triangular coarse node in Monte Carlo. In the case of the squared grid coarse mesh, when increasing the neutron population from 1E+04 to 1E+05, the accuracy on the effective multiplication factor deteriorates (the difference with respect to the reference goes from 5 to 15 pcm), while the computational effort is almost ten-folded. Future work is needed to propagate the Monte Carlo uncertainty of the probabilities to the estimation of the effective multiplication factor.

The scalar flux is calculated in the fine regions within the nodes of the coarse meshes. For the case of the square grid and the triangular coarse mesh, the scalar fluxes are homogenized in the different regions using Eq. (4.1). The scalar flux for the detailed hexagonal fuel pin-cell is plotted in Figure 4.9 for the three meshes and only for the ICM_{SP} set. The reference solution is plotted with uncertainty bars in Figure 4.9, having a maximum standard deviation of 0.175% for the helium gap and of 0.158% in the helium hollow, both in the energy group 8. The relative differences to the reference solution are plotted in Figure 4.10. The neutron scalar flux and the relative differences are plotted for the helium hollow (H), fuel (F), helium gap (HG), cladding (Z), and coolant (C). The values of the neutron scalar flux are normalized with respect to the maximum value.

The flux is still relatively flat but the spatial variations are a little bigger than in the previous case. The scalar flux calculated with the hybrid framework differs from the Monte Carlo reference solution less than 0.7% for all energy groups. The largest discrepancies are found in the helium regions. This may be due to the accuracy of the associated probabilities, whose Monte Carlo estimation has a statistical error higher than 100%.

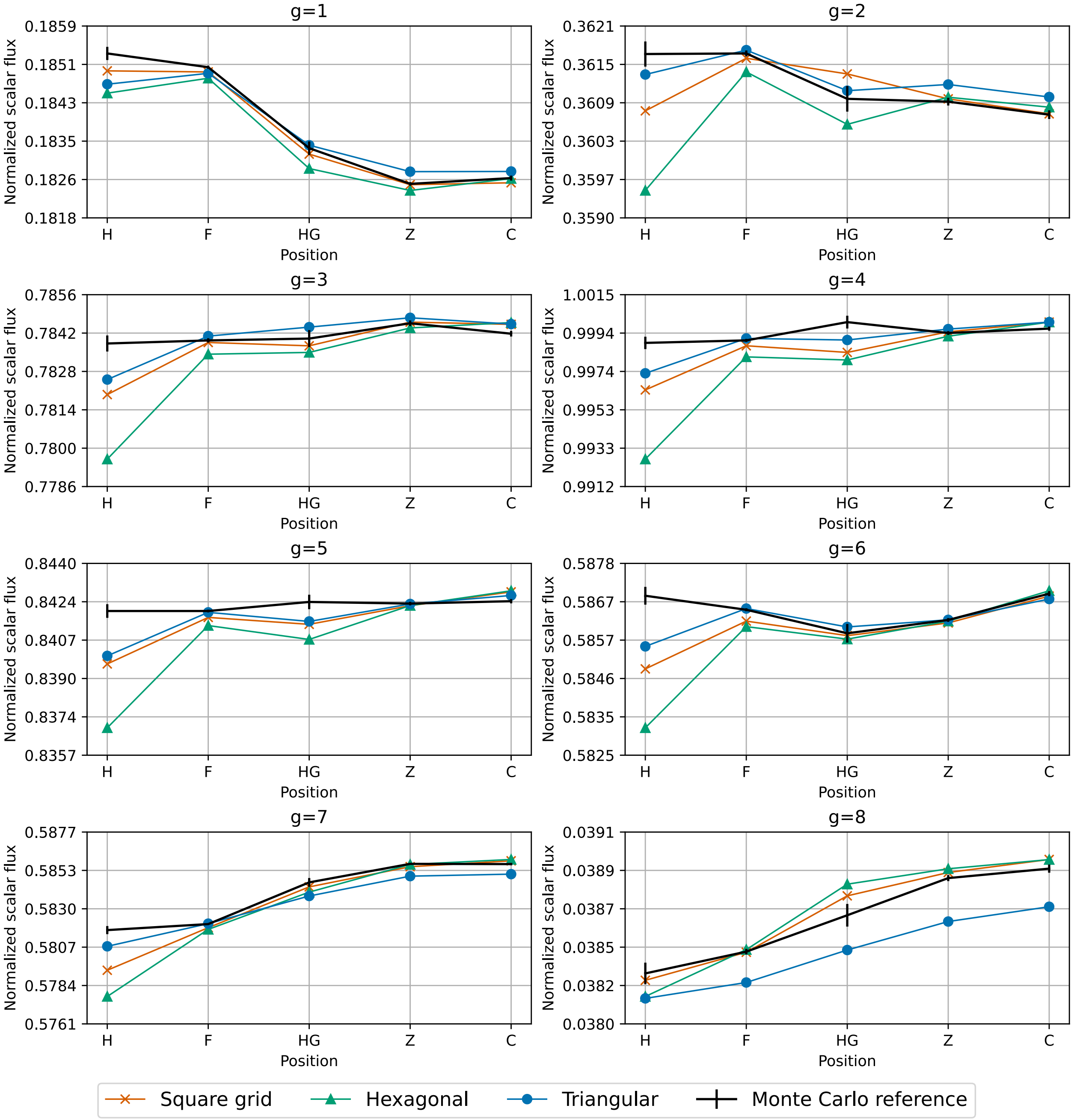


Figure 4.9: Scalar flux for the detailed hexagonal fuel pin cell.

