



---

# Modeling of boundary conditions in embedded lattice calculations

---

*Author:*

Daniel Martín Domínguez  
martnd@chalmers.student.se

Chalmers University of Technology  
Department of Applied Physics  
Division of Nuclear Engineering SE-4196 Göteborg

## Abstract

The coupling of the fine-mesh high-order transport operator with a coarse-mesh low-order approximation, in order to reduce the computational requirements when simulating whole reactor cores, is studied here through the boundary conditions for the fine-mesh solver. A recently developed software, solving the neutron transport equation through a discontinuous Galerkin Finite Element discrete-ordinates method, is used for the fine-mesh high-order solver, and the coarse-mesh low-order solver is simulated by a coarsening process.

As a first step, a verification and validation process is necessary to be performed. This is carried out using the softwares DRAGON and MCNP, respectively, as the references in the verification and validation processes. A very good agreement is obtained during the verification process, while for the validation, results show that different quadratures should be considered in order to improve the accuracy. In addition, a parametric study is presented, where the different parameters of the spatial and angular discretizations are analyzed, in order to understand the behavior of the solver for different configurations.

A second part of the work consists in studying the effect of coarsening the boundary conditions for a particular problem (C5G7 benchmark). This coarsening is performed to simulate the low-order approximation of the boundary conditions obtained with the coarse-mesh solver. Understanding the loss of accuracy for the fine-mesh calculations is necessary in order to improve the fine-mesh/coarse-mesh coupling for the neutronic solvers.

## Acknowledgments

Firstly, I would like to express my greatest gratitude to my supervisor Dr. Sebastián González-Pintor for his patience, and his will of being always ready to help me.

I would like to thank Dr. Manuel Calleja Reyna for all his advices, always with a touch of humour.

I would also thank my examiner Christophe Demazière for giving me the opportunity to do this Master Thesis at Chalmers University of Technology.

Last but not least, I would like to thank my parents and my sister for all the support received during these months.

# Contents

<b>List of Figures</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Methodology</b>	<b>6</b>
2.1 Description of FOREST . . . . .	6
2.1.1 Problem discretization . . . . .	6
2.1.2 Inputs to the solver . . . . .	9
2.2 Comparison tool . . . . .	10
<b>3 Test Cases</b>	<b>14</b>
3.1 The ASSEMBLY benchmark problem . . . . .	14
3.2 The C5G7 benchmark problem . . . . .	16
<b>4 Verification and validation of the Neutron transport solver</b>	<b>18</b>
4.1 Parametric study and results . . . . .	18
4.1.1 ASSEMBLY-VOID benchmark problem . . . . .	19
4.1.2 ASSEMBLY-REFLECTIVE benchmark problem . . . . .	25
4.1.3 C5G7 Benchmark problem . . . . .	29
4.1.4 Discussion of the most accurate approximations . . . . .	32
4.2 Conclusions . . . . .	36
<b>5 Study of the boundary conditions</b>	<b>38</b>
5.1 Spatial and angular approximations . . . . .	39
5.1.1 Spatial coarsening of boundary conditions . . . . .	39
5.1.2 Angular coarsening of boundary conditions . . . . .	41

5.2	Results . . . . .	42
5.3	Conclusions . . . . .	48
<b>6</b>	<b>Conclusions</b>	<b>49</b>
	<b>Bibliography</b>	<b>51</b>
<b>A</b>	<b>Input files for FOREST</b>	<b>54</b>

# List of Figures

2.1	Mesh of pin cells. . . . .	8
2.2	Flow chart of the comparison tool. . . . .	11
3.1	Geometry of the core solved with DRAGON from [1]. Indexes '0' and '1' represent the fueled and the moder- ator regions, respectively. . . . .	15
3.2	A detailed view of the fuel assemblies (left) and the core configuration (right), from [2]. . . . .	16
3.3	View of the interior of a pin cell, from [2]. . . . .	17
4.1	Variation of the AVG error for the ASSEMBLY-VOID problem as a function of $n_{ref}$ and $fe_{degree}$ , for a fixed $S_N$ order (Ordinates axis is in $\log_{10}$ scale). . . . .	20
4.2	Variation of the AVG error for the ASSEMBLY-VOID problem as a function of $S_N$ order and $n_{ref}$ , for a fix $fe_{degree}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	20
4.3	Variation of the AVG error for the ASSEMBLY-VOID problem as function of $S_N$ order and $fe_{degree}$ , for a fix $n_{ref}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	21
4.4	Comparison of the tendency in the RMS, and MRE er- ror in the ASSEMBLY-VOID problem (Ordinates axis is in $\log_{10}$ scale). . . . .	22
4.5	Variation of the k-effective error in pcm for the ASSEMBLY- VOID problem as function of the $n_{ref}$ and the $fe_{degree}$ , for a fixed $S_N$ (Ordinates axis is in $\log_{10}$ scale). . . . .	23

4.6	Variation of the k-effective error in pcm for the ASSEMBLY-VOID problem as function of the $S_N$ and the $n_{ref}$ , for a fixed $fe_{degree}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	24
4.7	Variation of the k-effective error in pcm for the ASSEMBLY-VOID problem as function of the $S_N$ and the $fe_{degree}$ , for a fixed $n_{ref}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	24
4.8	Variation of the AVG error for the ASSEMBLY-REFL problem as function of the discretization parameters (Ordinates axis is in $\log_{10}$ scale). . . . .	26
4.9	Variation of the k-effective error in pcm for the ASSEMBLY-REFL problem as function of the $n_{ref}$ and the $fe_{degree}$ , for a fixed $S_N$ (Ordinates axis is in $\log_{10}$ scale). . . . .	27
4.10	Variation of the k-effective error in pcm for the ASSEMBLY-REFL problem as function of the $S_N$ and the $n_{ref}$ , for a fixed $fe_{degree}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	27
4.11	Variation of the k-effective error in pcm for the ASSEMBLY-REFL problem as function of the $S_N$ and the $fe_{degree}$ , for a fixed $n_{ref}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	28
4.12	Variation of the AVG error for the C5G7 problem as function of the $n_{ref}$ and the $fe_{degree}$ , for a fixed $S_N$ (Ordinates axis is in $\log_{10}$ scale). . . . .	29
4.13	Variation of the AVG error for the C5G7 problem as function of the $S_N$ and the $n_{ref}$ , for a fixed $fe_{degree}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	30
4.14	Variation of the AVG error for the C5G7 problem as function of the $S_N$ and the $fe_{degree}$ , for a fixed $n_{ref}$ (Ordinates axis is in $\log_{10}$ scale). . . . .	30
4.15	Values of the k-effective for different spatial and angular discretization. . . . .	31
4.16	Distribution of the relative error per pin in the ASSEMBLY problems. . . . .	34
4.17	Distribution of the relative error per pin in the C5G7 benchmark problem for two different simulations. . . . .	35
4.18	Neutron flux distribution for the C5G7 benchmark, for energy group $g=7$ . . . . .	35

5.1	Spatial coarsening levels for the albedo . . . . .	40
5.2	Distribution for the relative error for the simulation with $n_{ref}=1$ , $S_N$ order=8 and $fe_{degree}=2$ . . . . .	43
5.3	Variation of the AVG and the maximum relative error in the inner-UO <sub>2</sub> (Ordinates axis is in log <sub>10</sub> scale). . . .	43
5.4	Variation of the AVG and the maximum relative error in the MOX (Ordinates axis is in log <sub>10</sub> scale). . . . .	44
5.5	Variation of the AVG and the maximum relative error in the outer-UO <sub>2</sub> (Ordinates axis is in log <sub>10</sub> scale). . . .	44
5.6	Distribution of relative power pin error for the three assemblies, for angular and spatial coarsening level 0. . .	46
5.7	Distribution of relative power pin error for the three assemblies, for angular and spatial coarsening level 2. . .	48



# Chapter 1

## Introduction

In the study of nuclear reactors, one of the principal targets is to understand the behavior of the reactor core for a specific operational state. In order to know this, it is very important to obtain the different parameters of the core, such as its k-effective, its neutron flux or its pin power distribution. Neutronic codes are developed in order to study the system for the different operating conditions. These codes solve a module describing the neutron transport phenomena coupled with the thermal hydraulic of the system.

In the present work we focus on the neutronic module. There are mainly two approaches for the modeling of neutron transport equation inside the reactor core. These two approaches are based on stochastic and deterministic methods. These testings allow us to obtain different parameters of a studied reactor for several operational states and they make it possible to predict how the reactor will behave if some of its initial conditions would be changed.

The stochastic approach is based on the Monte Carlo method. Examples of such code are MCNP [3] or OpenMC [4], and they simulate the neutron transport phenomena statistically. This procedure consists in recording all the events in which neutrons are involved, and for doing it every particle is followed. Knowing in which nuclear reactions (such as absorption, fission or capture) neutrons are involved, permits to

reproduce precisely the behavior of these particles. The computing time to reach the results using Monte Carlo codes is larger than the required time by deterministic codes, but they give more accurate results. Consequently, these codes are generally validated by using as reference Monte Carlo results. However, deterministic methods are capable to give accurate results in a much lower computing time and because of this reason, they are employed in core simulations.

Deterministic codes, such as DRAGON [5] or OpenMOC [6], solve an approximation to the transport (Boltzmann) equation, obtaining a solution that depends on the different methods. This is an integro-differential equation describing the distribution of neutrons in a reactor core, with the following form:

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial}{\partial t} \psi(\vec{r}, \vec{\Omega}, E, t) + \vec{\Omega} \cdot \nabla \psi(\vec{r}, \vec{\Omega}, E, t) + \Sigma_T(\vec{r}, E, t) \psi(\vec{r}, \vec{\Omega}, E, t) & (1.1) \\ = \int_{(4\pi)} \int_0^\infty \Sigma_s(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E, t) \psi(\vec{r}, \vec{\Omega}', E', t) d\vec{\Omega}' dE' & \\ + \frac{\chi(E)}{4\pi} \int_0^\infty \nu(E') \Sigma_f(\vec{r}, E', t) \int_{(4\pi)} \psi(\vec{r}, \vec{\Omega}', E', t) d\vec{\Omega}' dE', & \end{aligned}$$

where the angular neutron flux,  $\psi(\vec{r}, \vec{\Omega}, E, t)$ , depends on the following independent variables:  $E$  for energy,  $\vec{\Omega}$  for direction of travel,  $\vec{r}$  for the position and  $t$  for the time. A detailed explanation of the different terms can be found in [7]. It is because of having seven variables which are time, energy, three variable in space and two in direction that it is a very complex problem from the computational point of view.

There exist several variations of the Boltzmann neutron equation, such as the integral or the differential form of the equation, the even/odd parity form, or the steady state formulation, depending on the particular problem being simulated [8].

The equation shown above is solved doing discretizations of the variables, and more specifically there are an angular, a spatial and an energy discretization. Several computational methods have been de-

veloped in order to do these discretizations. Although validated codes give accurate solutions, they usually combine these different discretizations in order to solve the neutron Boltzmann equation.

The energy discretization is usually made by collapsing the energy range in several subdomains. The energy discretized cross-section and neutron flux are obtained from collapsing the energy-dependent cross-section with an energy-dependent flux depending on the characteristics of the reactor being modeled [9].

Regarding to the angular discretization, different options are possible [10]. Deterministic methods implemented on different codes are, for instance, the Discrete Ordinates Method, also called the  $S_N$  method, used for example by the code APOLLO II [11] and HELIOS 2 [12], the Method Of Characteristics (MOC) used by CRX code [13], the Collision Probability Method used by HELIOS 2 [12], and the Spherical Harmonics method used by the code VARIANT [14]. The method used in this work is the discrete-ordinates.

As for the angular discretization, several methods exist for doing the spatial discretization, as for example the Finite Element method or nodal methods used by code VARIANT [14], Flat source arbitrary spatial mesh used by codes APOLLO II [11] and CRX [13], Cartesian Finite Differences method used by the code DORT, and the method of Finite Volumes [15]. The Finite Element Method [16], is the one used in the present work.

Due to the complexity of the neutron transport equation, and the way that it is solved, the simulation of nuclear reactor cores is a very challenging problem from the computational point of view. For this reason, the spatial resolution is split in two different scales, which are fine-mesh and coarse-mesh. This division of the problem allows to reduce the time that is needed for having the results, keeping a high accuracy.

A fine-mesh transport solver is used for small subdomains, typically one assembly. This allows us to study the behavior of neutrons on a reduced scale, reducing errors on calculations. On the fine-mesh

solver, the core is divided in independent assemblies, which are then sub-divided in pin cells, and every pin cell is composed of moderator, fuel pin, cladding and gap defining a complex geometry. Hence, the fine-mesh is generally solved by the neutron transport equation solver to capture the different heterogeneities, although it is possible to solve it by a low-order solver.

A low-order approximation of the neutron transport equation, such as the diffusion equation, is generally used in the coarse-mesh calculations. This module is used because it is intended to couple the calculations of the fine-mesh module for each node, so that a global solution for the whole reactor is determined. Due to the computational effort necessary to solve the neutron transport equation, low-order approximations are usually employed for full core calculations.

The aim of the project described in this thesis is the verification and validation of the code FOREST, as well as the analysis of how changes in boundary conditions affect the results. Hence, these two tasks have been studied separately, as it is explained hereafter.

In order to do the verification of a code, the results given by its simulations have to be compared with the solution given by a benchmark, or other obtained simulating codes that have already been verified. In this case, for testing the accuracy of the code, results have been compared with solutions given by other codes taken as reference. Several problems having different geometries and boundary conditions have been simulated. Firstly, the comparison has been done with the solution given by the collision probability code DRAGON [5]. Then, results of the code FOREST have been compared with the ones given in the C5G7 benchmark [2], which have been obtained by using the Monte Carlo code MCNP5 [17]. Problems named before have been solved for different combinations of the discretization's parameters in order to perform a parametric study. In this way, it is possible to optimize the calculations.

In the analysis of the boundary conditions, the simulations have been done only using the problem in the C5G7 benchmark [2]. Firstly, the

whole core of the C5G7 benchmark problem have been solved for an specific combination of the discretization parameters. From this simulation, the pin powers of the problem as well as the boundary neutron fluxes have been obtained. From the results of the pin power calculations, the pin power of every assembly has been extracted and taken as reference. Then, different spatial and angular coarsenings have been done to the boundary neutron fluxes, and for every assembly the pin power have been recalculated with the new boundary conditions. In this way, comparing the values of the pin power obtained before and after the coarsening of the boundary conditions, it is possible to know how are the errors introduced by the coarsenings.

The structure of this thesis is as follows. In Chapter 2, the characteristics of the code that is being used are exposed and the different modules of the Python tool that have been developed are explained. Then, in Chapter 3, the two problems that have been studied in the work are presented. Chapter 4 studies the verification and validation of the neutron transport solver and the effect of coarsening the boundary conditions is studied in Chapter 5. Finally, Chapter 6 summarizes the conclusions of the present work.

# Chapter 2

## Methodology

In this chapter, the methodology of the work is exposed. This chapter is divided in two sections, firstly, the description of FOREST (Framework Oriented to REactor Simulations) is presented, where its characteristics, such as the equations that it solves, the mathematical methods that it uses, or the parameters that the user can change in simulations, are explained. Finally, the different modules of the comparison tool developed during this work are presented.

### 2.1 Description of FOREST

#### 2.1.1 Problem discretization

The solver that has been tested is a discrete ordinates discontinuous Galerkin fine-mesh solver for the neutron transport equation, and it allows us to model the behavior of neutrons inside a core. The code model the steady-state formulation of the Boltzmann equation:

$$\begin{aligned}
& \vec{\Omega} \cdot \nabla \psi(\vec{\mathbf{r}}, \vec{\Omega}, E) + \Sigma_T(\vec{\mathbf{r}}, E) \psi(\vec{\mathbf{r}}, \vec{\Omega}, E) \quad (2.1) \\
= & \int_{(4\pi)} \int_0^\infty \Sigma_s(\vec{\mathbf{r}}, \vec{\Omega}' \rightarrow \vec{\Omega}, E' \rightarrow E) \psi(\vec{\mathbf{r}}, \vec{\Omega}', E') d\vec{\Omega}' dE' \\
& + \frac{1}{\lambda} \frac{\chi(E)}{4\pi} \int_0^\infty \nu(E') \Sigma_f(\vec{\mathbf{r}}, E') \int_{(4\pi)} \psi(\vec{\mathbf{r}}, \vec{\Omega}', E') d\vec{\Omega}' dE',
\end{aligned}$$

where the eigenvalue  $\lambda$  determines the criticality of the reactor, measuring the ratio between the loss and production of neutrons in the reactor [7].

The chosen method to perform the angular discretization has been the Discrete Ordinates Method ( $S_N$ ). In this method, the equation in its multigroup format is discretized by a collocation method on a set of quadrature points  $\{\Omega_n\}_{n=1}^{N'}$ , with their respective weights  $\{\omega_n\}_{n=1}^{N'}$ . The order of the discrete ordinates,  $N$ , is related to the number of quadrature points,  $N'$ , as follows:

$$\begin{aligned}
N' &= N & \text{for } d &= 1, \\
N' &= N(N+2)/2 & \text{for } d &= 2, \\
N' &= N(N+2) & \text{for } d &= 3,
\end{aligned}$$

being  $d$  the spatial dimension of the problem.

One of the parameters that it can be changed in the input of the problem is the  $S_N$  order that will be used in the angular discretization. Theoretically, when a higher  $S_N$  order is used, more angular directions are considered, so the transport phenomena is better represented.