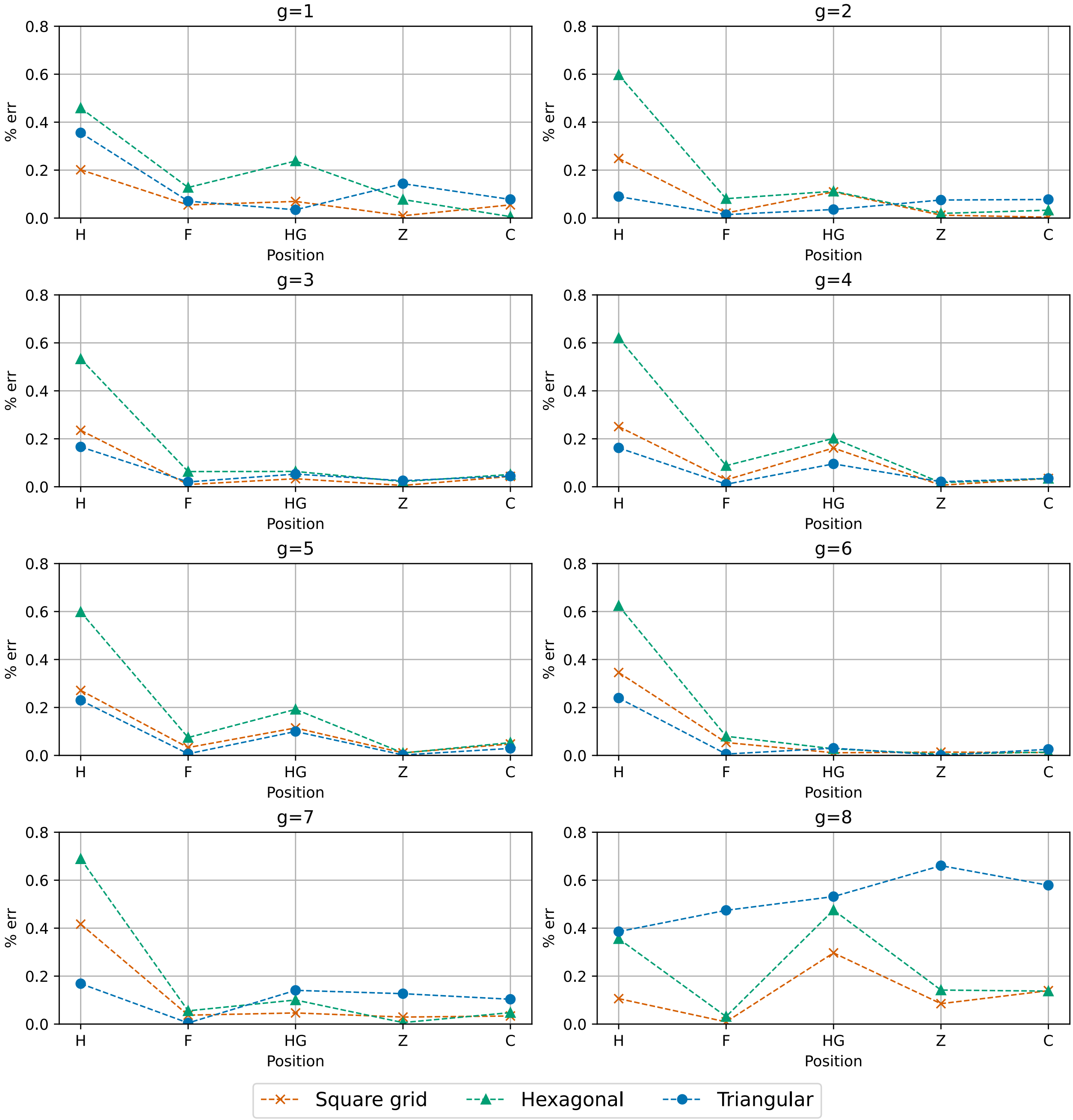


Figure 4.10: Relative difference with respect to the reference neutron flux estimated with Monte Carlo in the detailed fuel pin cell.

4.3 Scaled hexagonal fuel assembly

The second test case consists of a hexagonal lattice of 7 fuel pins with simplified geometry. The system is discretized according to a coarse mesh that includes both triangular and square nodes, see Figure 4.11. Three different types of coarse nodes, whose orientation may vary, namely an inner triangular subchannel (Figure 4.12a), a square subchannel (Figure 4.12b), and an outer triangular subchannel (Figure 4.12c). Then, the global problem has a total of 24 coarse nodes, 66 fine regions, and 78 surfaces.

To study the accuracy of the Monte Carlo estimation of the collision probabilities, different numbers of neutron histories are taken (maintaining 200 skip cycles and 2000 active cycles), i.e., 2E+03, 1E+04, and 1E+05. Serpent is used to calculate the collision probabilities for the eight energy groups for the three different coarse nodes.

For the inner triangular subchannel, the collision probabilities were estimated for the neutrons emitted from the three fuel regions, the coolant, and the three surfaces numbered 1-3 of the enclosing triangle (see Figure 4.12a). Reflective boundary conditions are applied to the enclosing square. For the square subchannel, the collision probabilities were estimated for the neutrons emitted from the two fuel regions, the coolant, and from the surfaces numbered from 1 to 4 and marked with thicker lines (see Figure 4.12b). To model this problem in Serpent, reflective boundary conditions were assumed in the enclosing rectangle. For the outer triangular subchannel, the collision probabilities were estimated for the neutrons emitted from the fuel, coolant, and the surfaces numbered (see Figure 4.12c). To model this coarse node, reflective boundary conditions were assumed on the enclosing rectangle, and void space was set (depicted in light gray).

The collision probabilities calculated for the three coarse nodes are reported in Ref. [21]. The computation time of the estimation of the collision probabilities with different numbers of neutron histories for the three types of coarse nodes is reported in Table 4.19.

Table 4.19: Computational time of the estimation of the collision probabilities with different number of neutron histories for the coarse mesh applied to the scaled hexagonal fuel assembly.

Neutron histories	CPU-min
2,000	252
10,000	938
100,000	9,359

The reference value of the effective multiplication factor is 1.33257 ± 6.8E-06 and it was obtained with Serpent2 using 1E+06 neutron histories, 3,000 active cycles, and 1,500 inactive cycles that resulted in 1,867 CPU min. In this case, the reference solution was more expensive than estimating the collision probabilities with 2E+03 and 1E+04 neutron histories. Both for the collision probabilities and the reference solution a supercomputer cluster with a processor Intel® Xeon® Gold 6130 CPU @ 2.10GHz was used.

The relative standard deviation of the probabilities are calculated and reported for the energy groups 1,4, and 8 in Tables 4.20, 4.21, and 4.22 for the inner triangular subchannel, rectangular subchannel, and the outer triangular subchannel respectively. These results were generated with 1E+05 neutron histories. In this problem, the highest uncertainty value corresponds to the estimation of the number of neutrons emitted from the coolant in the thermal energy range (g = 8). The standard deviation associated with these Monte Carlo estimations ranges from 2.16% to 6.65% for the inner triangle, from 0.45% to 1.46% for the rectangular subchannel, and from 0.54% to 1.64% for the outer triangle.

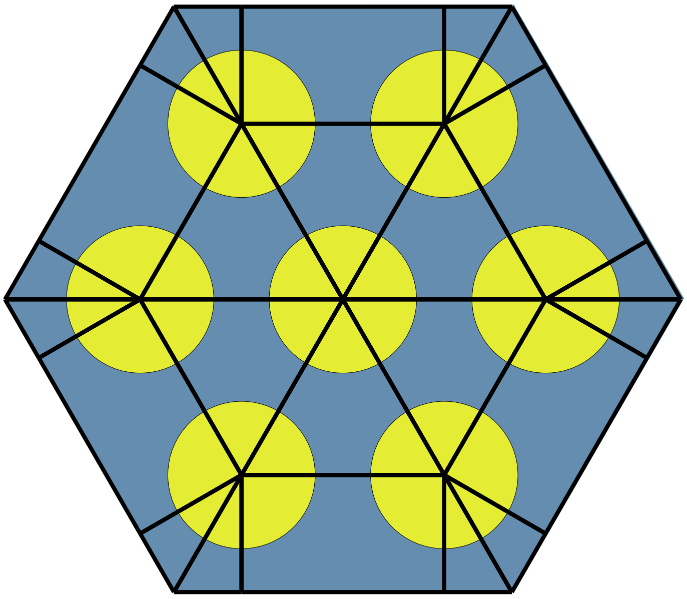


Figure 4.11: 2-D scaled hexagonal fuel assembly with coarse mesh (thick lines) and fine mesh (thin line within the coarse nodes).

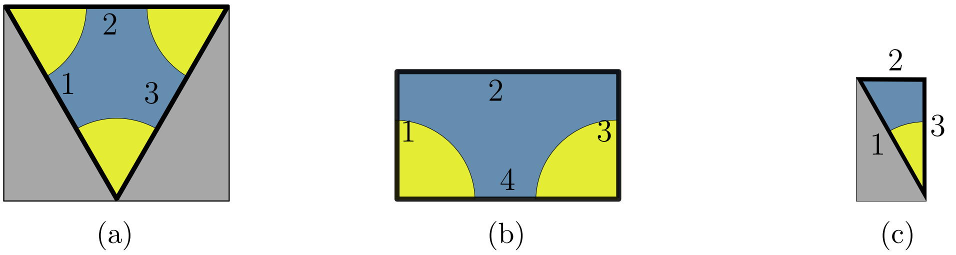


Figure 4.12: Type of coarse nodes obtained from the triangular mesh applied to the hexagonal assembly.

Table 4.20: Relative standard deviation (%) of the collision probabilities calculated for the scaled hexagonal fuel assembly with triangular coarse mesh in the inner triangular subchannel using 1E+05 neutron histories.