For the spatial discretization the method applied by the solver is the Finite Element Method, and more specifically the discontinuous Galerkin method. This method is the most popular among the Finite Elements Method, due to its flexibility to deal with complex domains and the straightforward extension to high order approximations [18]. This method consists in splitting the space in smaller subdomains called finite elements, where the flux will be expanded in a set of functions. Thus, knowing the boundary conditions of the problem, the

equation can be solved connecting the individual solutions of every subelements.

Moreover, in order to implement the Finite Element Method, the deal.II library [19] has been used. This library allows to reuse a large amount of code already tested and optimized for this specific method. We are able, as it was the case for the angular discretization, to change some parameters for the spatial discretization. In this case, the input allows us to modify the refinement of the mesh (see Figure 2.1) and the truncation degree for the polynomial expansions used for the angular neutron flux. In this way, it is possible to adjust the discretization to our needs.

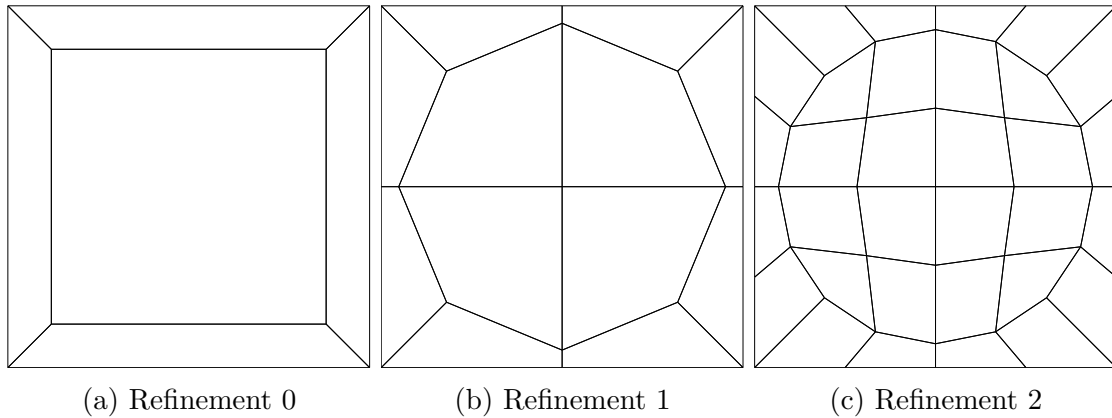


Figure 2.1: Mesh of pin cells.

Since the three parameters named above can be modified to adjust simulations, it is interesting to know which are the most interesting combinations for reaching a good accuracy without spending too much computational time. A parametric study has thus been carried out, as explained at the end of this chapter.



## 2.1.2 Inputs to the solver

The characteristics of the studied problems are introduced to the solver with three inputs. The three inputs describe the geometry, the materials and the settings of the problem that is analyzed. These inputs are in a \*.xml format file, and all of them have to be defined in order to run the simulations.

In the input of geometry (see File [A.1](#) in Appendix [A](#)), the composition of the core as well as its size and boundary conditions are defined. The shape of the input for different problems should be very similar, but variations exist depending on the problem that is being defined. For the problems studied in this project, which are described in the next section of the chapter, there exist some differences due to the fact that two of them are a one-assembly core, and the other is a core composed by several assemblies. This input is formed by nested structures as explained below.

Firstly, the parameters of the whole core, such as its dimensions, its numbers of nodes, the distribution of the assemblies inside the core and the boundary conditions on its sides, have to be introduced. After the core characteristics, the input continues with the description of the assemblies. Here, the lattice is defined by introducing its geometry, that is, defining how the different pin cells are arranged inside them, with every pin cells assigned by a number depending on its material. Every pin cell has inside a circular pin, and its definition is the last part of the input. These pins are defined introducing their type (pin or box), their radius and the material by which they are surrounded.

The properties of the materials that compose the assemblies are defined in the second input(see File [A.2](#) in Appendix [A](#)). The first parameter that has to be defined is the number of energy groups in the problem. Then for the materials, which are assigned by a number, all their properties used for solving the problem have to be introduced. In order to know the behavior of the neutrons on the core, it is essential to know how the different materials affect the neutron population. The properties that have to be defined for every material are thus

the total cross-section, the scattering block, the Nu, the Chi and the Nu\*Fission cross-sections. As the problems that we have to simulate must be identical to their reference, all the neutron data used have been taken from their respective references.

The last input that has to be defined concerns the settings for the simulations (see File [A.3](#) in Appendix [A](#)). Here, the angular and spatial discretizations are defined: the number of  $S_N$  order, mesh refinements, and degree of the polynomial expansion. The algebraic parameters to calculate the eigenvalue are also defined. Those are the maximum number of iterations allowed and the tolerance used in the group iterations or in the eigenvalue calculations.

## 2.2 Comparison tool

How accurate the tested solver is, should be determined by comparing its results with codes like DRAGON and MCNP, which have already been verified. Since simulations of the same problems have been run several times and that the output data of every simulation are, among others, large matrices, in this case 19 x 19 and 51 x 51, it is not possible analyze these results without a computational help. In order to calculate the different errors, as well as to obtain graphics for the parametric study (which is described in the next section of the chapter), a comparison tool has been developed.

This comparison tool has been created in the programming language Python and it is composed by four modules connected between them, as shown in [Figure 2.2](#).

When the problem is simulated with different settings, inputs of the solver describing geometry and materials remain constant whereas the input defining the solver settings is the only one that changes. The first module generates automatically the settings input file of every problem and also launches the simulations. Thus, it allows us to run the code and to obtain the results of the studied cores automating the process. In this way, the time has been reduced only to the time that

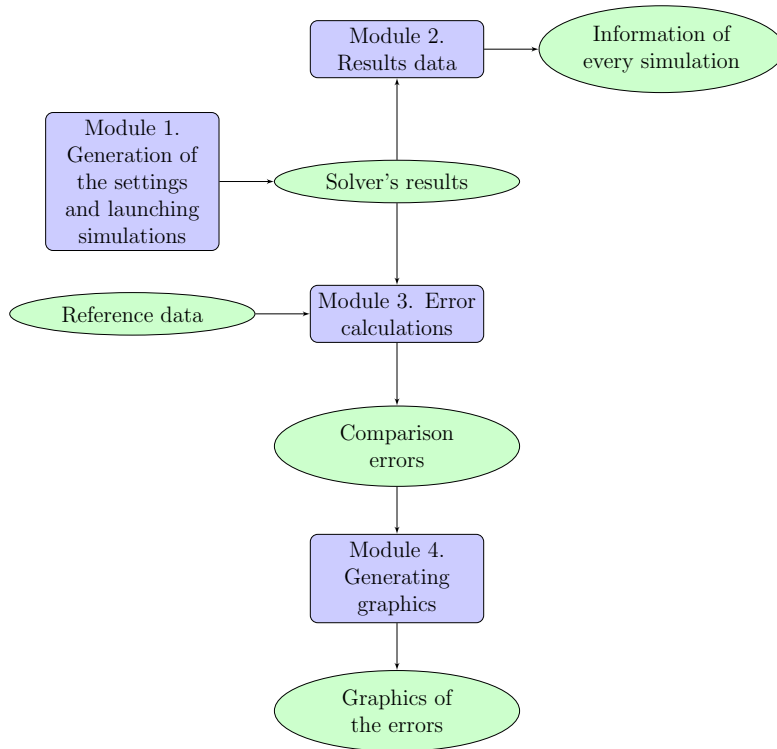


Figure 2.2: Flow chart of the comparison tool.

the solver spends on calculations.

The second module gives the results of the solver, that are the k-effective and the pin power distribution, specifying the parameters of the discretization used for reaching these results. Hence, it is only an informative module in order to quickly check the results for a specific simulation.

The third module calculates differences generated by two input files. In this case, the reference file is assigned, and then it uses the results given by the first module as its second input in order to do the comparisons. It is thus possible to automatically obtain errors derivation of every simulation launched by the first module. The calculated errors are the average (AVG), the root mean square (RMS) and the mean relative error (MRE), with the reference solution coming from the C5G7 Benchmark [2]. The two last errors, the RMS and the MRE, have to be compared with the AVG error, in order to interpret the information that they provide. The three errors are described as:

- The average per cent error (AVG) given by,

$$AVG = \frac{\sum |e_n|}{N}, \quad (2.2)$$

where  $e_n$  is the calculated relative error in the pin  $n$  and  $N$  is the number of fuel pins.

This error gives the average pin power error between the solver's results and the reference. It is a useful parameter because it allows us to quickly know if the pin power distribution given by the tested solver is close to the reference, but it does not give information about the power distribution.

- The root mean square of the pin power per cent error distribution (RMS),

$$RMS = \sqrt{\frac{\sum e_n^2}{N}}, \quad (2.3)$$

This error allows us to know the error distribution with respect to the AVG error, i.e, an RMS error larger than the AVG error, indicates larger errors between pins. Even if two solver's results have the same AVG error, they will not have the same pin power values and with RMS error it is possible to know the range of values for the error of the pin power distribution in a problem.

- The mean relative pin power per cent error (MRE),

$$MRE = \frac{\sum |e_n| \cdot p_n}{N \cdot p_{avg}}, \quad (2.4)$$

where  $p_n$  is the power in the pin  $n$ , and  $p_{avg}$  is the average power by pin.

The MRE error used in the benchmark allows us to know if the pin errors are more concentrated in pins which have the highest or the lowest power, or otherwise if it is concentrated in average power pins. When the value of this error is larger than the AVG error, the error is concentrated in the high-power pins whereas if

it is smaller, the error is concentrated in the low-power pins. If there are no important differences between MRE and AVG errors, it means that there are no large errors in pins with high or low power.

Therefore, with these three errors we can compare our solution with the one of the reference, and also know the distribution of errors in our results.

Finally, the fourth module generates graphics with the data obtained from the comparison with the benchmark data. The parametric study is done by the analysis of these graphics, and it allows us to see the behavior of the solver for different values of the mesh refinement,  $S_N$  order and truncation degree for the polynomial expansions. The values that have been analyzed are the AVG, the RMS, and the MRE errors, the k-effective value and the absolute error of the k-effective. Besides these graphics, the module gives a map of the pin power relative errors, which shows the distribution of the pin wise relative errors in the reactor core.

# Chapter 3

## Test Cases

In this chapter, the three problems simulated in this work are described in a detailed way. Firstly, the characteristics of the two ASSEMBLY problems, and then the C5G7 benchmark problem are exposed.

### 3.1 The ASSEMBLY benchmark problem

The code DRAGON has been chosen to generate the reference in the process of verification because, like FOREST, it is a discrete-ordinates code. Thus, we will be able to know if FOREST solves the mathematical model of the problem correctly. The problem described hereafter is a two-dimensional and three-energy group BWR bundle extracted from [1], and its geometry is shown in Figure 3.1.

The cross-sections used for the fuel (material 1) are shown in Table 3.1, while the cross-sections for the moderator (material 0) are shown in Table 3.2.

This problem is an assembly composed by 19 x 19 length cells made of two materials that are a smeared fuel-clad mixture, and a moderator (water), representing an assembly composed by 9x9 pins. Their dimensions depend on the material, in this way, the fuel mixture pin

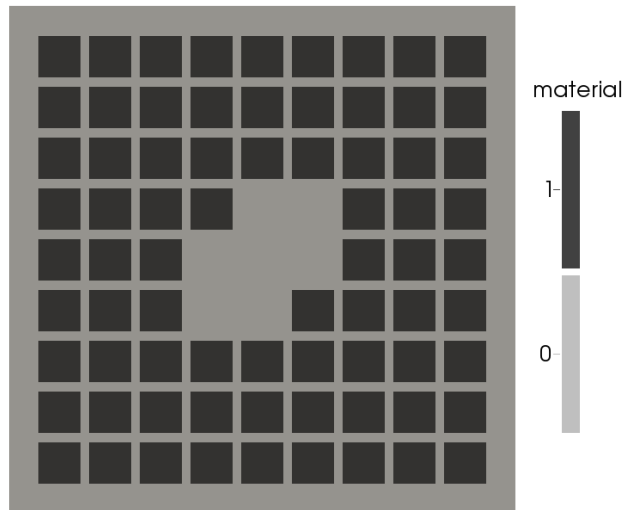


Figure 3.1: Geometry of the core solved with DRAGON from [1]. Indexes '0' and '1' represent the fueled and the moderator regions, respectively.

Table 3.1: Cross-sections for material 0 (fuel) in Figure 3.1

	$g = 1$	$g = 2$	$g = 3$
$\Sigma_{t,g}$	0.2822058997	0.4997685502	0.4323754911
$\Sigma_{f,g}$	0.0028231045	0.0096261203	0.1123513981
$\nu_g$	2.7202775245	2.4338148428	2.4338000000
$\chi_g$	0.9996892490	0.0003391680	0.0000000000
$\Sigma_{s,1\leftarrow g}$	0.2760152893	0.0000000000	0.0000000000
$\Sigma_{s,2\leftarrow g}$	0.0011230014	0.4533430274	0.0000378305
$\Sigma_{s,3\leftarrow g}$	0.0000000000	0.0014582502	0.2823864370

Table 3.2: Cross-sections for material 1 (moderator) in Figure 3.1

	$g = 1$	$g = 2$	$g = 3$
$\Sigma_{t,g}$	0.3333694980	0.5887110656	1.6512246291
$\Sigma_{s,1\leftarrow g}$	0.2432946408	0.0000000000	0.0000000000
$\Sigma_{s,2\leftarrow g}$	0.0898364840	0.4533430274	0.4413753398
$\Sigma_{s,3\leftarrow g}$	0.0000387911	0.1465683257	1.6300848232

cells have a length of 1.15 cm, while the moderator cell length is equal to 0.25 cm, apart from the ones placed in the border of the core, which have a size of 0.825 cm. The overall dimension of the core is 14 x 14 cm. This problem has been solved twice, firstly applying void boundary conditions (ASSEMBLY-VOID), and secondly applying reflective boundary conditions (ASSEMBLY-REFL), in order to verify

the correct implementation of the reflective boundary conditions in the solver.

### 3.2 The C5G7 benchmark problem

The code chosen for the validation of FOREST has been MCNP. This code uses probabilistic methods in order to solve the problem. It can give very accurate results, using the exact geometry, and reproducing the transport behavior of neutrons. Comparing with this code we will be able to prove if FOREST represents correctly the physical phenomena.

The problem chosen as reference is the one described in the C5G7 Benchmark of 2003 [2], and here the code used as reference is the Monte Carlo code MCNP. Characteristics of this problem are different from the ones of the ASSEMBLY problem described previously. It is a two-dimensional and seven-energy-group problem with different boundary conditions on its sides. This problem represents the quarter of a mini-core, and it is shown in Figure 3.2.

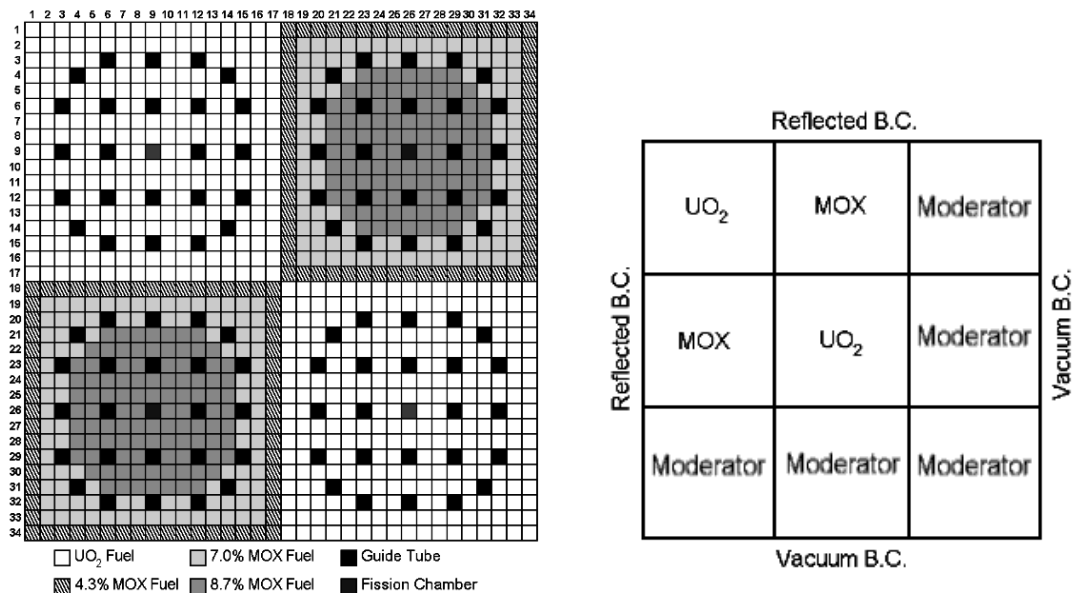


Figure 3.2: A detailed view of the fuel assemblies (left) and the core configuration (right), from [2].



This core is composed by 9 assemblies: 2 of MOX, 2 of UO<sub>2</sub> and 5 of moderator, arranged in a particular way. Assemblies are formed by 17x17 square lattice of pin cells, so the active core is formed by 34x34 pin cells, every pin cell having a fuel pin inside. Pins have a circular shape, as can be observed in Figure 3.3.

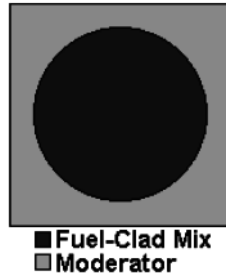


Figure 3.3: View of the interior of a pin cell, from [2].

Every pin has a radius of 0.54 cm, and the pitch is equal to 1.26 cm. Thus, the size of each assembly is 21.42 x 21.42 cm, and the overall dimension for the quarter of the reactor is 64.26 x 64.26 cm.

In addition, as shown in Figure 3.2, assemblies are composed by different materials which are moderator, UO<sub>2</sub> fuel, 4.3%, 7.0% and 8.7% enrichment of MOX, the guide tube and the fission chamber. All the characteristics of these materials, such as their different cross-sections are provided in the benchmark [2], in its Appendix A. Regarding boundary conditions, reflective conditions are applied in -x, +y sides, and vacuum are applied in +x, -y sides, as also shown in Figure 3.2.

# Chapter 4

## Verification and validation of the Neutron transport solver

In this chapter the results of the parametric study are presented. The conclusions of the processes of verification and validation from the analysis of the results of the parametric study are also explained.

### 4.1 Parametric study and results

The three problems simulated in this work have been run several times, for different angular and spatial discretizations. Discretizations are defined by values of the mesh refinement ( $n_{ref}$ ), the truncation degree for the polynomial expansions ( $fe_{degree}$ ), and the order for the discrete ordinates ( $S_N$ ), so these values are different for every simulation. The main aim of the parametric study is to know if the solver behaves correctly, as well as to see where it could have some limitations. In addition, the study has been done according to the complexity of the problem being solved. Thus, the one-assembly problem with void boundary conditions has been studied firstly, then the one with reflective boundary conditions, and finally the C5G7 problem.

The analysis has been performed from the graphics obtained with the fourth Python modules and they show the variables analyzed as a function of the discretization parameters. Graphics that have been

built represent the variables as a function of two out of the three discretization parameters, and the third one fixed at its maximum value. Hence, graphics for a variable are the following:

- $fe_{degree}$  vs  $n_{ref}$ , with the  $S_N$  order=20.
- $n_{ref}$  vs  $fe_{degree}$ , with the  $S_N$  order=20.
- $S_N$  order vs  $n_{ref}$ , with the  $fe_{degree}=2$ .
- $n_{ref}$  vs  $S_N$  order, with the  $fe_{degree}=2$ .
- $S_N$  order vs  $fe_{degree}$ , with  $n_{ref}=2$ .
- $fe_{degree}$  vs  $S_N$  order, with  $n_{ref}=2$ .

Where 20, 2 and 2 are the maximum values chosen here for the  $S_N$  order, the mesh refinement, and the finite element degree used in the simulations, respectively.

Analyzing these graphics, it is possible to know how interesting is for this solver to change the parameters of discretization, by studying the variation of the accuracy of the results. On one hand, this is important because higher discretization values require larger computational time spend on calculations. On another hand, it allows to see whether the FOREST code has some limitations in its way of solving a problem, and where it could be improved.

#### 4.1.1 ASSEMBLY-VOID benchmark problem

Here, the graphics previously described and representing the values obtained for two different variables are exposed. Through them, the behavior of the solver for the different values in its discretization parameters is studied.

In figure 4.1, the AVG error variation is shown, as a function of the spatial discretization. It is possible to see in Figure 4.1a that the error of the solution decreases, increasing the mesh refinement. The same behavior is observed in Figure 4.1b for  $fe_{degree}$ . This behavior is logical

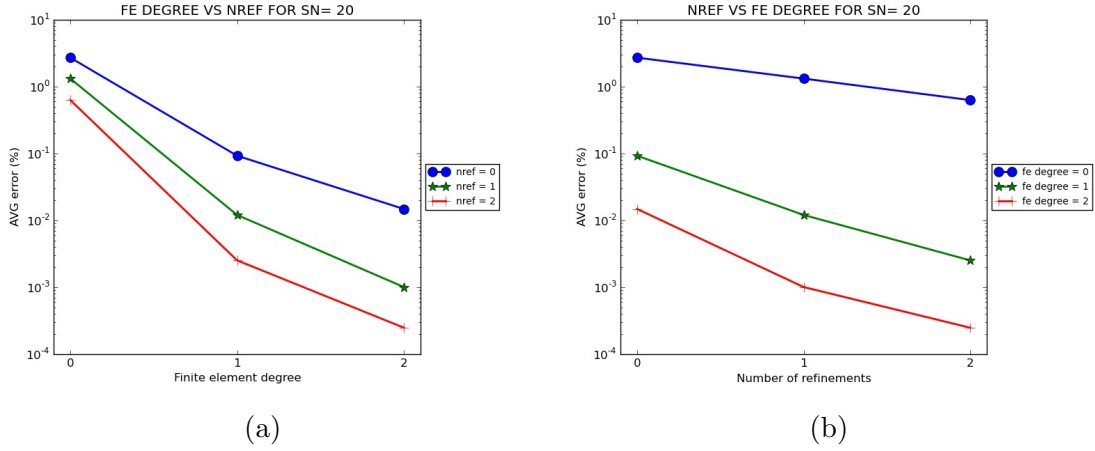


Figure 4.1: Variation of the AVG error for the ASSEMBLY-VOID problem as a function of  $n_{ref}$  and  $fe_{degree}$ , for a fixed  $S_N$  order (Ordinates axis is in  $\log_{10}$  scale).

because increasing these parameters, the spatial discretization is being improved.

As we can see in graphic 4.1a, for  $n_{ref}=0$ , results have a large amount of error. This occurs because  $n_{ref}=0$  is the coarsest spatial discretization, so it does not allow us to have a high accuracy of the calculations. Comments on the behavior for  $fe_{degree}=0$ , here given for Figure 4.3, where it is easier to explain.

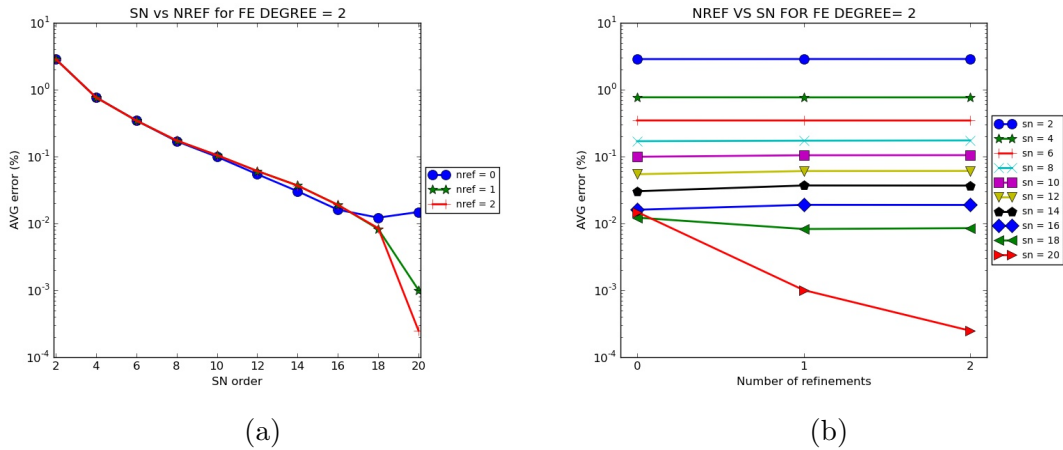


Figure 4.2: Variation of the AVG error for the ASSEMBLY-VOID problem as a function of  $S_N$  order and  $n_{ref}$ , for a fix  $fe_{degree}$  (Ordinates axis is in  $\log_{10}$  scale).

In figure 4.2, the AVG error is represented, for different  $fe_{degree}$  and different angular discretizations. From the graphics 4.2a and 4.2b,

we can see that increasing the  $S_N$  order improves the accuracy of the results considerably, so the part of the error coming from the angular discretization is being reduced with every higher  $S_N$  order. Besides, we can also see that when  $fe_{degree}$  is fixed to 2, curves for  $n_{ref}=1$  and  $n_{ref}=2$  are overlapping, that means that from  $fe_{degree}=2$ , and  $n_{ref}=1$  the spatial error is lower than the angular. As we see in 4.2a it is only when the  $S_N$  order is very high ( $S_N$  order=18), that the spatial error is more important than the angular one. This is due to the fact that for higher  $S_N$  orders, the angular error has been very much reduced, to a point where the spatial error becomes dominant.

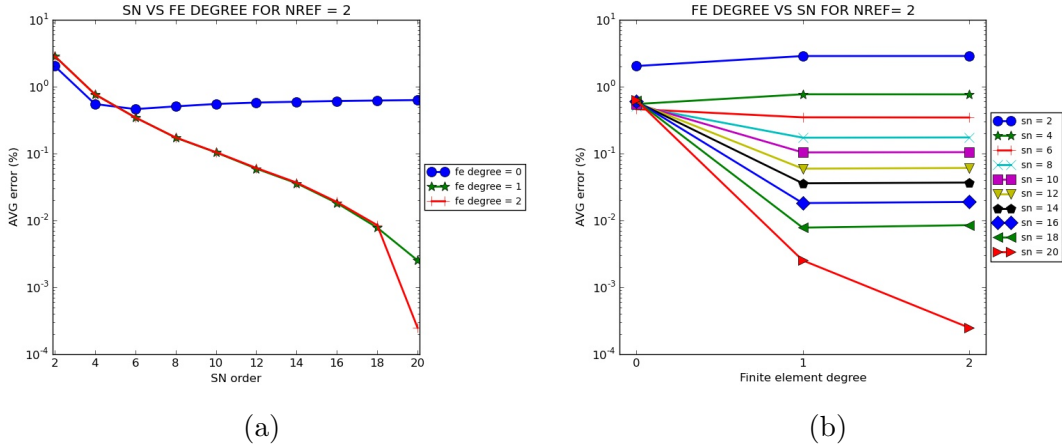
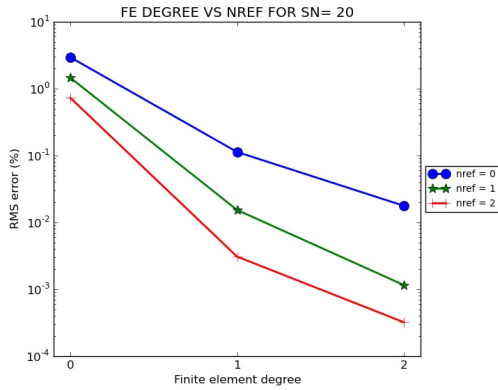


Figure 4.3: Variation of the AVG error for the ASSEMBLY-VOID problem as function of  $S_N$  order and  $fe_{degree}$ , for a fix  $n_{ref}$  (Ordinates axis is in  $\log_{10}$  scale).

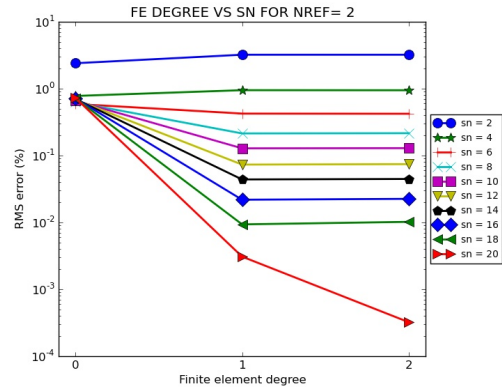
Figure 4.3 shows the AVG error, for different spatial and different angular discretizations. On one hand, we can observe a similar behavior to the one observed in 4.2a. In this case, for a maximum value of  $n_{ref}$ , we find the same values for  $fe_{degree}=1$ , and  $fe_{degree}=2$ . This occurs for the reason explained before, that is, for  $n_{ref}=2$  and  $fe_{degree}=1$ , the error coming from the spatial discretization is less important than the one from the angular discretization, which is the dominant one.

On the another hand, it is also possible to analyse the bad results given by the simulations with  $fe_{degree}=0$ . In Figure 4.3a, we see that results are not logical, that is, the error stay constant or increases instead of decreasing when a high  $S_N$  order value is given. In Figure 4.3b we can observe the improvement when going from  $n_{ref}=0$  to

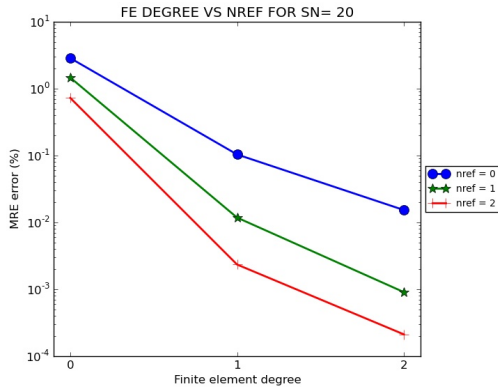
$n_{ref}=1$  or  $n_{ref}=2$ . The explanation of this behavior for  $fe_{degree}=0$ , is that the solution is approximated by a zero-degree polynomial, which give wrong approximations, thus, the results given by  $fe_{degree}=0$  are not interesting. Consequently, in the next two problems the results obtained the by  $fe_{degree}=0$  simulations are discarded.



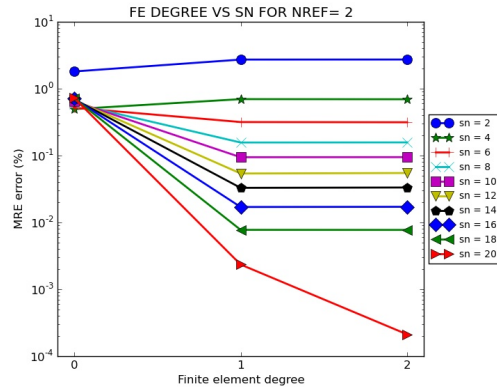
(a) RMS error for  $S_N=20$



(b) RMS error for  $n_{ref}=2$



(c) MRE error for  $S_N=20$



(d) MRE error for  $n_{ref}=2$

Figure 4.4: Comparison of the tendency in the RMS, and MRE error in the ASSEMBLY-VOID problem (Ordinates axis is in  $\log_{10}$  scale).

In Figure 4.4, the RMS and the MRE errors are represented, while in Figure 4.1a and 4.3b we can find the AVG error. Comparing figures 4.1a, 4.4a, 4.4c, and figures 4.3b, 4.4b, 4.4d, we realize that for these three variables the solver behaves identically, that is, the value of the error is different but tendency of the three errors is the same. As a consequence, from now only the AVG error is analyzed.

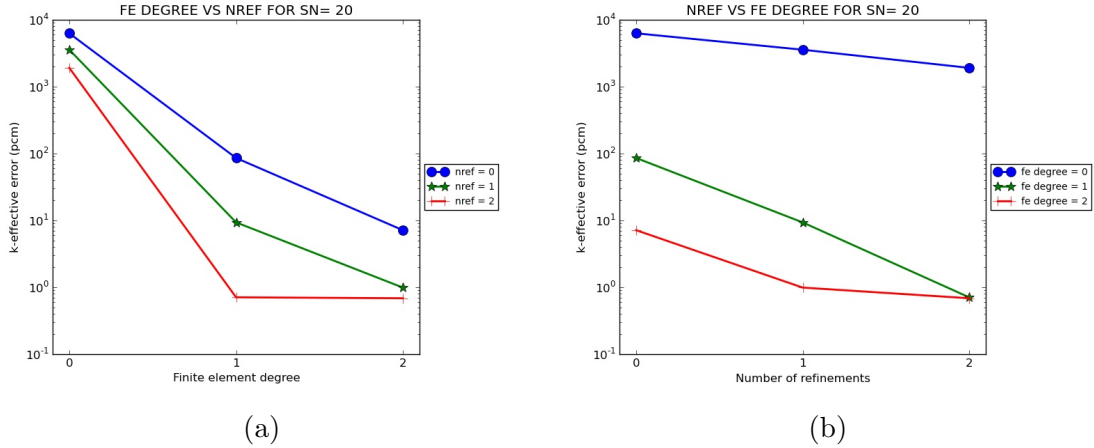


Figure 4.5: Variation of the k-effective error in pcm for the ASSEMBLY-VOID problem as function of the  $n_{ref}$  and the  $fe_{degree}$ , for a fixed  $S_N$  (Ordinates axis is in  $\log_{10}$  scale).

Figure 4.5 shows the absolute error for the k-effective for the different spatial discretizations. We can see that, as happens with the AVG error, the accuracy given by the solver depends directly on the values of the parameters of discretization. In figures 4.5a and 4.5b we observe that, the k-effective is closer to the reference value when higher are the spatial parameters of discretization. We can also see that simulations for values of  $fe_{degree}=1$  and  $n_{ref}=2$ , and for values of  $fe_{degree}=2$  and  $n_{ref}=1$ , give almost the same results as the ones obtained by  $fe_{degree}=2$  and  $n_{ref}=2$ , so the maximum accuracy is obtained when one of the two parameters has its maximum value.

Figure 4.6 shows the absolute error of the k-effective for different  $n_{ref}$  and angular discretizations. We can see in 4.6a that the accuracy in the k-effective is not improved increasing the  $n_{ref}$ , after  $n_{ref}=1$ . Regarding the angular directions, we can see that it is from  $S_N$  order=12 that the behavior turns in a constant tendency, that is, when the absolute error is reduced by increasing the angular directions. Then, from Figure 4.6b it is shown that by improving the  $S_N$  order, the error in the k-effective is reduced.