To → **From ↓**	**fuel-1**	**fuel-2**	**fuel-3**	**coolant**	**1**	**2**	**3**
g = 1							
fuel-1	0.74	0.76	0.61	0.39	0.30	0.17	0.29
fuel-2	0.77	0.59	0.61	0.39	0.30	0.30	0.16
fuel-3	0.77	0.62	0.39	0.39	0.18	0.29	0.29
coolant	1.47	1.26	1.00	0.65	0.50	0.49	0.48
1	0.01	0.01	0.01	0.01	0.00	0.01	0.01
2	0.01	0.01	0.01	0.01	0.01	0.00	0.01
3	0.01	0.01	0.01	0.01	0.01	0.01	0.00
g = 4							
fuel-1	0.034	0.036	0.029	0.019	0.014	0.008	0.013
fuel-2	0.036	0.027	0.029	0.019	0.014	0.014	0.008
fuel-3	0.036	0.029	0.019	0.019	0.009	0.014	0.013
coolant	0.050	0.043	0.035	0.024	0.017	0.017	0.016
1	0.001	0.001	0.001	0.001	0.000	0.001	0.001
2	0.001	0.001	0.001	0.001	0.001	0.000	0.001
3	0.001	0.001	0.001	0.001	0.001	0.001	0.000
g = 8							
fuel-1	3.59	3.93	3.29	2.47	1.53	1.00	1.44
fuel-2	4.01	3.04	3.29	2.48	1.54	1.50	0.88
fuel-3	4.00	3.37	2.35	2.49	1.12	1.48	1.44
coolant	6.65	5.86	4.94	3.85	2.41	2.29	2.16
1	0.20	0.19	0.18	0.15	0.00	0.10	0.10
2	0.21	0.18	0.16	0.14	0.10	0.00	0.10
3	0.20	0.19	0.16	0.15	0.10	0.10	0.00

Table 4.21: Relative standard deviation (%) of the collision probabilities calculated for the scaled hexagonal fuel assembly with triangular coarse mesh in the rectangular subchannel using 1E+05 neutron histories.

To → **From ↓**	**fuel-1**	**fuel-2**	**coolant**	**1**	**2**	**3**	**4**
g = 1							
fuel-1	0.32	0.33	0.22	0.12	0.11	0.09	0.11
fuel-2	0.33	0.22	0.22	0.11	0.10	0.11	0.11
coolant	0.65	0.53	0.37	0.22	0.25	0.17	0.19
1	0.03	0.03	0.02	0.00	0.01	0.01	0.01
2	0.01	0.01	0.01	0.01	0.00	0.01	0.01
3	0.03	0.02	0.02	0.01	0.01	0.00	0.01
4	0.01	0.01	0.01	0.01	0.01	0.00	0.00
g = 4							
fuel-1	0.016	0.017	0.012	0.007	0.006	0.006	0.006
fuel-2	0.017	0.012	0.012	0.007	0.006	0.006	0.006
coolant	0.023	0.019	0.014	0.009	0.009	0.007	0.007
1	0.002	0.002	0.002	0.000	0.001	0.001	0.001
2	0.001	0.001	0.001	0.001	0.000	0.001	0.001
3	0.003	0.002	0.002	0.001	0.001	0.000	0.001
4	0.001	0.001	0.001	0.001	0.001	0.000	0.000
g = 8							
fuel-1	0.81	0.90	0.70	0.39	0.36	0.32	0.30
fuel-2	0.92	0.65	0.70	0.41	0.32	0.31	0.30
coolant	1.46	1.27	1.05	0.65	0.62	0.50	0.45
1	0.18	0.18	0.15	0.00	0.10	0.09	0.07
2	0.09	0.08	0.07	0.05	0.00	0.04	0.05
3	0.19	0.15	0.15	0.10	0.08	0.00	0.07
4	0.09	0.08	0.08	0.05	0.05	0.03	0.00

Table 4.22: Relative standard deviation (%) of the collision probabilities calculated for the scaled hexagonal fuel assembly with triangular coarse mesh in the outer triangular subchannel using 1E+05 neutron histories.

To →	**fuel**	**coolant**	**1**	**2**	**3**
From ↓					
g = 1					
fuel	0.125	0.127	0.056	0.031	0.054
coolant	0.689	0.481	0.270	0.231	0.222
1	0.002	0.002	0.000	0.001	0.001
2	0.005	0.004	0.003	0.000	0.002
3	0.003	0.002	0.002	0.001	0.000
g = 4					
fuel	0.0063	0.0064	0.0029	0.0019	0.0027
coolant	0.0243	0.0178	0.0098	0.0085	0.0080
1	0.0002	0.0002	0.0000	0.0001	0.0001
2	0.0005	0.0004	0.0003	0.0000	0.0002
3	0.0003	0.0002	0.0002	0.0001	0.0000
g = 8					
fuel	0.37	0.39	0.17	0.12	0.16
coolant	1.64	1.35	0.67	0.59	0.54
1	0.02	0.02	0.00	0.01	0.01
2	0.04	0.03	0.02	0.00	0.02
3	0.02	0.02	0.01	0.01	0.00

After the preparation of the collision probabilities, the overall problem is solved using the deterministic approach and k_{eff} and the neutron flux are evaluated. The selected tolerance for convergence of the multiplication factor and the neutron flux was 1E-10. Similarly to the case of the fuel-pin cell, two sets of results were generated, considering the full scattering ($\mathrm{ICM_{SP}}$) or the approximated scattering ($\mathrm{ICM_{noSP}}$). The effective multiplication factor, its differences from the reference value, and the performance of the power iteration method in terms of the number of iterations and wall-clock time are shown in Table 4.23 for the sets of collision probabilities obtained from the different numbers of neutron histories.

Table 4.23: Multiplication factor calculation for the scaled hexagonal assembly.

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
		$\mathrm{ICM_{noSP}}$		
2,000	1.32816	441	635	1.890
10,000	1.32922	335		1.783
100,000	1.32919	338		1.833
		$\mathrm{ICM_{SP}}$		
2,000	1.33153	103	636	1.807
10,000	1.33258	-1		1.829
100,000	1.33255	2		1.827

The usage of the scattering production cross sections ($\mathrm{ICM_{SP}}$) significantly improves the prediction of the effective multiplication factor. The computational effort spent in the deterministic method is negligible compared to the estimation of the collision probabilities with Monte Carlo. The prediction of the effective multiplication factor has a minor variation (3 pcm) when increasing the neutron population from 1E+04 to 1E+05 although the computational effort is almost ten-folded. This behavior is related to the statistical uncertainties associated with the MC estimation of the collision probabilities and needs to be studied in the future. The total simulation time of the calculations based on the collision probabilities from 2E+03 and 1E+04 neutron histories is significantly lower than the one of the reference solution, i.e., saving 1,615 and 929 CPU-min, respectively.

The scalar flux is calculated in the fine regions within the nodes of the coarse meshes. The scalar flux value in each fuel pin is homogenized using Eq. (4.1). The calculated neutron scalar flux is plotted along the diagonal shown in Figure 4.13 and it is compared against the reference solution in Figure 4.14. Only the $\mathrm{ICM_{SP}}$ results with the collision probabilities estimated from 100,000 neutron histories are shown and discussed. Error bars for the Monte Carlo reference solution are small, e.g., the maximum standard deviation is equal to 0.062% and is found for the 8th energy group in the coolant region.

In this fast system with reflective boundary conditions, the spatial distribution of the neutron flux is expected to be relatively flat. The neutron scalar flux calculated with the hybrid framework shows in general very good agreement with the reference solution. The highest relative difference is found in the thermal energy group (g=8) with a value of 0.9%. For the energy groups 1, 7, and 8 the highest differences are found close to the boundary, while for the rest it is found close to the center of the assembly.

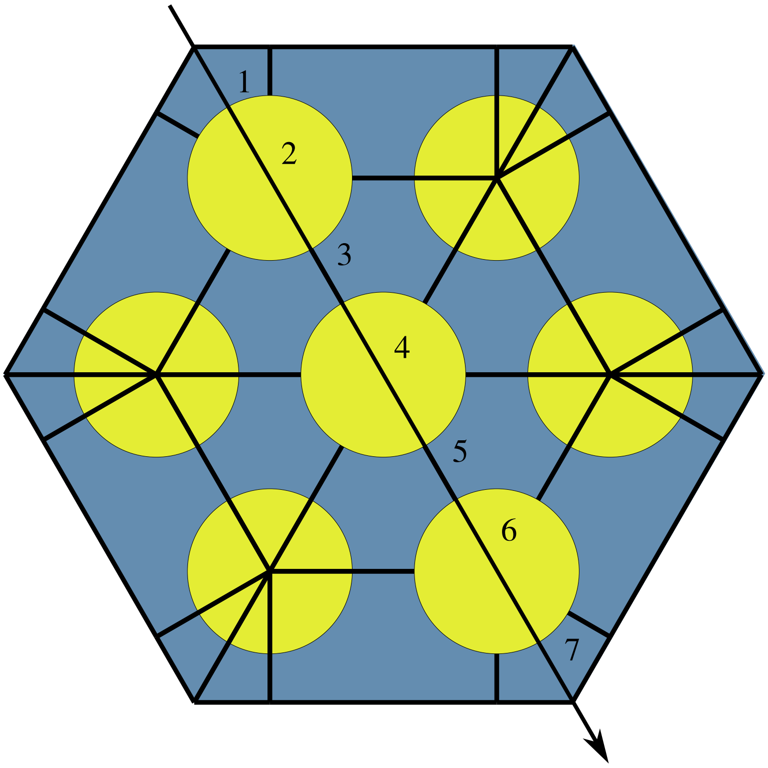


Figure 4.13: Regions 1-7 used to plot the neutron flux in Figure 4.14.

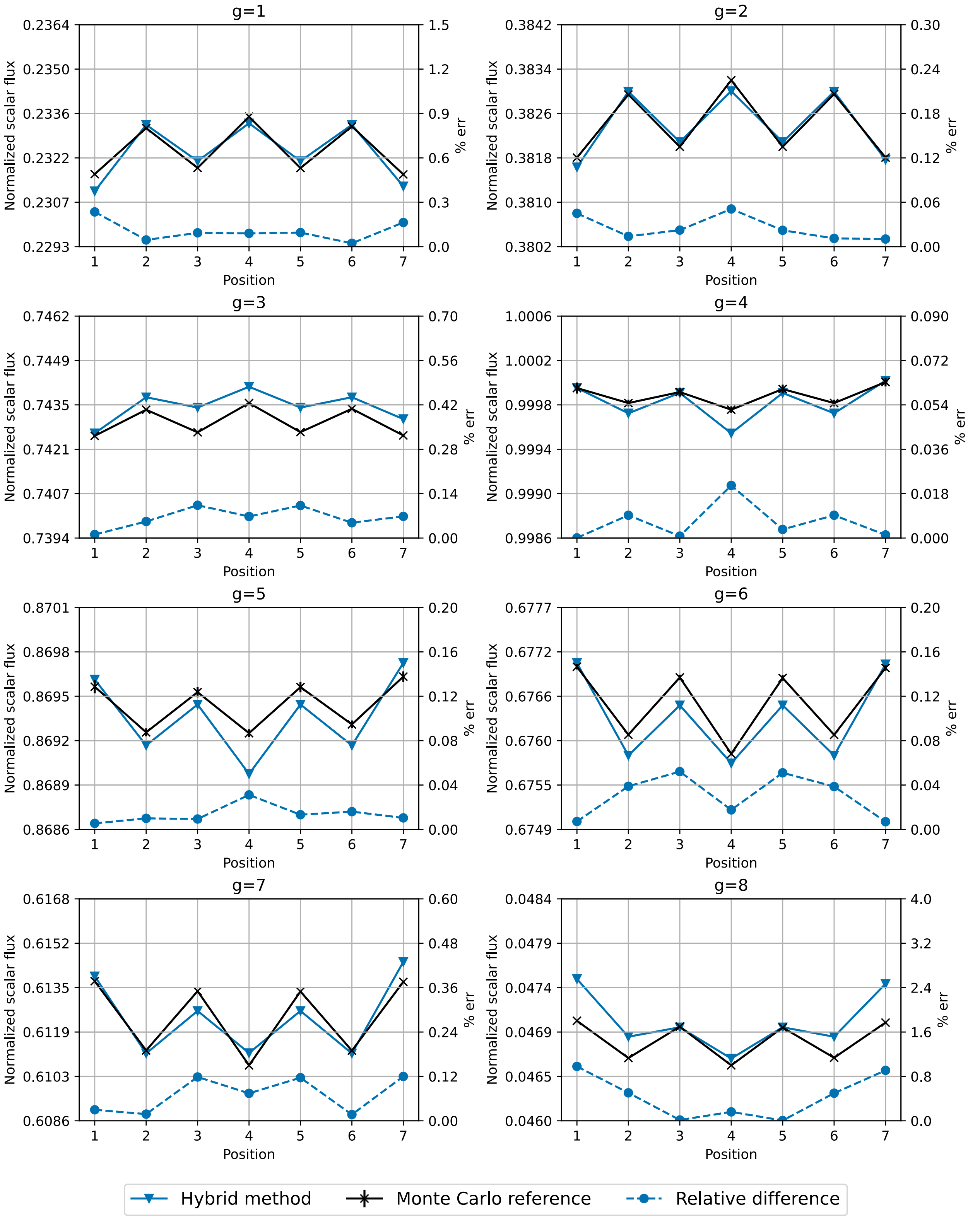


Figure 4.14: Scalar flux for the scaled hexagonal assembly problem.

4.4 Hexagonal fuel assembly

A full-scale simplified fuel assembly is taken, where the 271 fuel pins are modeled without cladding, gap, and hollow. This system is discretized according to a coarse mesh with triangular and rectangular nodes with a total of 552 nodes, 2,130 regions, and 1,710 surfaces, see Figure 4.15. Given the choice of the mesh and the symmetry of the assembly, 3 types of coarse nodes are identified. These are the same nodes found in the previous case, see Figure 4.12.