In Figure 4.7 the absolute error is shown, for different  $fe_{degree}$  and angular discretization. It is possible to observe in Figure 4.7a that the tendency of the error is similar to the one seen in Figure 4.6a

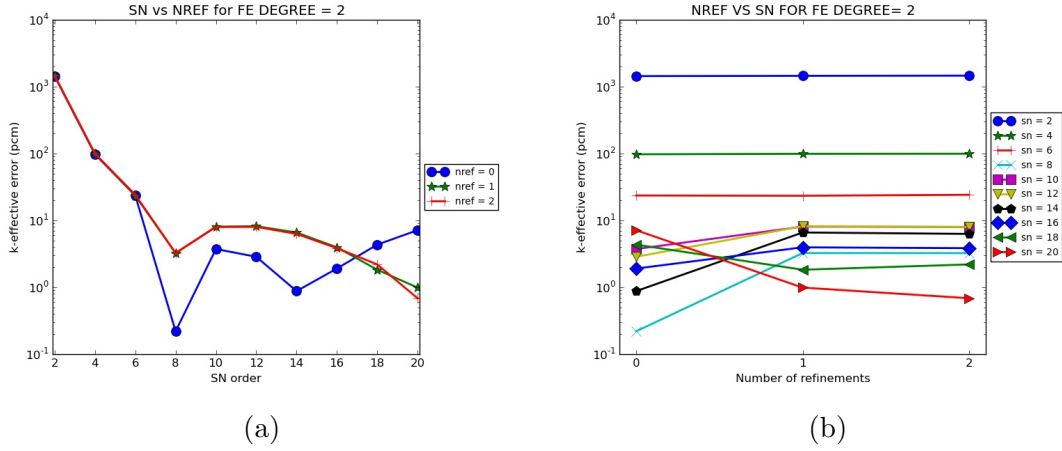


Figure 4.6: Variation of the k-effective error in pcm for the ASSEMBLY-VOID problem as function of the  $S_N$  and the  $n_{ref}$ , for a fixed  $fe_{degree}$  (Ordinates axis is in  $\log_{10}$  scale).

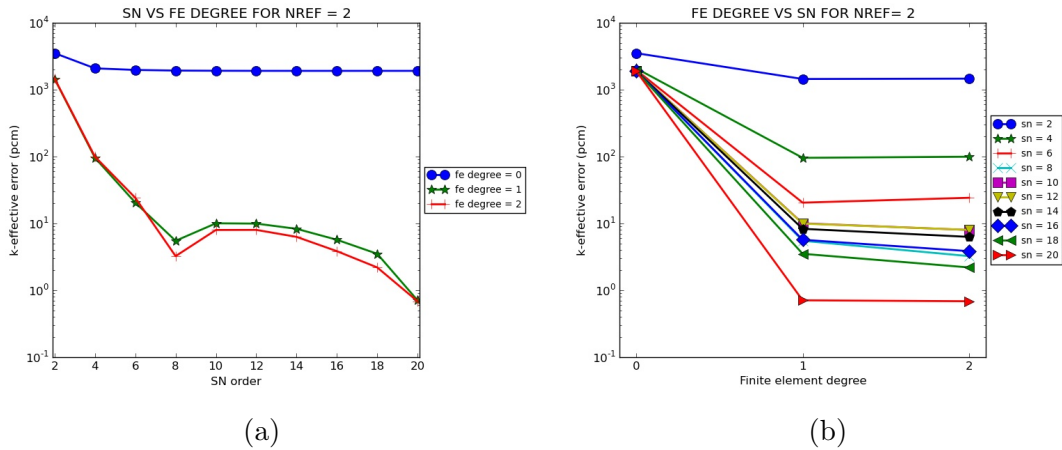


Figure 4.7: Variation of the k-effective error in pcm for the ASSEMBLY-VOID problem as function of the  $S_N$  and the  $fe_{degree}$ , for a fixed  $n_{ref}$  (Ordinates axis is in  $\log_{10}$  scale).

for  $n_{ref}=1$  and  $n_{ref}=2$ , but in this case there exist an improvement between  $fe_{degree}=1$  and  $fe_{degree}=2$ . For the variation of the angular discretization, shown in Figure 4.7b, the behavior as function of the  $fe_{degree}$ , is the same as for the  $n_{ref}$  in Figure 4.6b, i.e, increasing the  $S_N$  order improve the accuracy on the k-effective.

In this part of the parametric study, it has been proved that for the k-effective, as well as for the pin powers, the higher the  $S_N$  order is,



the more accurate the results become, with the error due to the angular discretization being the dominant one. Moreover, for the spatial discretization, the  $fe_{degree}$  affects more the accuracy in the k-effective than the  $n_{ref}$ , whereas for the pin power values their respective effects are similar.

#### 4.1.2 ASSEMBLY-REFLECTIVE benchmark problem

As was done in the previous subsection, here the graphics for the problem with reflective boundary conditions are presented, but results for  $fe_{degree}=0$  was discarded, for the reason previously given.

In Figure 4.8, the graphics describing the behavior of the solver regarding the AVG error, are presented. Analyzing these figures, we can see that, for reflective boundary conditions, the convergence of the solver behaves in the same way regarding the AVG error as for the ASSEMBLY-VOID problem.

In Figures 4.8a and 4.8b, we observe that increasing  $n_{ref}$  as well as  $fe_{degree}$ , the AVG error is reduced, so the pin powers are closer to the reference. In addition, looking at Figures 4.8d and 4.8f we realize that, as happened before, as long as the  $S_N$  order is increased, the accuracy of the results is improved. Finally, Figures 4.8c and 4.8e, show that in the spatial discretization, when one of the two parameters ( $n_{ref}$  and  $fe_{degree}$ ) is at its maximum value, until the  $S_N$  order does not reach a higher value ( $S_N$  order=18 in 4.8c, and  $S_N$  order=16 in 4.8e), the angular error is still higher than the one coming from the spatial discretization.

In Figure 4.9 the absolute error of the k-effective for the different spatial discretizations is presented. Figure 4.9a shows that for the k-effective, increasing one degree of the  $fe_{degree}$  with  $n_{ref}=1$  or  $n_{ref}=0$  improves the value of the k-effective, while doing the same with  $n_{ref}=2$ , the value of the k-effective obtained is worse. In Figure 4.9b, we see that a better solution is obtained with  $fe_{degree}=1$  than with  $fe_{degree}=2$ , for the higher value of  $n_{ref}$ . These behaviors are not logical, but we

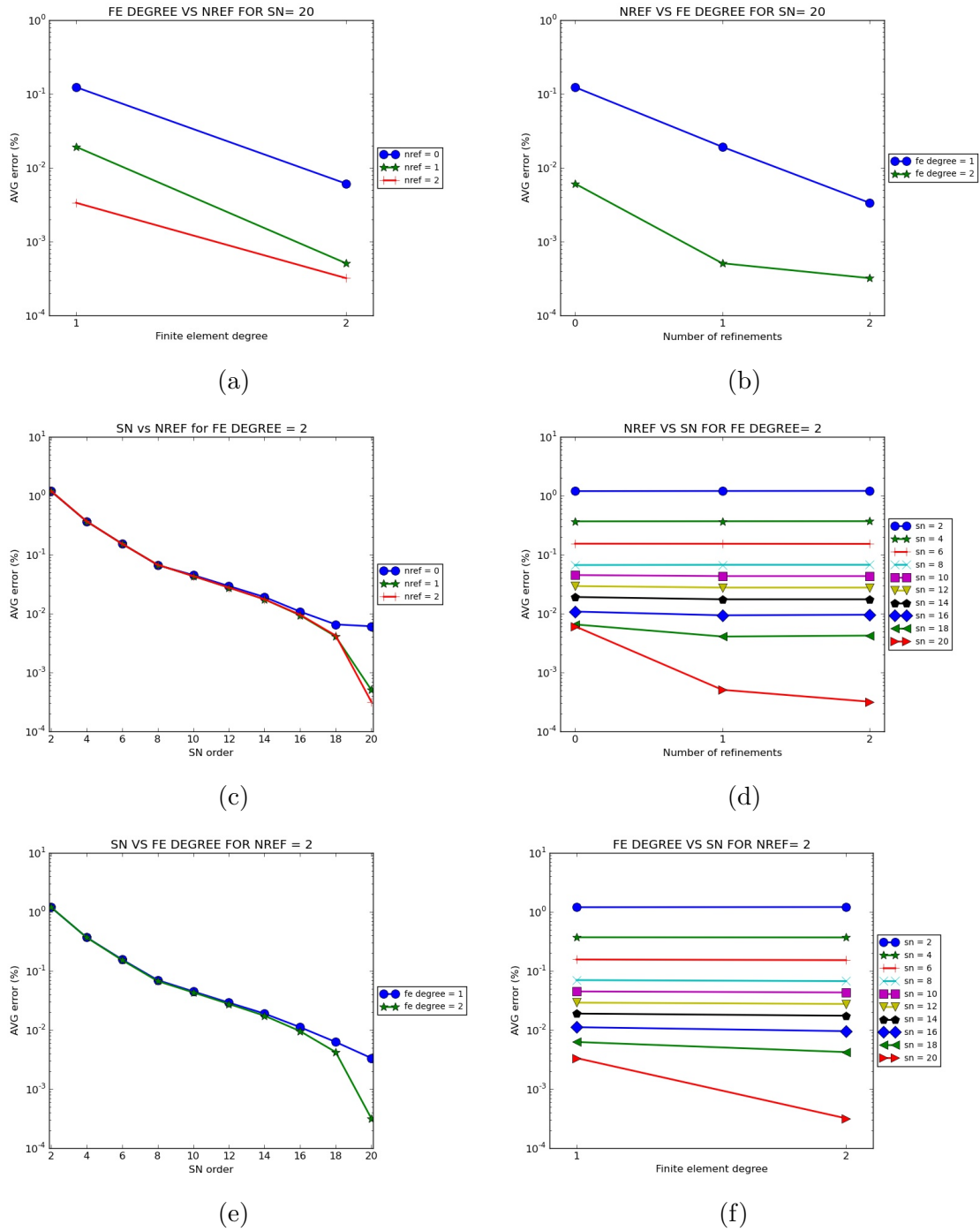


Figure 4.8: Variation of the AVG error for the ASSEMBLY-REFL problem as function of the discretization parameters (Ordinates axis is in  $\log_{10}$  scale).

have to take in account that the range of the differences between the different solutions is very small, in the order of 1 pcm in Figure 4.9a

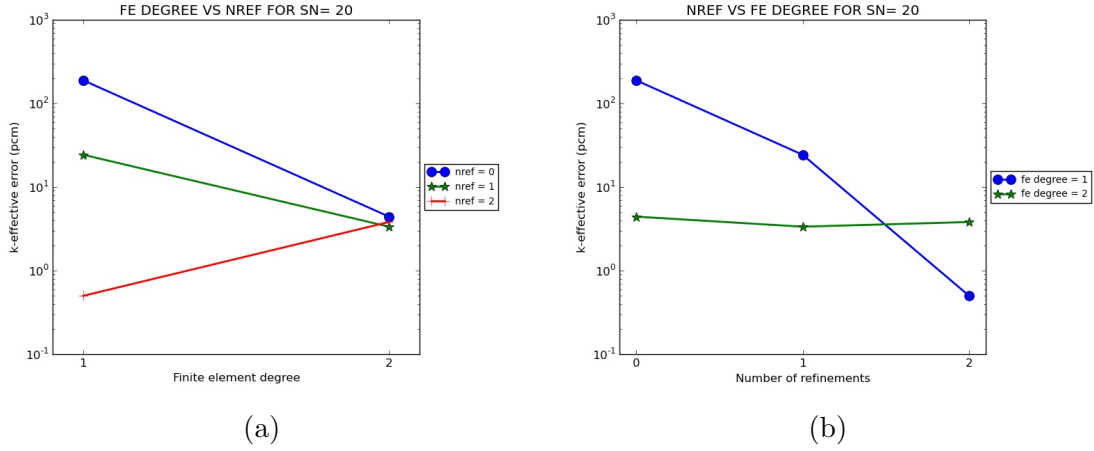


Figure 4.9: Variation of the k-effective error in pcm for the ASSEMBLY-REFL problem as function of the  $n_{ref}$  and the  $fe_{degree}$ , for a fixed  $S_N$  (Ordinates axis is in  $\log_{10}$  scale).

and of 3 pcm in Figure 4.9b. Being these differences as small as they are, it does not mean that FOREST does not behave correctly. The differences can be caused by other factors such as the numerics, or the fact that the spatial discretization method used in DRAGON is Finite Differences, while for FOREST it is Finite Elements.

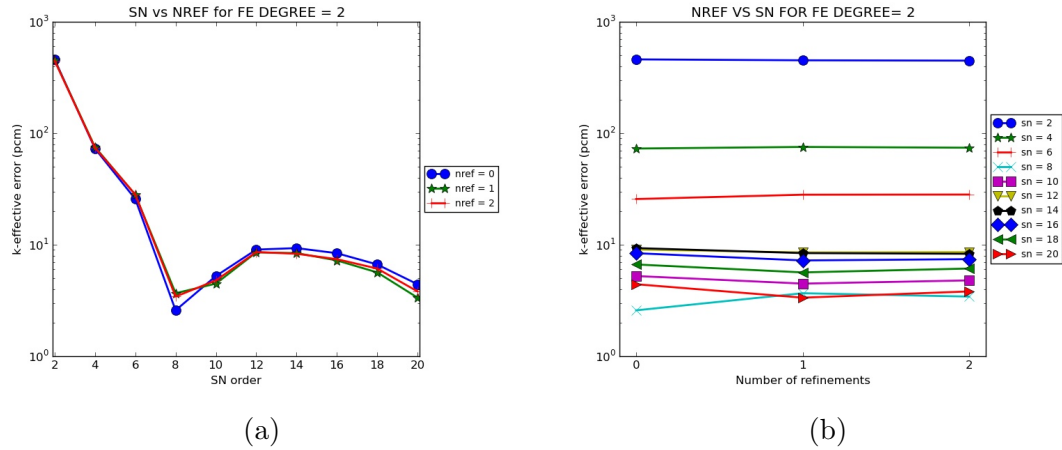


Figure 4.10: Variation of the k-effective error in pcm for the ASSEMBLY-REFL problem as function of the  $S_N$  and the  $n_{ref}$ , for a fixed  $fe_{degree}$  (Ordinates axis is in  $\log_{10}$  scale).

Figure 4.10 shows the evolution of the k-effective for different  $n_{ref}$ , and different  $S_N$  orders. In Figure 4.10a, we can see that for a maximum

$fe_{degree}$ , the value of the k-effective does not change considerably for the different  $n_{ref}$ . However, it is possible to see that for the coarser refinement ( $n_{ref}=0$ ), values are less accurate than in the other two refinements. Also, as happened for void boundary conditions, we observe that is when the  $S_N$  is higher than 16, that we start to see a difference between the curve for  $n_{ref}=1$  and  $n_{ref}=2$ . This is explained by the angular error becoming less important than the spatial error. In Figure 4.10b, we see that the higher the  $S_N$  order is, the more accurate the result are, showing the same behavior as what we found for the previous problem.

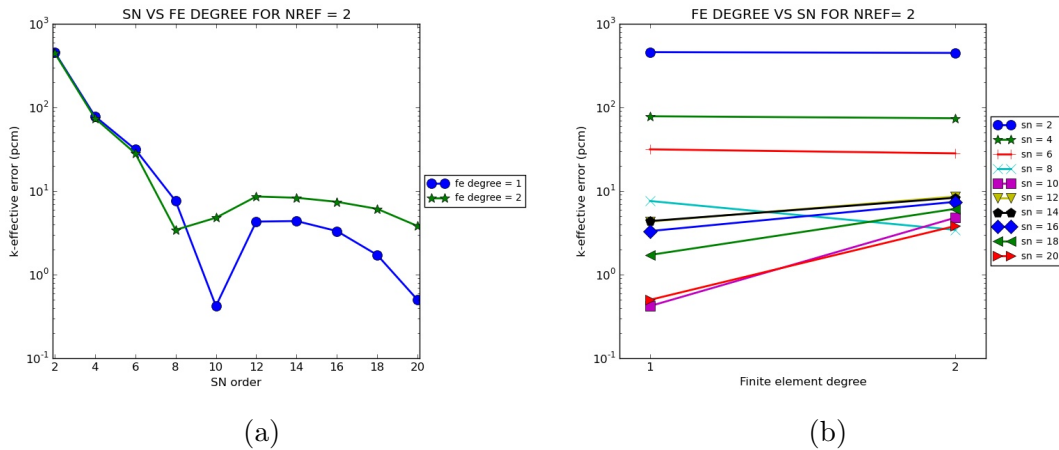


Figure 4.11: Variation of the k-effective error in pcm for the ASSEMBLY-REFL problem as function of the  $S_N$  and the  $fe_{degree}$ , for a fixed  $n_{ref}$  (Ordinates axis is in  $\log_{10}$  scale).

Finally, in Figure 4.11, we can study the variation of the k-effective for different  $fe_{degree}$  and different angular discretizations. In Figure 4.11a, we can see that from  $S_N=8$  to  $S_N=20$ , the behavior for the two values of the  $fe_{degree}$  is not usual because more accuracy is obtained for a lower  $fe_{degree}$ . Looking at Figure 4.11b, we realize that for the angular discretization, even if the global behavior is logical, there are some exception as for example for  $S_N=8$  and  $S_N=10$ , where the obtained values are more accurate than for the other finer angular discretization. It can be explained because of the global meaning of the k-effective, improving due to the cancellations of local errors.

From the graphics studied in this subsection it is seen that the behavior regarding the pin power error is similar to the problem with void boundary conditions. Regarding the k-effective accuracy, similarities and differences have also been found. We have observed, that the global behavior for the angular discretization was close between the problems with different boundary conditions.

Simulating with reflective boundary conditions, the mathematical problem to solve is more challenging than with void boundary conditions. It has been seen that for the pin power distribution, even if the problem became difficult, the solver has given good results.