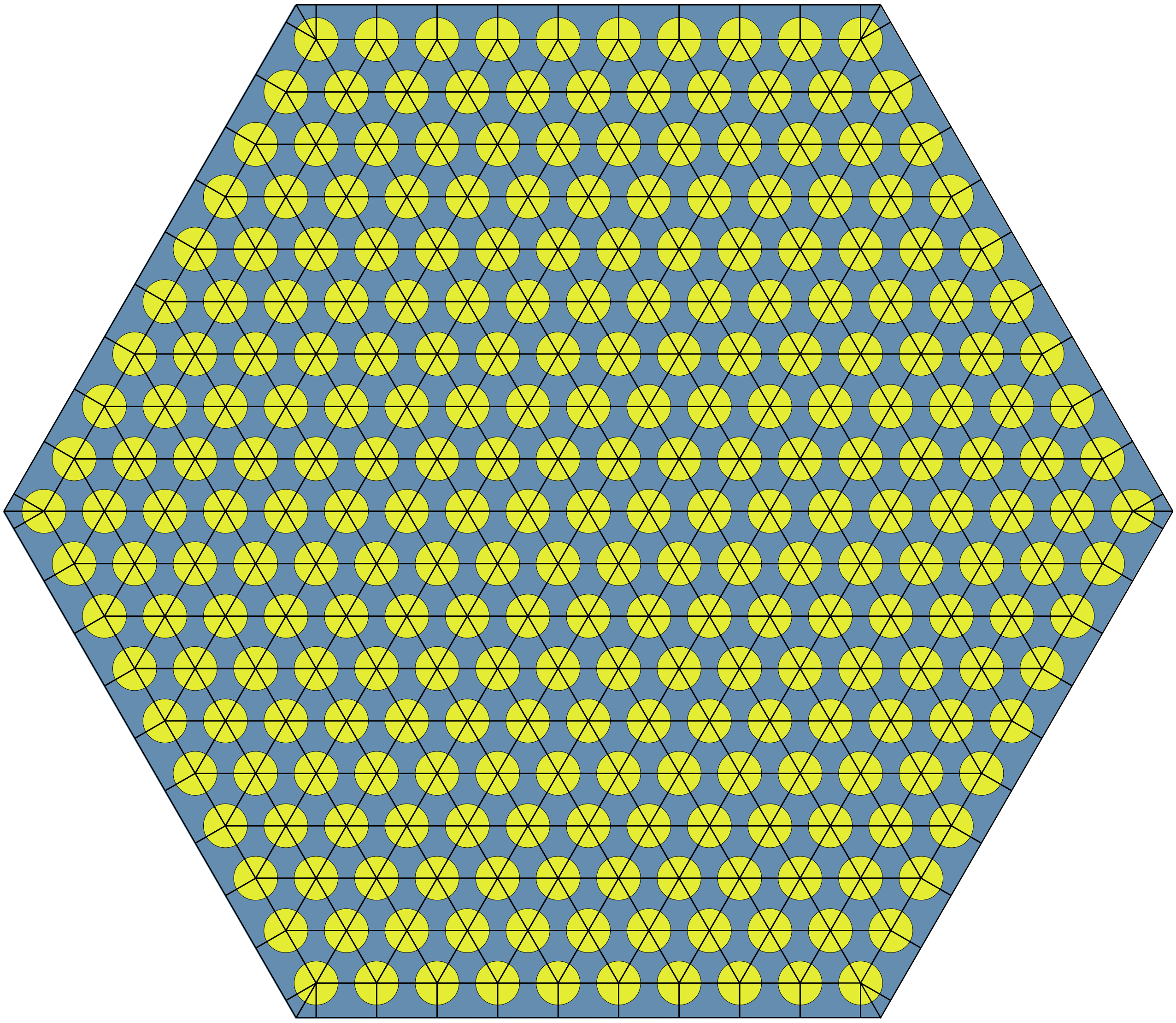


Figure 4.15: Triangular coarse mesh for the 2-D hexagonal assembly.

Since the neutron collision probabilities for each type of node were evaluated for the scaled hexagonal fuel assembly, there is no need to recalculate such values. Therefore, the collision probabilities discussed in section 4.3 can be recycled to solve the current bigger system. The computation time of the collision probabilities is reported in Table 4.19 and the standard deviation values for the three different coarse nodes in the energy groups 1, 4, and 8 are reported in Tables 4.20, 4.21, and 4.22 respectively.

After the collision probabilities were taken from the scaled hexagonal fuel assembly, the k_{eff} and the neutron scalar flux are calculated with the interface current method (Eqs. (2.39) and (2.40)). The convergence tolerance for the interface current method was selected to be 1E-10. Similarly to the previous cases, two sets of results were generated, considering scattering reactions of the type (n,n'), namely ICM_{noSP}, and considering the scattering neutron production reactions, i.e., ICM_{SP}. The effective multiplication factor is shown for the two sets of results in Table 4.24. The number of iterations to reach convergence and the wall-clock time spent in the deterministic calculation are also reported. The reference value of the effective multiplication factor is 1.35592 ± 4.8E-06 and it was obtained with Serpent2 using 2E+06 neutron histories, 3,000 active cycles, and 1,500 inactive cycles. The reference solution required 4,690 CPU min on a supercomputer cluster with a processor Intel® Xeon® Gold 6130 CPU @ 2.10GHz.

Table 4.24: Multiplication factor calculation for the 2-D hexagonal assembly.

Neutron histories	k_{eff}	$(k_{eff})_{ref} - k_{eff}$ [pcm]	No. iterations	Wall-clock time (s.)
ICM_{noSP}				
2,000	1.34714	879	611	103.106
10,000	1.34754	839		102.231
100,000	1.34779	814		104.279
ICM_{SP}				
2,000	1.35070	523	611	104.572
10,000	1.35113	480		104.388
100,000	1.35135	458		105.102

Consistently with the results in the previous tests, the use of the scattering production cross sections leads to a significant improvement of the evaluation of k_{eff}. As reported in Table 4.24, the difference from the reference goes from 814 pcm to 458 pcm (given the collision probabilities obtained from 100,000 neutron histories). Despite the use of the same collision probabilities, the agreement with the reference is lower than the one obtained for the scaled hexagonal fuel assembly. This may be due to the fact that the Monte Carlo statistical uncertainty and the approximations on the coarse nodes have more impact in the bigger system. The full fuel assembly has 552 coarse nodes and the errors may accumulate to a larger extent.

The computational effort required in the deterministic method is around 104 seconds, so it is still small in comparison with the simulation time of the collision probabilities with Monte Carlo. Although the multiplication factor was not reproduced accurately, the total simulation time of the hybrid framework, was significantly lower than that of the reference solution. The total computation time of the collision probabilities with 2E+03 and 1E+04 neutron histories is significantly lower than the one of the reference solution, i.e., saving 4,438 and 3,752 CPU-min, respectively.

After a spatial homogenization in the fine regions of the coarse nodes, the scalar flux is compared with the reference solution along the diagonal of the fuel assembly. The diagonal is shown in Figure 4.16 and the comparison of the neutron flux in Figure 4.17.

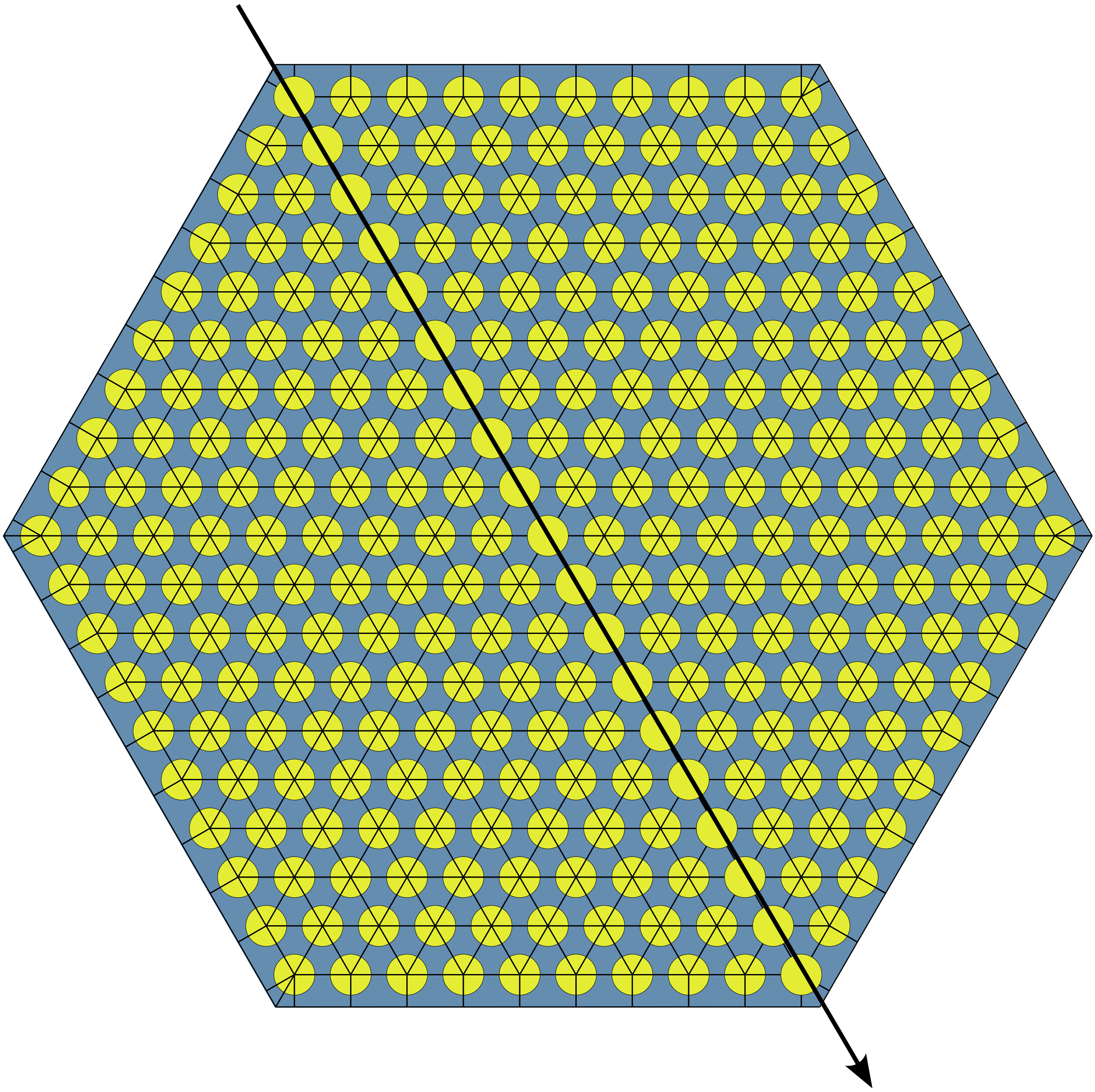


Figure 4.16: Plot line for the full-size hexagonal asembly.

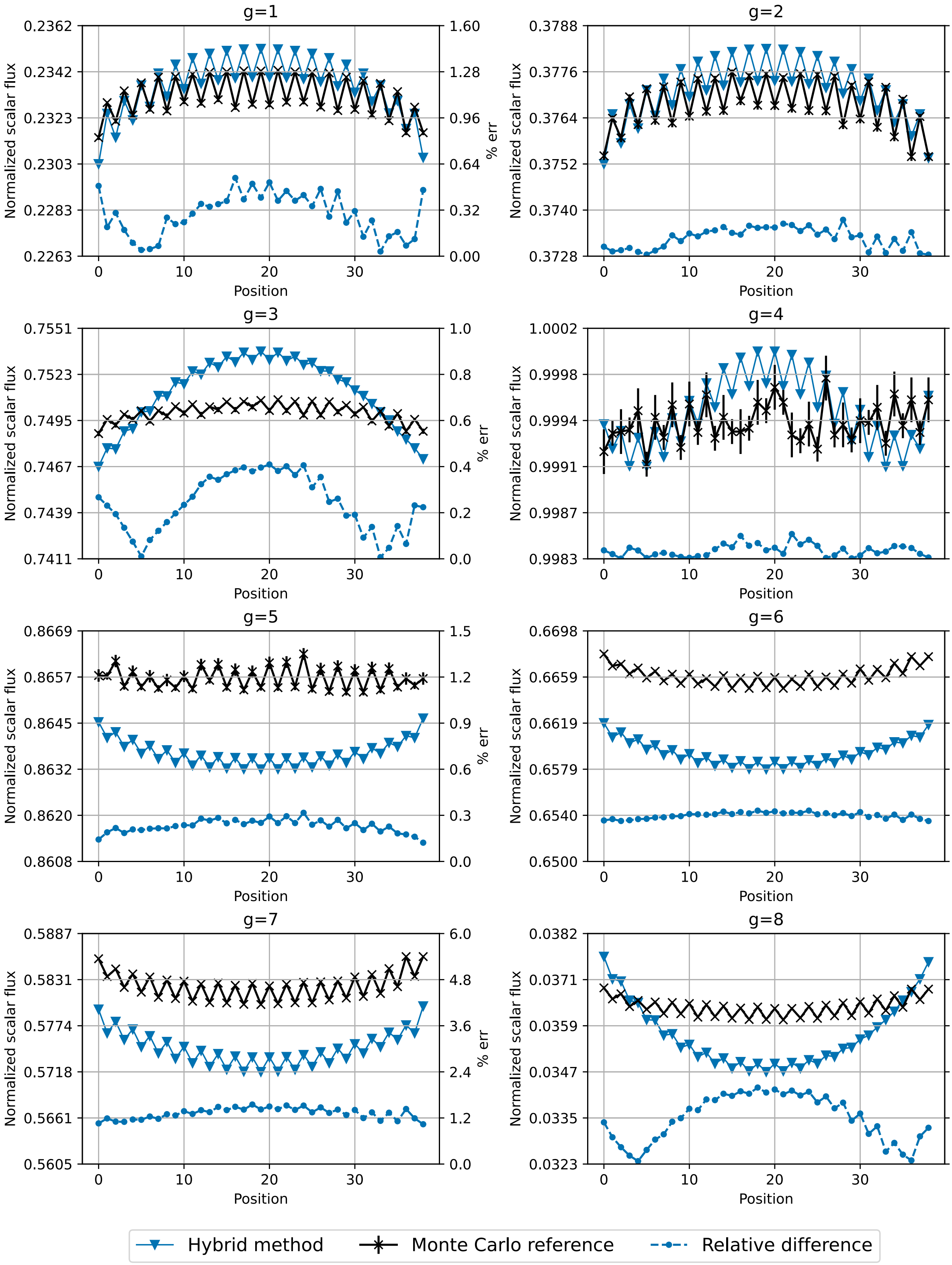


Figure 4.17: Scalar neutron flux along the diagonal of the assembly.

For this system, the spatial distribution of the flux is relatively flat. The hybrid method reproduces the scalar flux value more accurately for the energy groups 1-5 where the relative differences with respect to the Monte Carlo reference are below 0.54%. The scalar flux is reproduced with less accuracy in the energy groups 6-8, where the maximum difference is of 1.10%, 1.55%, and 3.98% respectively. The highest standard deviation in the reference solution corresponds to the neutron flux in the coolant region next to the boundary with a value of 0.1% in the 8th energy group.

In all the energy groups the relative differences are maximum close to the center of the fuel assembly. The shape of the flux corresponding to the hybrid framework appears to be less flat compared with the reference solution for energy groups 3, 5, and 8. Additionally, the shape of the neutron flux is reproduced accurately for the energy groups 6 and 7 but it is slightly shifted by a factor of 1.011 and 1.015 respectively.

Chapter 5

Conclusions

Conclusions and future work are provided in sections 5.1 and 5.2 respectively.

5.1 Summary

In this thesis, a flexible and user-friendly hybrid computational framework for 2-D static neutron transport calculations of fast reactors was developed. The framework is based on the deterministic Interface Current Method (ICM) where the collision probabilities are determined via Monte Carlo (MC) simulations. Accordingly, the computational domain is divided into smaller sub-domains where every subdomain is a local problem with fine regions and the entire set of subdomains corresponds to the global problem. The collision probabilities are estimated prior to the deterministic calculation for those local problems with unique geometry. Then, the calculated collision probabilities are replicated to the rest of the sub-domains that are geometrically equivalent. Finally, a linear system of equations is built from the combination of all the subdomains and is solved iteratively to calculate the effective multiplication factor and the scalar flux in every fine region.

The framework is designed according to a modular structure so that its use and future development are simplified. Each module has a specific functionality, such as geometry processing, calculation of collision probabilities, matrix calculations for the interface current method, and output data processing. A Python input file is used to specify the geometry, materials, mesh, and neutronics parameters for the problem of interest. For the calculation of the collision probabilities related to the local problems, the framework calls the MC code Serpent. Then, the framework assembles the matrices needed for the solution of the global problem via the ICM. The output files contain the multiplication factor, data about the convergence of the solution, the scalar fluxes in the fine regions, and the collision probabilities which can be retrieved with the ready-to-use post-processing tools of the software.

The verification process of the framework is based on 2-dimensional problems derived from data of a sodium-cooled fast reactor, i.e., 1) hexagonal fuel pin cell with simplified geometry; 2) detailed fuel pin cell; 3) scaled hexagonal lattice with simplified fuel pins; 4) full-scale hexagonal assembly with simplified fuel pins. Reference solutions are generated using Serpent.