### 4.1.3 C5G7 Benchmark problem

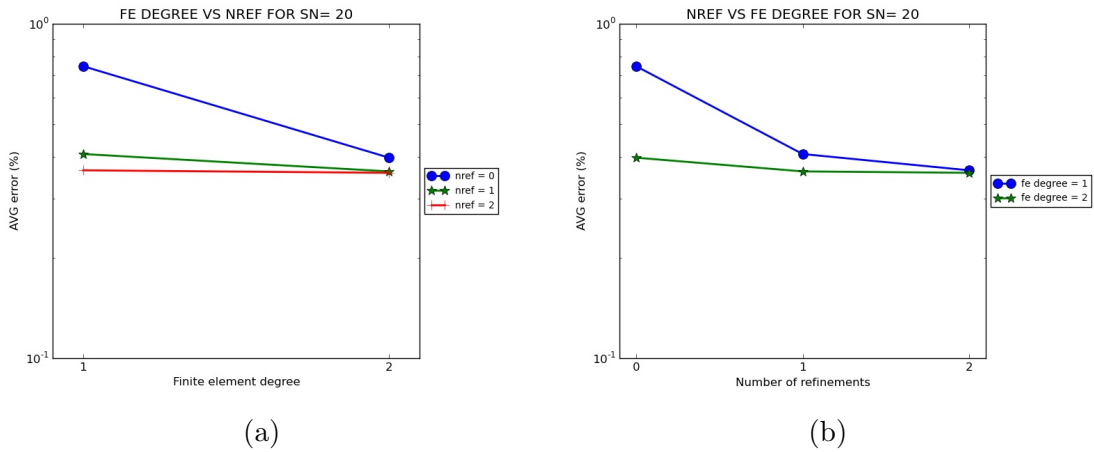


Figure 4.12: Variation of the AVG error for the C5G7 problem as function of the  $n_{ref}$  and the  $fe_{degree}$ , for a fixed  $S_N$  (Ordinates axis is in  $\log_{10}$  scale).

Analyzing the graphics in Figure 4.12, we can see that for a more heterogeneous problem, the results exhibit the same tendencies as in Figure 4.1. However, the value of the AVG error is higher for this problem and the improvement of the accuracy is lower than for the two other problems. Looking at Figure 4.12a and at Figure 4.12b, we see that for  $S_N=20$  and respectively, at least,  $fe_{degree}=1$  and  $n_{ref}=1$ , there is not a considerable improvement of the accuracy by increasing the values of the discretization parameters.

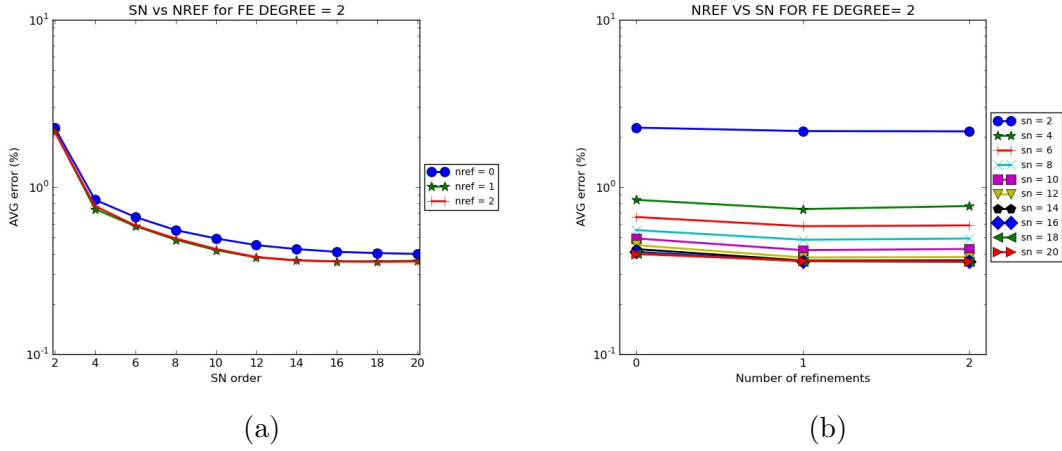


Figure 4.13: Variation of the AVG error for the C5G7 problem as function of the  $S_N$  and the  $n_{ref}$ , for a fixed  $fe_{degree}$  (Ordinates axis is in  $\log_{10}$  scale).

Figure 4.13a shows that for the C5G7 problem, the simulations with the maximum  $fe_{degree}$  value, for  $n_{ref} = 1$  and  $n_{ref} = 2$ , give the same results. In addition, in Figure 4.13b there is an improvement in the AVG error when the  $S_N$  order changes from 2 to 8, but for values higher than 8, the accuracy of the results does not improve considerably.

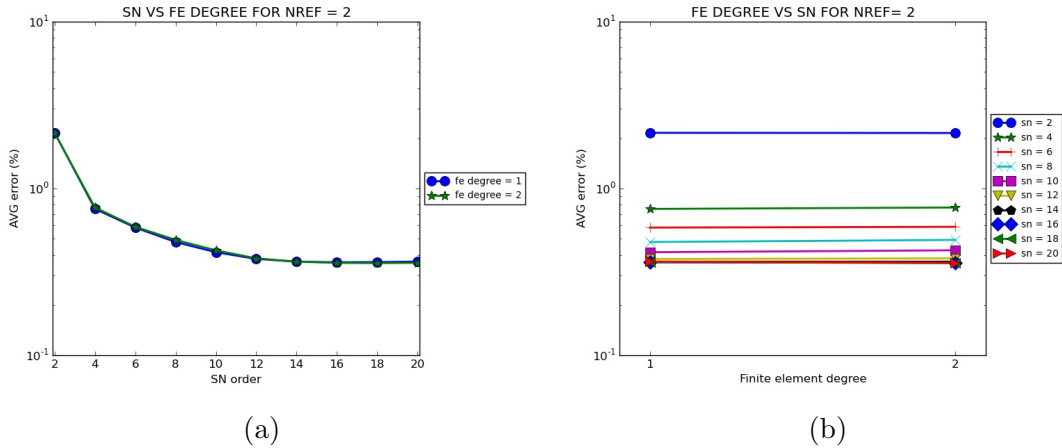


Figure 4.14: Variation of the AVG error for the C5G7 problem as function of the  $S_N$  and the  $fe_{degree}$ , for a fixed  $n_{ref}$  (Ordinates axis is in  $\log_{10}$  scale).

In Figure 4.14 we can see that for simulations with the maximum  $n_{ref}$  value, the same results are obtained with  $fe_{degree} = 1$  and  $fe_{degree} = 2$ . With this information and the one extracted from 4.13a, we see that when the maximum value of one of the spatial parameters of dis-

cretization is reached, the other is not important for the final results. In Figure 4.13b it is also seen, that for  $n_{ref}=2$ , from  $S_N$  order=12 to  $S_N$  order=20, the AVG error does not improv.

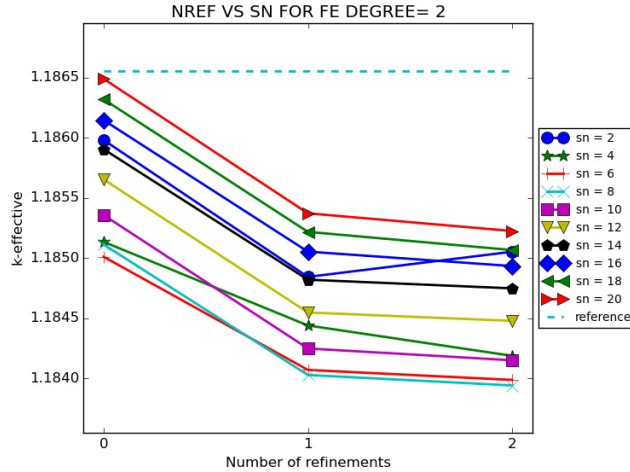


Figure 4.15: Values of the k-effective for different spatial and angular discretization.

In figure 4.15, the behavior of the k-effective value is presented, as a function of the  $n_{ref}$  (0,1,2) and the  $S_N$  (from 2 to 20), together with the reference eigenvalue (1,18655) shown as dashed a line. We can see that for the lowest spatial refinement, the k-effective obtained are closer to the reference. We also observe that the better the angular discretization is, a better solution is reached, and for the maximum  $S_N$  order the k-effective value is 1.18524, that is, 131 pcm from the reference. There are some exceptions, such as for  $S_N$  order=2 or  $S_N$  order=4, where we can find a strange behavior. Looking at the global behavior for the values, results are close to the ones presented on [20], where the benchmark problem has been solved with the discrete-ordinates code DORT, so FOREST behaves as other codes using the same discretization methods. It is also possible to see in [20] that the k-effective results are improved with different quadratures.

#### 4.1.4 Discussion of the most accurate approximations

In this subsection, the results obtained for the most accurate solution, which are for every problem the simulations corresponding to  $n_{ref}=2$ ,  $S_N$  order=20 and  $fe_{degree}=2$ .

	Void boundary conditions	Reflective boundary conditions	C5G7
Reference eigenvalue	0.30713	1.31127	1.18655
Eigenvalue	0.30714	1.31125	1.18524
Absolute error (pcm)	-1	2	131
Per cent error (%)	-0.00193	0.00165	0.09543

Table 4.1: K-effective results.

Table 4.1 shows the best values of the k-effective for every problem. We can observe that the values obtained for the void and the reflective ASSEMBLY problems have errors of 1 and 2 pcm respectively, which is a very small value. For the C5G7 problem, there is an error value of 131 pcm with the reference. It is logical to have a higher value than the one obtained comparing with DRAGON, because MCNP represents the transport behavior more accurately than a deterministic method. Besides the different methods of discretization, because of a more heterogeneous problem, the convergence of the k-effective is more difficult to reach. Other discrete-ordinates codes, such as DORT [20], reached a maximum accuracy of 143 pcm, so FOREST results agree with other codes for the C5G7 benchmark.

	Void boundary conditions	Reflective boundary conditions	C5G7
AVG(%)	0.000249	0.000320	0.358416
RMS(%)	0.000322	0.000395	0.542631
MRE(%)	0.000212	0.000329	0.252443

Table 4.2: Pin power distribution error measures for the three problems.

Table 4.2 shows the values of the AVG, RMS and MRE errors for the most accurate solutions of the problems. Analyzing this table we can study several aspects of the pin power error distribution.



Firstly, regarding the AVG error, solutions for the ASSEMBLY problem are more accurate than the one of the C5G7 problem. However the three solutions have a good accuracy.

Secondly, the RMS error of the two ASSEMBLY problems is closer to their AVG error, whereas there is an important difference between these two errors in the C5G7 problem. This shows that for the ASSEMBLY problems, the range of pin power errors is close to the average pin power value, whereas for the C5G7 problem this range is larger, so there is more heterogeneous distribution of the pin errors.

Finally, looking at the MRE error, distributions of the pin power errors are different in the three problems. We can see that for the ASSEMBLY problem with reflective boundary conditions, the value of the AVG and MRE error are similar. That means that, in this problem, power pins with closer values to the average pin power value, are where the power pin error is higher. For the other two problems, it is possible to see that MRE error is lower than the AVG error. That shows that, specially for the C5G7, the power pin errors are more concentrated in the pins that have lower pin power values.

It is worth to notice that the MCNP calculation of the AVG, RMS and MRE errors, respectively led to of 0.32, 0.34 and 0.27 [2]. Hence, there is the same behavior using MCNP than in FOREST, but FOREST has a larger magnitude.

In table 4.3 values of the maximum and minimum power pin errors are given, as well as the maximum relative error obtained for every problem. Analyzing this table, we can confirm that informations given by MRE error in table 4.2 are correct.

As we can see, for the ASSEMBLY problem with void boundary conditions, the error obtained in its minimum pin power is higher than the one obtained in the maximum pin power, and also it is the maximum calculated pin power error. In addition, it is more clear in the C5G7 problem, where the error in the minimum pin power error is more than 10 times higher than the one in the maximum power pin. These two observations confirm that the information given by the comparison of

	Void boundary conditions	Reflective boundary conditions	C5G7
Max. reference pin power	2.05385	1.25611	2.49788
Maximum pin power	2.05390	1.25610	2.49296
Per cent error (%)	-0.00224	0.0012	0.19695
Min. reference pin power	0.43165	0.88236	0.23152
Minimum pin power	0.43170	0.88240	0.23660
Per cent error (%)	-0.01048	-0.00445	-2.19596
Maximum per cent error (%)	-0.01048	-0.00534	-2.57690

Table 4.3: Per cent error results for specific pin powers.

the AVG and the MRE error are correct. There are no relevant errors for the ASSEMBLY problem with reflective boundary conditions in its maximum and minimum power pin values, as table 4.2 also shows.

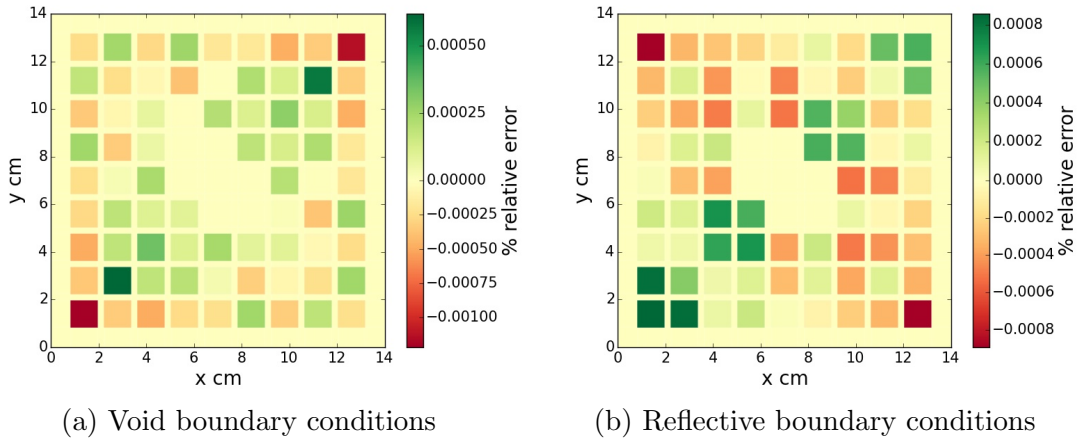


Figure 4.16: Distribution of the relative error per pin in the ASSEMBLY problems.

In Figure 4.16, distributions of errors for void and reflective ASSEMBLY problems are shown. We observe that the two pin power distribution for the two test cases are very close to the one given by DRAGON. The highest errors are located at the corners, and also in the center for the test case with reflective boundary conditions, but the results remain very close to the reference.

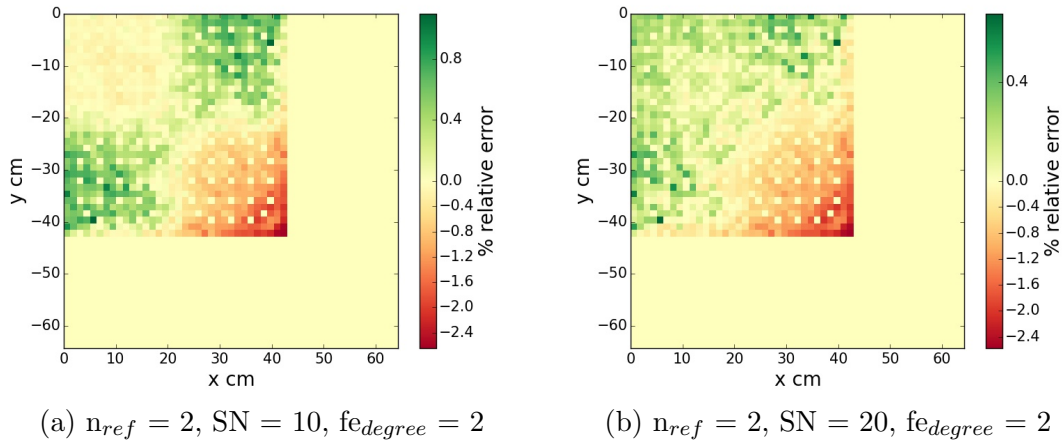


Figure 4.17: Distribution of the relative error per pin in the C5G7 benchmark problem for two different simulations.

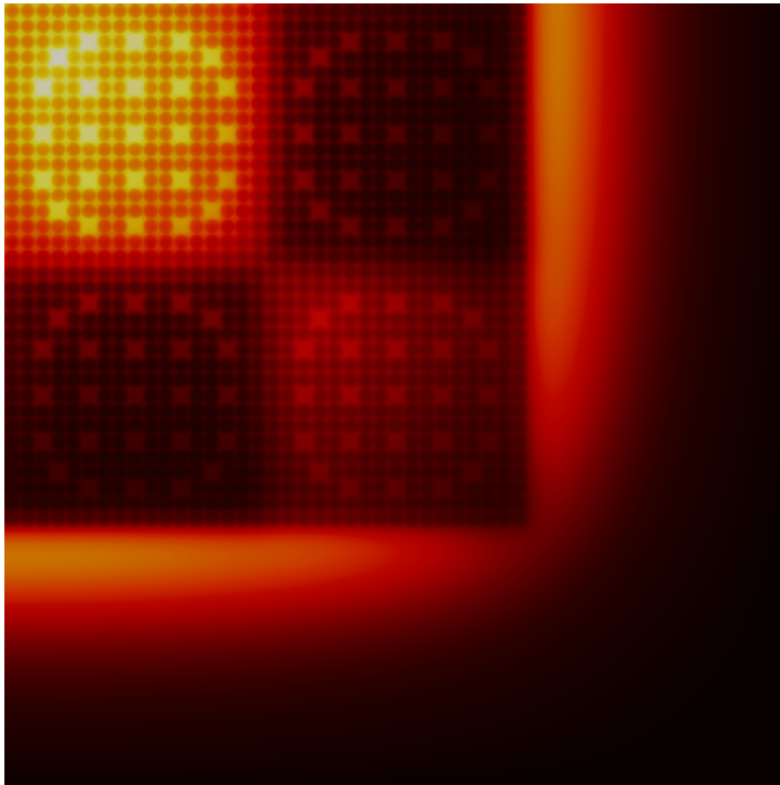


Figure 4.18: Neutron flux distribution for the C5G7 benchmark, for energy group  $g=7$ .