In the first two cases, three different coarse meshes were tested: a single-node coarse mesh, a square-grid coarse mesh, and a triangular coarse mesh. For both the simplified fuel pin and the detailed fuel pin, the usage of the scattering production cross sections significantly improves the criticality calculations. The results obtained from the different meshes are similar in terms of k_{eff} and neutron fluxes, and they are close to the reference values. In the case of the detailed fuel pin, neutron interactions in the hollow and gap of the fuel pin are very low. Hence, an accurate MC estimation of the collision probabilities in these regions requires a high number of neutron histories and thus a very demanding computational effort. Yet, both k$_{eff}$ and the neutron fluxes are predicted reasonably well despite the fact that the collision probabilities used for the hollow and the gap have large statistical uncertainties.

In the third problem, a scaled hexagonal system was discretized using a coarse mesh with rectangular and triangular nodes. Again, the scattering production cross sections are important for an accurate solution. Both the effective multiplication factor and the scalar flux are close to the reference values.

In the fourth problem, a full-scale fuel assembly was discretized with a mix of rectangular and triangular coarse nodes. Considering the scattering production cross sections in the scattering matrix is important to achieve a solution closer to the reference. The discrepancy between the effective multiplication factor predicted with the framework and with the reference Monte Carlo calculation is high (around 500 pcm). The comparison of the scalar neutron fluxes shows that the buckling was reproduced accordingly, except for the energy groups 3 and 8, which will be an object of study in the future. The differences between the hybrid solution and the reference solution are small, but they grow in the less energetic groups where the collision probabilities are less accurate.

For all the verification cases, the sensitivity of the Monte Carlo estimation of the collision probabilities to the number of neutron histories was analyzed. Increasing the number of neutron histories leads to the estimation of collision probabilities with lower statistical uncertainty and thus leads to more reliable values of the effective multiplication factor and neutron fluxes. However, in some cases, a lower number of neutron histories reproduces a criticality value closer to the reference. This behavior can be attributed to the higher variability of the MC simulations using more limited statistics, and it needs to be studied in the future. The modeling of a detailed geometry with materials with low neutron reaction rates might be challenging since it requires higher MC statistics to estimate accurate collision probabilities in all the regions of the system.

The estimation of the collision probabilities through Monte Carlo is computationally more expensive than the deterministic calculation, which, for the 4 problems considered in Chapter 4, was negligible. The computational effort of the reference solution for the simplified fuel pin cell and the detailed fuel pin cell was less expensive than the overall calculation of the collision probabilities. For the bigger systems such as the scaled and the full-size hexagonal fuel assembly, the reference solution was more expensive than the hybrid one based on the sets of collision probabilities estimated from 2e+03 and 1e+04 neutron histories. In the case of the scaled system, the cheaper hybrid simulations also provided accurate results. Therefore, the hybrid framework has the potential to reduce computational costs when applied to a large system, but a careful selection of the coarse mesh is needed to minimize the number of collision probabilities to be estimated with Monte Carlo.

5.2 Future work

5.2.1 Nuclear Engineering Research Problems

The next phase of the research will be focused on the study of the applicability of the hybrid framework to more realistic system configurations.

First, the framework will be used for simulations of a fuel assembly where the fuel pins are modeled in a detailed manner, i.e., considering helium hollow, gap, and cladding regions. The introduction of such regions with low neutron reaction rates in assembly-level calculations will require generating a set of collision probabilities with lower standard deviations, at a feasible computational cost.

Second, to explore the modeling for efficient hybrid simulations of a multi-region fast breeder system, a simplified and miniaturized representation of a full core will be considered, including an inner and outer fuel region, a breeder blanket, and a shielding, see Figure 5.1. Then, one of the fuel assemblies in the outer fuel region (light blue region in Figure 5.1) will be replaced with a control rod. The control rod contains enriched boron, so it is a strong localized neutron absorber and can pose challenges because of its significant impact on the spatial distribution of the neutrons. In this way, the performance of the framework in solving non-symmetrical systems will also be evaluated.

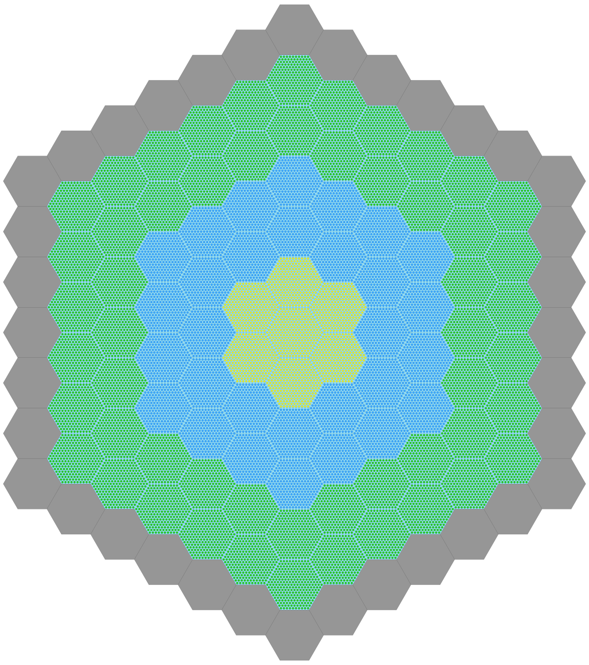


Figure 5.1: Example of a mini core characteristic of sodium-cooled fast reactor

The last step will be to optimize the framework and evaluate the possible benefits of the approach for simulations of a real sodium-cooled fast reactor core. To avoid excessive use of computational resources, the neutron transport problem will be solved only for a fraction of the core, taking advantage of the 1/3 symmetry of this system, see Figure 5.2. Relevant points to be addressed will be the scalability of the framework and the selection of a suitable coarse mesh. Since the most consuming part of the hybrid framework is the calculation of the collision probabilities, the performance against Monte Carlo will be evaluated to answer if the hybrid framework can be competitive against pure Monte Carlo calculations for full-core. A coarse mesh in the pin-cell level will require excessive computational resources for full-core calculations. Furthermore, the selection of a suitable coarse mesh needs to be studied. Additionally, the linear problem solved with the interface current method will require high-performance computational techniques, such as the implementation of sparse matrix calculation algorithms and CPU parallelization will be needed.