In Figure 4.17, two distributions of the relative power pin error for different angular discretization but the same spatial discretization are

represented . We can see that the range of the relative error between Figure 4.17a and Figure 4.17b has not been considerably improved, even if there is a difference of 10  $S_N$  order for the angular discretization. Different quadratures should be tested in order to circumvent this problem.

It also shown that the maximum errors are located in the assembly of  $UO_2$  which is in contact with the moderator. These results are logical because, as we observe in Figure 4.18, it is in the border between fuel and moderator that the most important differences in the neutron flux are present. For this reason, there is a worse approximation for the neutron flux, so the accuracy in the pin power calculations is also worse than in any other part of the core where the solution is smoother.

In [6], the results for the same problem are given, using a code with MOC as method of discretization. We observe that the same neutron flux distribution, as well as the same distribution for the pin power error are obtained. Hence, results obtained by FOREST are also in agreement, but having a worse accuracy in the pin power calculation, as well as in the k-effective.

## 4.2 Conclusions

On the ASSEMBLY problems, it has been seen that FOREST gives very accurate results, in the order of  $10^{-2}$  and  $10^{-3}\%$  of pin power relative error and it also behaves correctly. Hence, the verification process led to very good agreement with the reference, so it has been proved that FOREST correctly solves the mathematical model. However, the parametric study has also proved that the best accuracy was reached when  $n_{ref}$  or  $fe_{degree}$  has its highest value, suggesting that the error due to the angular approximation is dominant for a wide range of values.

In the process of validation it has also been proved that for more heterogeneous problem, as the C5G7 benchmark problem, the results obtained and the behavior of the solver are not as good as in the

ASSEMBLY problems. Even if FOREST behaves in the same way as the code DORT for the k-effective, and as OpenMOC for the pin power distribution, the best simulation has a divergence of 131 pcm for k-effective and a maximum pin power relative error of 2.5%, values that are higher than for these codes. In addition, when the problem is solved for a relative low value of  $S_N$  order, the maximum accuracy is reached.

With these results it is possible to conclude that the code has been verified, but there should be some modifications in the discretization quadrature, in order to correct the limitations in the angular discretization for more heterogeneous problems.

# Chapter 5

## Study of the boundary conditions

The computational time used by a higher-order solver in the calculations of a complex and heterogeneous problem, such as the C5G7, is large due to the complexity of the neutron transport equation. Consequently, the whole core is not, generally, solved by higher-order solvers. For a studied reactor, every isolated assembly is solved with a fine-mesh high-order solver, while the whole reactor is solved with a coarse-mesh. Hence, there is a coupling between both solvers, which allows the coarse-mesh solver to approximate the flux over the whole reactor core, by using the homogenized cross-sections obtained from the fine-mesh. As we can see in [21], several iterative transport-diffusion methods can be used for the whole core calculations and different types of boundary conditions, such as albedo or incoming flux, can be considered for the coupling.

In this chapter, for the study of boundary conditions, the problem used has been the C5G7 benchmark problem, with albedo boundary conditions for the fine-mesh solver. It is worth to explain, that there is no coupling in FOREST between fine and coarse meshes. Instead, the boundary conditions are obtained from the spatial and the angular coarsening of the fluxes obtained from the fine-mesh calculations.

In the following, the steps used in order to test how the changes in boundary conditions affect to the final results are explained. Firstly, the pin power of every assembly (see Figure 3.2 in Chapter 3), as

well as the boundary fluxes for every assembly have been obtained, by simulating every problem with the benchmark properties. In our case, we have studied the two  $\text{UO}_2$  assemblies, referring by Inner- $\text{UO}_2$  the one with is not in contact with the moderator, and by Outer- $\text{UO}_2$  the other which is in contact with the moderator. As the C5G7 is a symmetric problem, only one assembly of MOX has been studied, because both of them have the same pin power values. Here, the simulation used as reference has been the one with  $n_{ref}=1$ ,  $S_N$  order=8 and  $fe_{degree}=2$ , so the reference pin power values for the two  $\text{UO}_2$  and the MOX assemblies have been extracted from the pin power reference of the simulation. Then, in order to simulate a coarse-mesh calculation, the boundary conditions of the different assemblies have been approximated for every pin cell size over the boundary of the assembly. Then, for every spatial and angular coarsening level, the albedos obtained are reintroduced in FOREST in order to obtain the pin power distribution of every assembly. Finally, the comparison between the pin power obtained for the different albedo approximation, against the pin power obtained for the reference have been obtained. In this way, it is possible to know how different degradations of the boundary conditions affects the results of FOREST, that is, we can know which coarsening introduces more error in the calculations.

## 5.1 Spatial and angular approximations

The albedo obtained by with the reference is coarsened spatially and angularly. In this subsection, the approximations on the albedo are explained.

### 5.1.1 Spatial coarsening of boundary conditions

From the solution of the simulation with FOREST for the C5G7 problem, the albedo corresponding to every pin cell of the assembly have been obtained. Regarding the spatial coarsening of the albedos, this was by doing the average between the values of different angular fluxes

for adjacent pins, and recalculating the albedos, as it is shown in Figure 5.1. That is, the first approximation has been for one albedo per pin and then, the spatial coarsening continues by using the same albedo for two pins, until the 17x17 assembly is approximated by only one albedo for the whole face of the assembly.

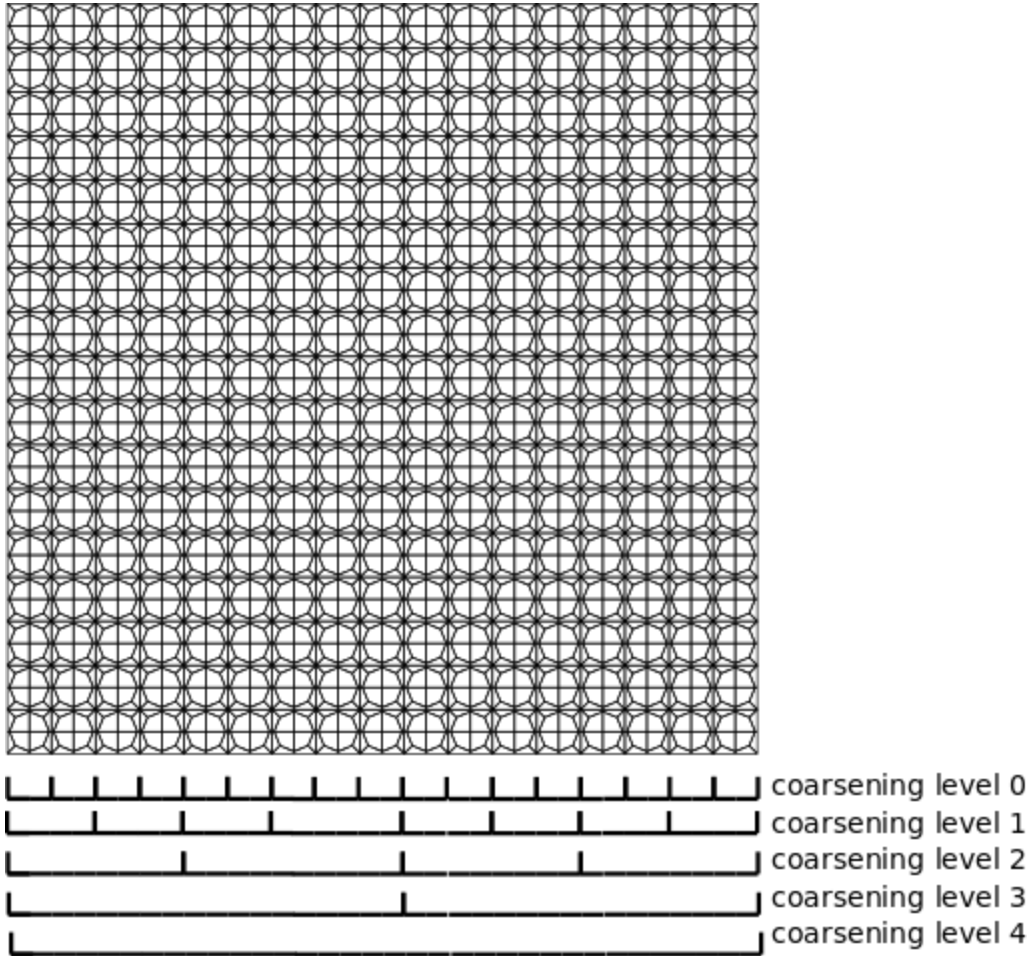


Figure 5.1: Spatial coarsening levels for the albedo

It is worth to notice that for the first spatial approximation, the albedo values coincide with the ones given by the reference, but when these values are reintroduced in FOREST in order to recalculate the power pin values, results will be different from the reference. This is because even if it is the albedo given by the reference, two error have been introduced. The first one comes from considering one albedo per pin cell, since for  $n_{ref}=1$  (as we can observe in 2.1b in Chapter 2), every pin cell is divided in two so there should be two albedos for a better



approximation, and secondly, the other error comes from considering the albedo as a number, when actually the fluxes are represented by polynomials.

Next, in Figures 5.3, 5.4, and 5.5 the variation of the AVG error, as well as the maximum relative error are shown. In this way we can study how the approximation of the boundary conditions affect the accuracy for the pin power results, and which assemblies are more affected by these approximations.

### 5.1.2 Angular coarsening of boundary conditions

The angular coarsening is produced through projection of the angular flux on Spherical Harmonics, then the truncation order of the expansion series of Spherical Harmonics is reduced, and the resulting coarsened flux is then projected back in the quadrature points used for the  $S_N$  approximation.

For the first projection, the Directions-to-moments operator,  $\mathcal{D}$ , is applied to the angular flux defined over a quadrature set,  $\{\Omega_i, \mu_i\}_{i=1}^{N_{\text{dir}}}$ , as follows

$$\phi_l^m(\mathbf{r}) = \mathcal{D}_l^m \Psi(\mathbf{r}) = \sum_{i=1}^{N_{\text{dir}}} \mu_i Y_l^m(\Omega_i) \psi_i(\mathbf{r}). \quad (5.1)$$

where  $\Psi(\mathbf{r})$  is the vector with all the different angular directions,  $[\psi_i(\mathbf{r})] = [\psi(\mathbf{r}, \Omega_i)]$ .

Then, once all the angular moments have been calculated, we have a representation of the angular flux in terms of Spherical Harmonics. For Discrete-Ordinates of order  $N$  ( $S_N$ ) the required order of the Spherical Harmonics is  $L = N/2$ ,

$$\psi(\mathbf{r}, \Omega) = \sum_{l=0}^L \sum_{m=-l}^l \phi_l^m(\mathbf{r}) Y_l^m(\Omega). \quad (5.2)$$

Once the angular neutron flux is expanded in Spherical Harmonics, the coarsening is produced by reducing the truncation order of the

Spherical Harmonics to a smaller value,  $L_c$ , and then projecting in a very natural way to the quadrature set by the Moments-to-directions operator,  $\mathcal{M}$ , as follows

$$\psi_i(\mathbf{r}) = \mathcal{M}_i \Phi(\mathbf{r}) = \sum_{l=0}^{L_c} \sum_{m=-l}^l \phi_l^m(\mathbf{r}) Y_l^m(\boldsymbol{\Omega}_i). \quad (5.3)$$

where  $\Phi(\mathbf{r})$  is the vector of angular moments,  $[\phi_l^m(\mathbf{r})]$ .

In our case, as in this part of the work the simulation done has  $S_N$  order=8, the maximum truncation order of the Spherical Harmonics is  $L = 4$ . The truncation order of the Spherical Harmonics after the angular coarsening of the neutron flux is

$$L_c = L - AC_{Level}, \quad (5.4)$$

where  $AC_{Level}$  is the angular coarsening level. In this way, for the coarsest level ( $AC_{Level}=4$ ) the truncation order of the Spherical harmonics is  $L_c=0$ .

## 5.2 Results

First of all, in Figure 5.2, the distribution of the relative error for the simulation used as reference is shown. In this way, we can observe the existing error in the reference, and later, comparing with the error introduced by the spatial and angular approximations, it is possible to know if the error introduced by coarsening the boundary conditions is canceled with the one of the FOREST. If the error introduced has the same or higher order of magnitude than the one of the reference, the error increases the total error in the pin power results, and if it has a lower magnitude error, the approximations should not affect considerably the results.

From Figures 5.3a, 5.4a, 5.5a, it is possible to observe that for every assembly, the behavior for the different coarsening levels, spatial as well as angular, is very similar for the three assemblies. That is, the higher spatial coarsening level is used, the larger is the error in the pin

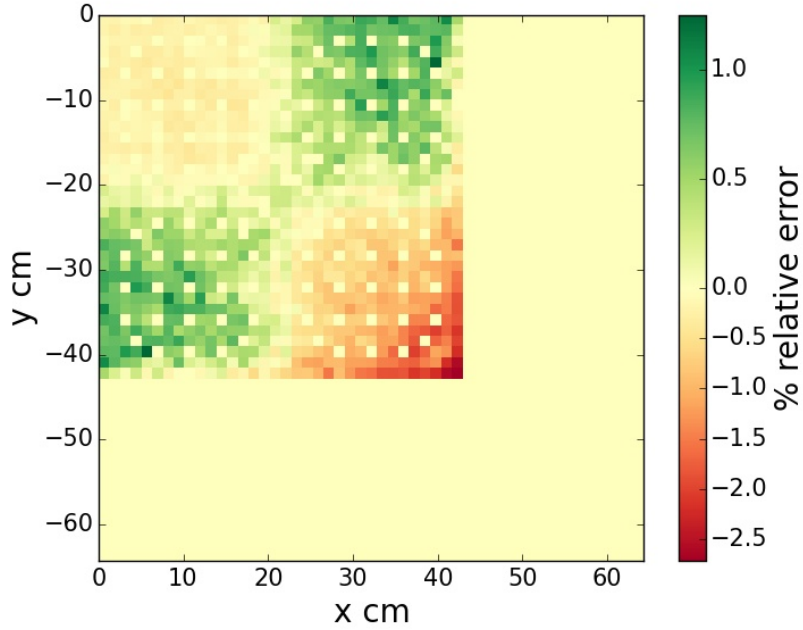
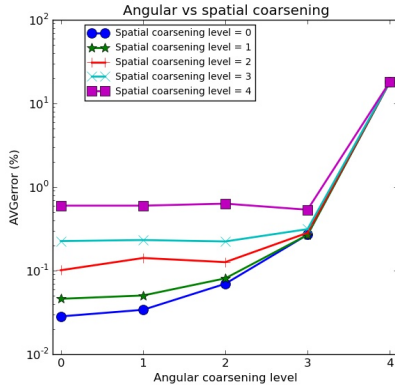
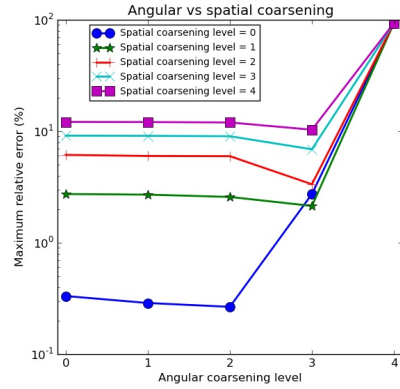


Figure 5.2: Distribution for the relative error for the simulation with  $n_{ref}=1$ ,  $S_N$  order=8 and  $fe_{degree}=2$ .



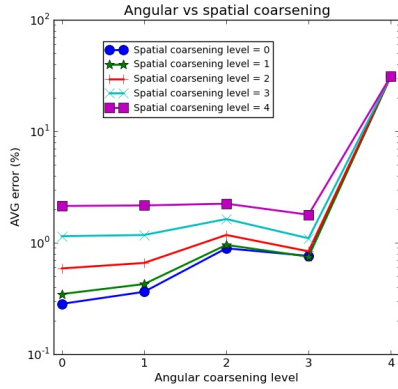
(a) AVG error



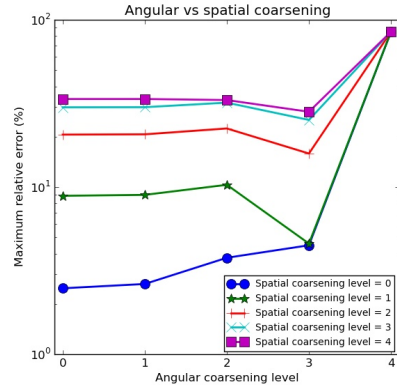
(b) Maximum relative error

Figure 5.3: Variation of the AVG and the maximum relative error in the inner-UO<sub>2</sub> (Ordinates axis is in log<sub>10</sub> scale).

power. For the spatial level 0 and 1, we can see that the differences are not very important for the AVG error, but from level 1 until level 4, this difference increases considerably. This behavior is totally normal, because for a higher coarsening level, the approximation of the albedo becomes worse in every side of the assembly, as it is shown in

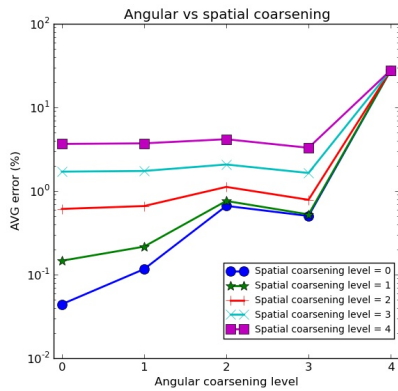


(a) AVG error

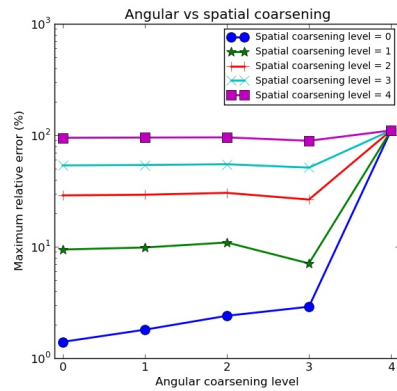


(b) Maximum relative error

Figure 5.4: Variation of the AVG and the maximum relative error in the MOX (Ordinates axis is in  $\log_{10}$  scale).



(a) AVG error



(b) Maximum relative error

Figure 5.5: Variation of the AVG and the maximum relative error in the outer-UO<sub>2</sub> (Ordinates axis is in  $\log_{10}$  scale).

Figure 5.1. Hence, a consequence of using a higher coarsening level is the loss of accuracy in the solution.

We observe that for the angular coarsening, as happened for the spatial one, the AVG error changes in a similar way in the three assemblies, but the behavior is different from before. We observe that for a spatial level lower or equal to 1, there is a loss of accuracy, when the coarsening angular level is increased, whereas for spatial level higher than 1 the AVG error stays practically constant, except for an angular coarsening level equal to 4, representing reflective boundary conditions, where the

accuracy is worse.

In Figures 5.3b, 5.4b, 5.5b, we can see that for the angular and spatial coarsenings, the behavior is also similar in all assemblies. Here, the differences between the various spatial discretizations are more obvious than for the AVG error. Up to the angular coarsening level 3, increasing the spatial coarsening level increases a lot the errors. We also observe that for the inner-UO<sub>2</sub> (Figure 5.3b) the errors for the angular coarsening of level 3 are closer than in the other two assemblies, being the outer-UO<sub>2</sub> (Figure 5.5b) the one with more differences between errors.

We shall make a few remarks on the results of these graphics: firstly, for the maximum angular coarsening level (4), for all the spatial levels, the same values for AVG and maximum relative errors are obtained. This is because for the lower truncation order of the Spherical Harmonics, all the angular fluxes have the same value, which is the average of all the angular fluxes. In this way, all the albedos for the angular coarsening level 4, are equal to 1, whatever the spatial approximation is. Finally, we can see that in some cases, between the angular coarsening level 2, and the level 3, there is a slight improvement of the accuracy. Some particularity of the flux in the C5G7 benchmark problem is responsible for this outer-intuitive behavior.

In Figure 5.6, the distribution of the relative error in the three assemblies is represented. The assemblies have been located in the same way as they are in the C5G7 problem, and a different error scale has been used in every assembly in order to appreciate the errors for each assembly.

It is worth to remember that the C5G7 benchmark problem represents the quarter of a whole core. As a consequence, so where the reflective boundary conditions are applied (albedo=1), the coarsening does not degrade the solution (upper and left side of the inner-UO<sub>2</sub>, and upper side of MOX).

It is possible to see that for the Inner-UO<sub>2</sub> (Figure 5.6a), the distribution of the error is mainly located in its right and bottom sides. This

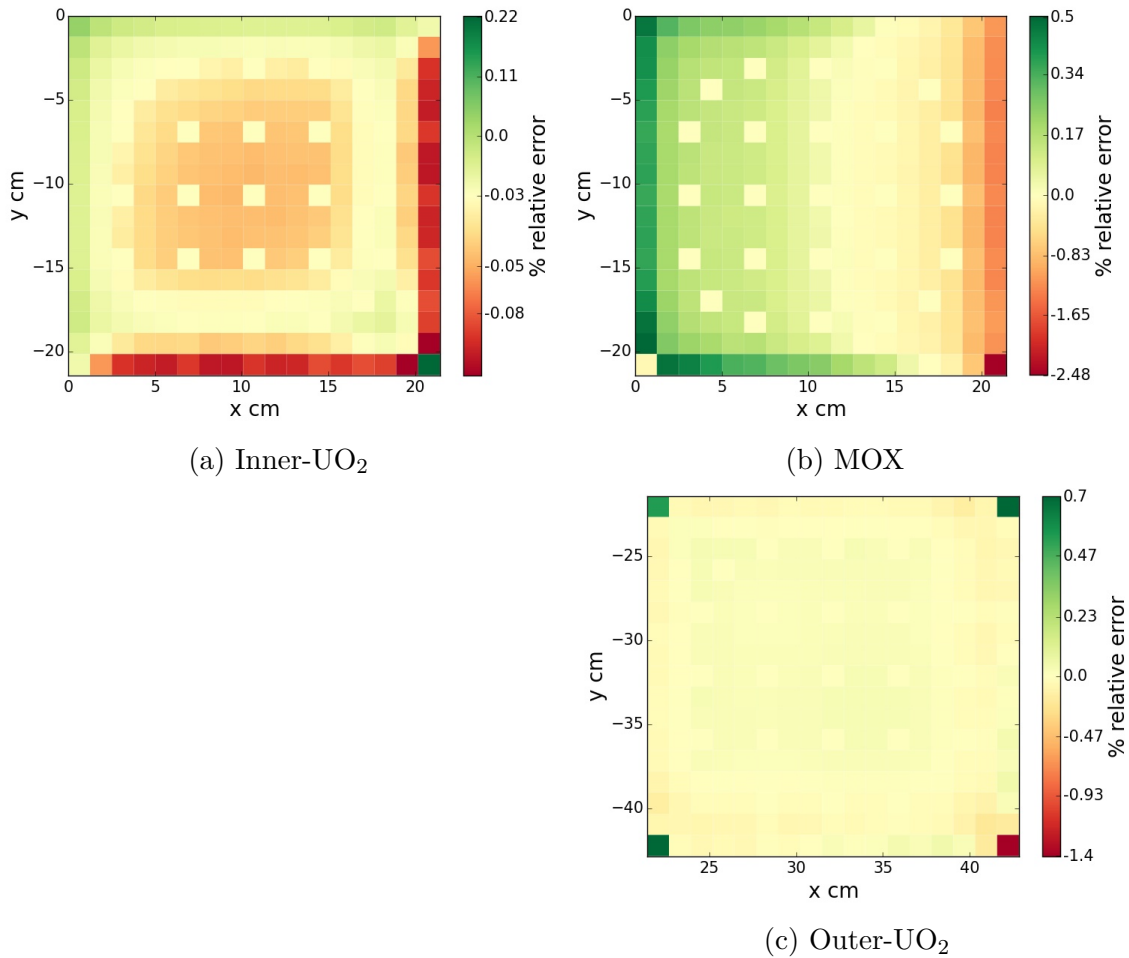


Figure 5.6: Distribution of relative power pin error for the three assemblies, for angular and spatial coarsening level 0.

result is normal, because these sides are in contact with the 2 MOX assemblies, so there exist some strong changes in the flux between these assemblies. We also observe that in its other sides, the error is very low because the adjacent assemblies are of the same type. The corner in the right bottom shows the larger error. It can be due to the strong change of the flux.

For the MOX assembly (Figure 5.6b), something similar occurs. Its upper side is the one with the lowest error because the adjacent side is a MOX assembly, whereas on the other sides, the error is much higher. Here, the highest errors are located in the face which is in contact with the moderator, and it is normal because it is there, that

the neutron flux changes faster (see Figure 4.18) and thus it is more difficult to approximate it. The place where the error is higher is in the corner which is in contact with the moderator and the outer-UO<sub>2</sub> assembly, due to the same reason.

Finally, in the outer-UO<sub>2</sub> (Figure 5.6c) we can observe that is in the corners where the relative error is more important.

For this configuration of the angular and spatial approximation the most important error in every assembly is found in the corners which are in contact with other types of assemblies. This behavior is logical, because it is where there exist bigger differences between the incoming and the outgoing flux. We can see that is in the MOX assembly, where the highest errors are reached. Looking at Figure 5.2, we observe that the maximum error for this simulation is around 2.5%, which has the same order of magnitude as the highest error in the whole core calculation. Thus, the errors introduced by the coarsening affect in an important way to the global results.

To observe how the error changes in every assembly, in Figure 5.7, the distribution of the relative error, for angular and spatial coarsening level 2 is presented. Comparing with in Figure 5.6, the errors in Figure 5.7 are much higher. Increasing the coarsening level gives worse approximations of the albedo, so the results obtained are affected by those approximations. Even if there are some points where the error remains low, in the corners of the assembly, where we have observed that it is harder to approximate the neutron flux, errors are increased one order of magnitude. We also observe than for higher coarsening levels the outer-UO<sub>2</sub> assembly is where the largest errors are found. This assembly is where stronger changes for the flux happen due to the other different types of assemblies, including the moderator, surrounding it.

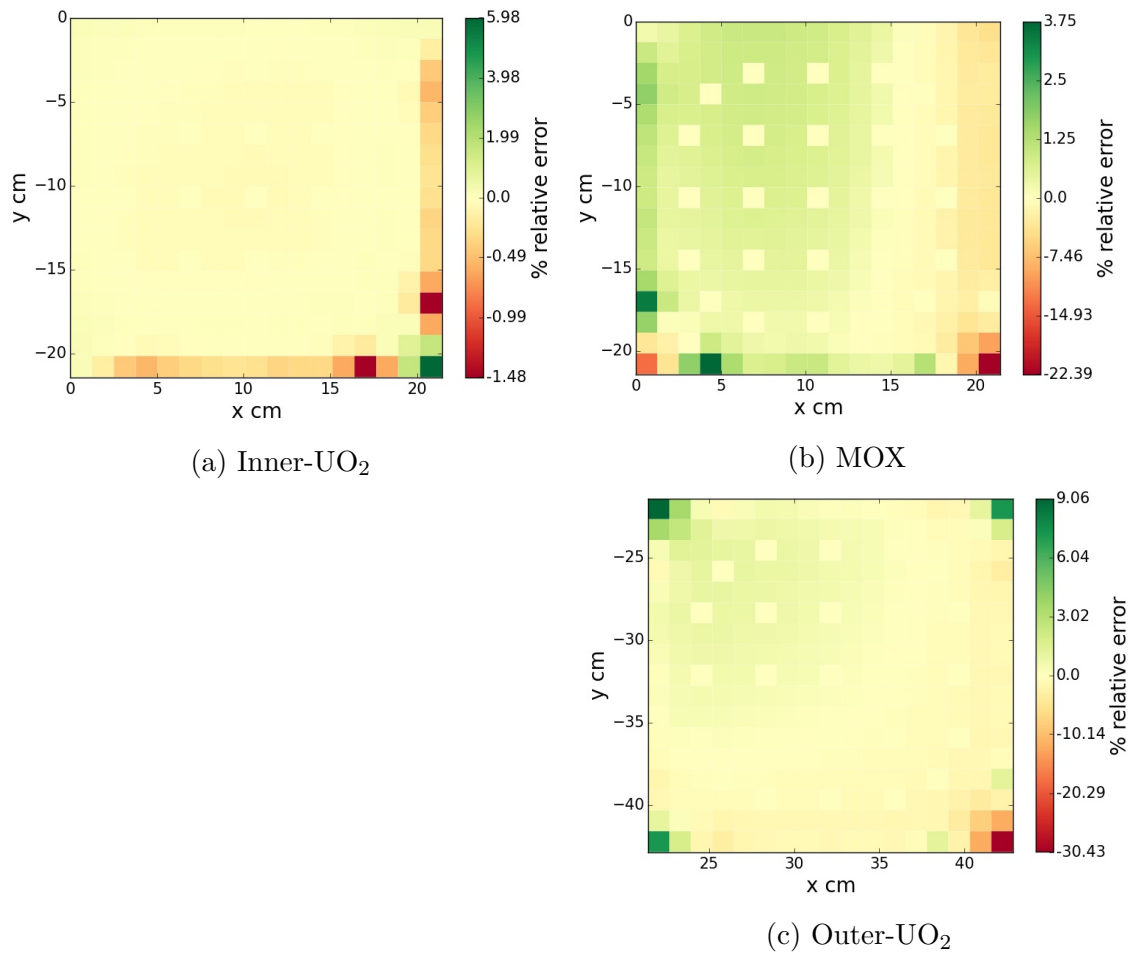


Figure 5.7: Distribution of relative power pin error for the three assemblies, for angular and spatial coarsening level 2.

### 5.3 Conclusions

In this chapter, we have tested how the changes in the boundary conditions affects to the pin power results. From the different graphics and pin power distributions shown above, it has been proved that the coarsening of the boundary conditions increases the errors in calculations. It has been also proved that the spatial coarsening introduces larger errors than the angular one, which effects are considerable for the coarsest level (which is equivalent to reflective boundary conditions).



# Chapter 6

## Conclusions

In the present work, the verification and validation of the code FOREST has been performed. Also, the changes of the boundary conditions for the problem has been studied, simulating an approximative boundary condition obtained from a coupling of low-order/high-order solvers by mean of a coarsening process.

On one hand, the verification process has been carried out satisfactorily. The code FOREST has been tested against DRAGON for a single assembly problem, considering void and reflective boundary conditions. The agreement of the higher -order approximation with the solution provided by DRAGON is of the order of 1 or 2 pcm for the eigenvalue, and around 0.01% for the pin power distribution. We have proved that the code was verified, due to the high agreement of the solution with respect to the one obtained by DRAGON. We conclude that FOREST solves the mathematical model (discrete-ordinates approximation) correctly.

On the another hand, in the process of validation done for the C5G7 problem (which reference is calculated by MCNP), we have observed that the convergence of the k-effective as well as the accuracy in the pin power calculations had to be improved. The actual quadrature used in the angular discretization has to changed in order to correct these limitations. In any case, the results agree with the ones provided in [20] for the convergence of the eigenvalue, and the pin power error

distribution agrees with the one obtained by OpenMOC [6], showing a good behavior when comparing with state of the art codes.

The changes in the boundary conditions have shown to affect the pin power distribution as expected. It has been observed that the spatial accuracy of the boundary conditions is more important, in order to preserve a good approximation when doing an isolated assembly calculation, that the angular accuracy. It is in agreement with the use of a fine-mesh low-order solver for the whole core, instead of a coarse-mesh high-order approximation. Moreover, it has been shown that the isolated assembly calculations with reflective boundary conditions (angular coarsening equal to 4) results in a very bad approximation for the neutron power distribution.

It has been proved here that reflective boundary conditions provides unacceptable results when calculating the power distribution, being one of the most used approaches in todays calculations. Thus, a low-order/high-order coupling methodology should be taken into account in order to improve the accuracy of these calculations.

# Bibliography

- [1] Jeremy A Roberts and Benoit Forget. Multigroup diffusion preconditioners for multiplying fixed-source transport problems. *Journal of Computational Physics*, 274:455–472, 2014. [iv](#), [14](#), [15](#)
- [2] M. A. Smith, E. E. Lewis, and Byung Chan Na. Benchmark on deterministic 2-D MOX fuel assembly transport calculations without spatial homogenization. In *Progress in Nuclear Energy*, volume 45, pages 107–118, 2004. [iv](#), [4](#), [11](#), [16](#), [17](#), [33](#)
- [3] Forrest B. Brown. Fundamentals of Monte Carlo particle transport, 2005. [1](#)
- [4] Paul K. Romano and Benoit Forget. The OpenMC Monte Carlo particle transport code. *Annals of Nuclear Energy*, 51:274–281, January 2013. [1](#)
- [5] G Marleau, A Hébert, and R Roy. A USER GUIDE FOR DRAGON VERSION4. *Technical report IGE-335*, 2014. [2](#), [4](#)
- [6] William Boyd, Samuel Shaner, Lulu Li, Benoit Forget, and Kord Smith. The OpenMOC method of characteristics neutral particle transport code. *Annals of Nuclear Energy*, 68:43–52, June 2014. [2](#), [36](#), [50](#)
- [7] Weston M Stacey. *Nuclear reactor physics*. John Wiley & Sons, 2007. [2](#), [7](#)
- [8] B.D Ganapol. Analytical Benchmarks for Nuclear Engineering Applications: Case Studies in Neutron Transport Theory. *Nuclear Energy Agency*, 2008. [2](#)

- [9] Christophe Demazière. Multi-physics modelling of nuclear reactors: current practices in a nutshell. *International Journal of Nuclear Energy Science and Technology*, 7(4):288–318, 2013. 3
- [10] Xudeng Hang, Zhenying Hong, Shuanggui Li, and Guangwei Yuan. Deterministic numerical methods for particle transport equations. *Jisuan Wuli/Chinese Journal of Computational Physics*, 31(2):127–154, 2014. 3
- [11] Richard Sanchez, Jacques Mondot, Zarko Stankovski, Antoine Cossic, and Igor Zmijarevic. APOLLO II: A user-oriented, portable, modular code for multigroup transport assembly calculations. *Nuclear Science and Engineering*, 100(3):352–362, 1988. 3
- [12] C. A. Wemple, H. N M Gheorghiu, R. J J Stamm’ler, and E. A. Villarino. Recent advances in the HELIOS-2 lattice physics code. In *International Conference on the Physics of Reactors 2008, PHYSOR 08*, volume 2, pages 921–928, 2008. 3
- [13] Ser G. Hong and Nam Zin Cho. CRX : A code for rectangular and hexagonal lattices based on the method of characteristics. *Annals of Nuclear Energy*, 25(8):547–565, 1998. 3
- [14] Giuseppe Palmiotti, C B Carrico, and Elmer E Lewis. Variational Nodal Transport Methods with Anisotropic Scattering. *Nuclear Science and Engineering*, 115:233–243, 1993. 3
- [15] S. Jareteg, K. and Vinai, P. and Sasic, , and C. Demazière. Influence of an SN solver in a fine-mesh neutronics/thermal-hydraulics framework. *Physor*, 2014. 3
- [16] Pierre Lesaint. Finite element methods for the transport equation. *Computer Physics Reports*, 6(1-6):351–369, August 1987. 3
- [17] X-5 Monte Carlo Team. MCNP — A General Monte Carlo N-Particle Transport Code, Version 5. 836, 2003. 4
- [18] Douglas N Arnold, Franco Brezzi, Bernardo Cockburn, and Donatella Marini. Discontinuous Galerkin methods for elliptic prob-

- lems. In B Cockburn, G Karniadakis, and C Shu, editors, *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 89–101. Springer, Berlin, 2000. [7](#)
- [19] W Bangerth, R Hartmann, and G Kanschat. deal.II—A general-purpose object-oriented finite element library, 2007. [8](#)
- [20] Andreas Pautz, Siegfried Zangenbuch, Armin Seubert, and Winfried Zwermann. Results on the OECD/NEA C5G7-MOX benchmark obtained with the discrete ordinates code dort. *Progress in Nuclear Energy*, 45(2-4):153–168, January 2004. [31](#), [32](#), [49](#)
- [21] Boyan Ivanov, Erwin Müller, Mohamed Ouisloumen, and Kostadin Ivanov. Embedded lattice transport calculations based on PARAGON-NEM code system for reactor core analysis. *Physor*, 1(814):1–9, 2008. [38](#)

# Appendix A

## Input files for FOREST

In the following the input files for FOREST are presented. First, in File [A.1](#) we can see how the geometry of the problem is introduced in the solver. Secondly, in File [A.2](#), we can observe how the materials used in the problem are defined. Finally, in File [A.3](#), the input containing the different values of the parameters of discretization is presented.