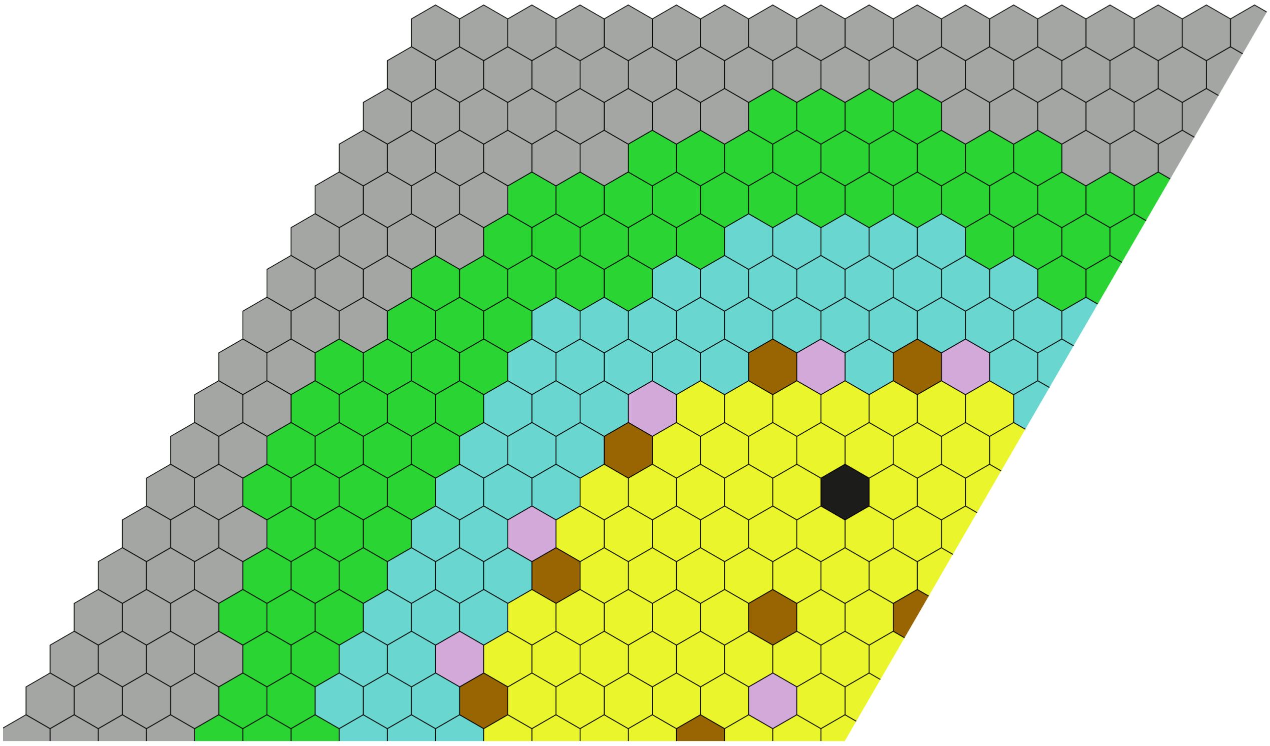


Figure 5.2: Example of a mini core characteristic of sodium-cooled fast reactor

5.2.2 Improvements to the framework

Improvements to the modeling methodology and the computational framework can be addressed in the future. For the methodology, the optimization of the coarse meshes and the application of MC fixed source calculations may be considered to reduce the computational cost and improve the accuracy of the estimation of the collision probabilities associated with the local problems.

For the computational framework, developments of future interest are summarized as follows:

- The solution of the eigenvalue problem is based on a simple power iteration method, which is known to have a low convergence rate. Consequently, implementing an acceleration method such as a standard Chebyshev technique is necessary. Non-linear acceleration using the Jacobian Free Newton Krylov algorithm can also be explored.

- The implementation of high-performance computational (HPC) techniques, such as C/C++ binders to the current Python software and parallelization at the CPU level are needed to enable criticality calculations of the full core.
- For full core calculations the matrices of the linear system become extremely large. Thus, algorithms for sparse matrix operations will be implemented as well. This will decrease the memory storage requirements and will speed up the calculations.

Bibliography

[1] *Nuclear Power Reactors in the World* (Reference Data Series 2). Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2023, ISBN: 978-92-0-137123-2. [Online]. Available: `https://www.iaea.org/publications/15485/nuclear-power-reactors-in-the-world` (cit. on p. 1).

[2] A. J. Plompen, O. Cabellos, C. De Saint Jean *et al.*, "The joint evaluated fission and fusion nuclear data library, JEFF-3.3," *The European Physical Journal A*, vol. 56, no. 7, pp. 1–108, 2020 (cit. on pp. 2, 3).

[3] N. Soppera, M. Bossant and E. Dupont, "Janis 4: An improved version of the NEA Java-based nuclear data information system," *Nuclear Data Sheets*, vol. 120, pp. 294–296, 2014 (cit. on pp. 2, 3).

[4] A. E. Waltar and A. B. Reynolds, *Fast breeder reactors.* Pergamon Press, 1981 (cit. on p. 2).

[5] P. Deng, "Computational methods for coupled fast and thermal spectrum reactor core analysis," Ph.D. dissertation, 2022 (cit. on p. 4).

[6] D. G. Cacuci, *Handbook of Nuclear Engineering*, D. G. Cacuci, Ed. Springer Science, 2010, vol. 1, pp. 538–540 (cit. on p. 6).

[7] M. J. Lee, H. G. Joo, D. Lee and K. Smith, "Coarse mesh finite difference formulation for accelerated monte carlo eigenvalue calculation," *Annals of Nuclear Energy*, vol. 65, pp. 101–113, Mar. 2014, ISSN: 0306-4549. DOI: `10.1016/J.ANUCENE.2013.10.025` (cit. on p. 6).

[8] E. R. Wolters, E. W. Larsen and W. R. Martin, "Generalized Hybrid Monte Carlo - CMFD methods for fission source convergence," 2011. [Online]. Available: `http://inis.iaea.org/Search/search.aspx?orig_q=RN:47073013` (cit. on p. 6).

[9] S. Zhuang, P. He, G. Sun, L. Hao and Y. Wu, "Hybrid Monte Carlo-deterministic method for 3D neutron transport simulation based on energy region division," *Annals of Nuclear Energy*, vol. 130, pp. 271–276, Aug. 2019, ISSN: 0306-4549. DOI: `10.1016/J.ANUCENE.2019.02.043` (cit. on p. 6).

[10] A. Haghighat, K. Royston and W. Walters, "Mrt methodologies for real-time simulation of nonproliferation and safeguards problems," *Annals of Nuclear Energy*, vol. 87, pp. 61–67, Jan. 2016, ISSN: 0306-4549. DOI: `10.1016/J.ANUCENE.2015.06.004` (cit. on p. 6).

[11] W. Walters, N. Roskoff and A. Haghighat, *A fission matrix approach to calculate pin-wise 3-d fission density distribution*, 2015. [Online]. Available: `https://pure.psu.edu/en/publications/a-fission-matrix-approach-to-calculate-pin-wise-3-d-fission-densi` (cit. on p. 6).

[12] B. Forget and F. Rahnema, "Comet solutions to the 3-D C5G7 MOX benchmark problem," *Progress in Nuclear Energy*, vol. 48, pp. 467–475, 5 Jul. 2006, ISSN: 0149-1970. DOI: `10.1016/J.PNUCENE.2006.01.005` (cit. on p. 6).

[13] H. Yi, C. Demaziere, P. Vinai and J. Leppänen, "Development and test of a hybrid probabilistic-deterministic framework based on the interface current method," in *International Conference on Mathematics and Computational Methods applied to Nuclear Science and Engineering*, American Nuclear Society (ANS), 2019, pp. 2166–2177 (cit. on pp. 6, 21, 22).

[14] C. Demazière, *Modelling of Nuclear Reactor Multi-physics, From Local Balance Equations to Macroscopic Models in Neutronics and Thermal-Hydraulics*, 1st. 2019 (cit. on pp. 9, 10, 12–14).

[15] E. E. Lewis and W. F. Miller, *Computational methods of neutron transport.* Jan. 1984. [Online]. Available: `https://www.osti.gov/biblio/5538794` (cit. on pp. 11, 12).

[16] J. Leppänen *et al.*, "Serpent–a continuous-energy Monte Carlo reactor physics burnup calculation code," *VTT Technical Research Centre of Finland*, vol. 4, p. 455, 2013 (cit. on p. 21).

[17] T. Kaltiaisenaho, "Statistical Tests and the Underestimation of Variance in Serpent 2," *Research Report*, 2014 (cit. on p. 22).

[18] M. De Berg, *Computational geometry: algorithms and applications.* Springer Science & Business Media, 2000 (cit. on p. 23).

[19] J. Gourdon, B. Mesnage, J. Voitellier and M. Suescun, "An overview of Superphenix commissioning tests," *Nuclear Science and Engineering*, vol. 106, no. 1, pp. 1–10, 1990 (cit. on p. 33).

[20] A. Ponomarev, A. Bednarova and K. Mikityuk, "New sodium fast reactor neutronics benchmark," 2018 (cit. on p. 33).

[21] H. Palomares. "Collision probabilities for the verification cases." (2024), [Online]. Available: `https://github.com/hirepanpalomares/CollisionProbabilitiesLicentiate` (visited on 13/03/2024) (cit. on pp. 36, 47, 58).