File A.1: input.geometry.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <geometry>
3 <!--
4   This geometry is composed of nested structures. The first definition
5   is for the core, which is composed by assemblies, which are composed
6   by pins.
7 -->
8 <!-- Here is the core description -->
9 <core composed="true">
10 <name>C5G7</name>
11 <dimension> 2 </dimension>
12 <nnodes> 3 3 </nnodes> <!-- xnodes ynodes znodes -->
13 <length>
14   21.42 21.42 21.42; <!-- X- X+ AXIS -->
15   21.42 21.42 21.42; <!-- Y- Y+ AXIS -->
16 </length>
17 <components> <!-- LAT or BOX or PCNW PCNE PCSW PCSE -->
18   2 1 0;
19   1 2 0;
20   0 0 0;
21 </components>
22 <boundary> <!-- ALB or VOID or REFL -->
23   2; <!-- X- AXIS -->
24   0; <!-- X+ AXIS -->
25   0; <!-- Y- AXIS -->
26   2; <!-- Y+ AXIS -->
27 </boundary>
28 </core>
29 <!-- Here is the lattice description -->
30 <lattices>
31 <lattice id="0">
32 <name>moderator assembly</name>
33 <nnodes> 17 17 </nnodes> <!-- xnodes ynodes znodes -->
34 <components> <!-- PIN or BOX -->
35   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
36   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
37   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
38   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
39   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
40   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
41   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
```

```

42      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
43      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
44      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
45      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
46      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
47      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
48      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
49      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
50      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
51      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
52      </components>
53      <type>pin_map</type>
54      <pitch>1.26</pitch>
55      </lattice>
56      <lattice id="1">
57      <name>MOX</name>
58      <nnodes> 17 17 </nnodes> <!-- xnodes ynodes znodes -->
59      <components>
60      2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2;
61      2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2;
62      2 3 3 3 3 6 3 3 6 3 3 6 3 3 6 3 3 2;
63      2 3 3 6 3 4 4 4 4 4 4 4 4 4 3 6 3 3 2;
64      2 3 3 3 4 4 4 4 4 4 4 4 4 4 4 3 3 3 2;
65      2 3 6 4 4 6 4 4 6 4 4 6 4 4 6 4 4 6 3 2;
66      2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2;
67      2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2;
68      2 3 6 4 4 6 4 4 5 4 4 6 4 4 6 4 4 6 3 2;
69      2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2;
70      2 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2;
71      2 3 6 4 4 6 4 4 6 4 4 6 4 4 6 4 4 6 3 2;
72      2 3 3 3 4 4 4 4 4 4 4 4 4 4 4 3 3 3 2;
73      2 3 3 6 3 4 4 4 4 4 4 4 4 4 3 6 3 3 2;
74      2 3 3 3 3 6 3 3 6 3 3 6 3 3 6 3 3 3 2;
75      2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2;
76      2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2;
77      </components>
78      <type>pin_map</type>
79      <water_gap>0. 0. 0. 0.;</water_gap>
80      <pitch>1.26</pitch>
81      </lattice>
82      <lattice id="2">
83      <name>UOX</name>
84      <nnodes> 17 17 </nnodes> <!-- xnodes ynodes znodes -->
85      <components>
86      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
87      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
88      1 1 1 1 1 6 1 1 6 1 1 6 1 1 6 1 1 1 1;
89      1 1 1 6 1 1 1 1 1 1 1 1 1 1 1 6 1 1 1;
90      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
91      1 1 6 1 1 6 1 1 6 1 1 6 1 1 6 1 1 6 1 1;
92      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
93      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
94      1 1 6 1 1 6 1 1 5 1 1 6 1 1 6 1 1 6 1 1;
95      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
96      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
97      1 1 6 1 1 6 1 1 6 1 1 6 1 1 6 1 1 6 1 1;
98      1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
99      1 1 1 6 1 1 1 1 1 1 1 1 1 1 1 6 1 1 1;
100     1 1 1 1 1 6 1 1 6 1 1 6 1 1 6 1 1 1 1;
101     1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
102     1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
103     </components>
104     <type>pin_map</type>
105     <water_gap>0. 0. 0. 0.;</water_gap>
106     <pitch>1.26</pitch>
107     </lattice>
108     </lattices>
109     <!-- Here is the pin description -->
110     <pins>
111     <pin id="0">
112     <name>moderator</name>
113     <type>box</type>
114     <materials>0;</materials>
115     </pin>
117     <pin id="1">
118     <name>UO2</name>
119     <type>pin</type>
120     <materials>0 1;</materials>
121     <fuel_radius>0.54</fuel_radius>
122     </pin>
124     <pin id="2">
125     <name>MOX4.3</name>
126     <type>pin</type>
127     <materials>0 2;</materials>
128     <fuel_radius>0.54</fuel_radius>
129     </pin>
131     <pin id="3">

```

```

132 <name>MOX7.0</name>
133 <type>pin</type>
134 <materials>0 3;</materials>
135 <fuel_radius>0.54</fuel_radius>
136 </pin>
137
138 <pin id="4">
139 <name>MOX8.7</name>
140 <type>pin</type>
141 <materials>0 4;</materials>
142 <fuel_radius>0.54</fuel_radius>
143 </pin>
144
145 <pin id="5">
146 <name>fissionchamber</name>
147 <type>pin</type>
148 <materials>0 5;</materials>
149 <fuel_radius>0.54</fuel_radius>
150 </pin>
151
152 <pin id="6">
153 <name>guidetube</name>
154 <type>pin</type>
155 <materials>0 6;</materials>
156 <fuel_radius>0.54</fuel_radius>
157 </pin>
158
159 </pins>
160
161 </geometry>

```

## File A.2: input.materials.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <materials ngroups="7">
3
4 <!--
5 Here we define the different materials associated to the
6 material id specified when defining the geometry
7 -->
8
9 <mix id="0">
10 <name>water</name>
11 <SigmaT>
12 0.159206 0.41297 0.59031 0.58435 0.718 1.25445 2.65038;
13 </SigmaT>
14 <Chi>
15 0.58791000000 0.41176000000 0.00033906000 0.00000011761 0.00000000000 0.00000000000 0.00000000000;
16 </Chi>
17 <SigmaS>
18 0.044477700000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000;
19 0.113400000000 0.282334000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000;
20 0.000723470000 0.129940000000000 0.345256000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000;
21 0.000003749900 0.000623400000000 0.224570000000000 0.091028400000000 0.000071437000000 0.000000000000000 0.000000000000000;
22 0.000000053184 0.000048002000000 0.016999000000000 0.415510000000000 0.139138000000000 0.002215700000000 0.000000000000000;
23 0.000000000000 0.000007448600000 0.002644300000000 0.063732000000000 0.511820000000000 0.699913000000000 0.132440000000000;
24 0.000000000000 0.000001045500000 0.000050344000000 0.012139000000000 0.061229000000000 0.537320000000000 2.480700000000000;
25 </SigmaS>
26 <NuSigF>
27 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000;
28 </NuSigF>
29 <SigF>
30 0.00000000000 0.00000000000 0.00000000000;
31 </SigF>
32 <Nu>
33 0.00000000000 0.00000000000 0.00000000000;
34 </Nu>
35 <SigmaA>
36 0.00060105 0.000015793 0.00033716 0.0019406 0.0057416 0.015001 0.037239;
37 </SigmaA>
38 </mix>
39
40 <mix id="1">
41 <name>fuelUO2</name>
42 <SigmaT>
43 0.177949 0.329805 0.480388 0.554367 0.311801 0.395168 0.564406;
44 </SigmaT>
45 <Chi>
46 0.58791000000 0.41176000000 0.00033906000 0.00000011761 0.00000000000 0.00000000000 0.00000000000;
47 </Chi>
48 <SigmaS>
49 0.127537000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000;
50 0.042378000000000 0.324456000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000;
51 0.000009437400000 0.001631400000000 0.450940000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000;
52 0.000000005516300 0.000000003142700 0.002679200000000 0.452565000000000 0.000125250000000 0.000000000000000 0.000000000000000;
53 0.000000000000000 0.000000000000000 0.000000000000000 0.005566400000000 0.271401000000000 0.001296800000000 0.000000000000000;
54 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.010255000000000 0.265802000000000 0.008545800000000;

```



```

55 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.00000010021000 0.0168090000000000 0.2730800000000000;
56 </SigmaS>
57 <NuSigF>
58 0.0200599843 0.002027303 0.0157059918 0.0451830102 0.0433420839 0.2020900962 0.5257105352;
59 </NuSigF>
60 <SigF>
61 0.00721206 0.000819301 0.0064532 0.0185648 0.0178084 0.0830348 0.216004;
62 </SigF>
63 <Nu>
64 2.78145 2.47443 2.43383 2.4338 2.4338 2.4338 2.4338;
65 </Nu>
66 <SigmaA>
67 0.0080248 0.0037174 0.026769 0.096236 0.03002 0.11126 0.28278;
68 </SigmaA>
69 </mix>
71 <mix id="2">
72 <name>fuelMOX4.3</name>
73 <SigmaT>
74 0.178731 0.330849 0.483772 0.566922; 0.426227 0.678997 0.682852;
75 </SigmaT>
76 <Chi>
77 0.5879100000 0.4117600000 0.0003390600 0.0000011761 0.0000000000 0.0000000000 0.0000000000;
78 </Chi>
79 <SigmaS>
80 0.1288760000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
81 0.0414130000000000 0.3254520000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
82 0.0000082290000000 0.0016395000000000 0.4531880000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
83 0.000000005040500 0.000000001598200 0.0026142000000000 0.4571730000000000 0.0001604600000000 0.0000000000000000 0.0000000000000000;
84 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0055394000000000 0.2768140000000000 0.0020051000000000 0.0000000000000000;
85 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0093127000000000 0.2529620000000000 0.0084948000000000;
86 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.000000009165600 0.0148500000000000 0.2650070000000000;
87 </SigmaS>
88 <NuSigF>
89 0.0217530045 0.0025351033 0.0162679915 0.0654740997 0.0307240878 0.6666509616 0.7139904304;
90 </NuSigF>
91 <SigF>
92 0.00762704 0.000876898 0.00569835 0.0228872 0.0107635 0.232757 0.248968;
93 </SigF>
94 <Nu>
95 2.85209 2.89099 2.85486 2.86073 2.85447 2.86415 2.8678;
96 </Nu>
97 <SigmaA>
98 0.0084339 0.0037577 0.02797 0.10421 0.13994 0.40918 0.40935;
99 </SigmaA>
100 </mix>
102 <mix id="3">
103 <name>fuelMOX7.0</name>
104 <SigmaT>
105 0.181323 0.334368 0.493785 0.591216 0.474198 0.833601 0.853603;
106 </SigmaT>
107 <Chi>
108 0.5879100000 0.4117600000 0.0003390600 0.0000011761 0.0000000000 0.0000000000 0.0000000000;
109 </Chi>
110 <SigmaS>
111 0.1304570000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
112 0.0417920000000000 0.3284280000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
113 0.0000085105000000 0.0016436000000000 0.4583710000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
114 0.000000005132900 0.000000002201700 0.0025331000000000 0.4637090000000000 0.0001761900000000 0.0000000000000000 0.0000000000000000;
115 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0054766000000000 0.2823130000000000 0.0022760000000000 0.0000000000000000;
116 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0087289000000000 0.2497510000000000 0.0088645000000000;
117 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.000000009001600 0.0131140000000000 0.2595290000000000;
118 </SigmaS>
119 <NuSigF>
120 0.023813952 0.0038586888 0.0241340014 0.09436622 0.0457698761 0.9281814045 1.0432001182;
121 </NuSigF>
122 <SigF>
123 0.00825446 0.00132565 0.00842156 0.032873 0.0159636 0.323794 0.362803;
124 </SigF>
125 <Nu>
126 2.88498 2.91079 2.86574 2.87063 2.86714 2.86658 2.87539;
127 </Nu>
128 <SigmaA>
129 0.0090657 0.0042967 0.032881 0.12203 0.18298 0.56846 0.58521;
130 </SigmaA>
131 </mix>
133 <mix id="4">
134 <name>fuelMOX8.7</name>
135 <SigmaT>
136 0.183045 0.336705 0.500507 0.606174 0.502754 0.921028 0.955231;
137 </SigmaT>
138 <Chi>
139 0.5879100000 0.4117600000 0.0003390600 0.0000011761 0.0000000000 0.0000000000 0.0000000000;
140 </Chi>
141 <SigmaS>
142 0.1315040000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
143 0.0420460000000000 0.3304030000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;
144 0.0000086972000000 0.0016463000000000 0.4617920000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000 0.0000000000000000;

```

```

145 0.000000005193800 0.00000002600600 0.002474900000000 0.468021000000000 0.000185970000000 0.000000000000000 0.000000000000000;
146 0.000000000000000 0.000000000000000 0.000000000000000 0.005433000000000 0.285771000000000 0.002391600000000 0.000000000000000;
147 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.008397300000000 0.247614000000000;
148 0.000000000000000 0.000000000000000 0.000000000000000 0.000000000000000 0.000000008928000 0.012322000000000 0.256093000000000;
149 </SigmaS>
150 <NuSigF>
151 0.0251860041 0.0047395095 0.029478054 0.1122499985 0.0553030128 1.0749988378 1.23929836992;
152 </NuSigF>
153 <SigF>
154 0.00867209 0.00162426 0.0102716 0.0390447 0.0192576 0.374888 0.430599;
155 </SigF>
156 <Nu>
157 2.90426 2.91795 2.86986 2.87491 2.87175 2.86752 2.87808;
158 </Nu>
159 <SigmaA>
160 0.0094862 0.0046556 0.03624 0.13272 0.2084 0.6587 0.69017;
161 </SigmaA>
162 </mix>
164
164 <mix id="5">
165 <name>fissionchamber</name>
166 <SigmaT>
167 0.126032 0.29316 0.28425 0.28102 0.33446 0.56564 1.17214;
168 </SigmaT>
169 <Chi>
170 0.58791000000 0.41176000000 0.00033906000 0.00000011761 0.00000000000 0.00000000000 0.00000000000;
171 </Chi>
172 <SigmaS>
173 0.066165900000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000;
174 0.059070000000000 0.240377000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000;
175 0.000283340000000 0.052435000000000 0.183425000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000;
176 0.000001462200000 0.000249900000000 0.092288000000000 0.079076900000000 0.000037340000000 0.0000000000000 0.0000000000000;
177 0.000000020642000 0.000019239000000 0.006936500000000 0.169990000000000 0.099757000000000 0.000917420000000 0.0000000000000;
178 0.000000000000000 0.000002987500000 0.001079000000000 0.025860000000000 0.206790000000000 0.316774000000000 0.049793000000000;
179 0.000000000000000 0.000000421400000 0.000205430000000 0.004925600000000 0.024478000000000 0.238760000000000 1.099100000000000;
180 </SigmaS>
181 <NuSigF>
182 0.0000000132340 0.0000000143450 0.0000011285993 0.0000127629932 0.0000003538502 0.0000017400989 0.0000050633019;
183 </NuSigF>
184 <SigF>
185 0.0000000047900 0.0000000058256 0.0000004637190 0.0000052440600 0.0000001453900 0.0000007149720 0.0000020804100;
186 </SigF>
187 <Nu>
188 2.76283 2.46239 2.4338 2.4338 2.4338 2.4338 2.4338;
189 </Nu>
190 <SigmaA>
191 0.00051132 0.000075813 0.00031643 0.0011675 0.0033977 0.0091886 0.023244;
192 </SigmaA>
193 </mix>
195
195 <mix id="6">
196 <name>Guidetube</name>
197 <SigmaT>
198 0.126032 0.29316 0.28424 0.28096 0.334440 0.56564 1.17215;
199 </SigmaT>
200 <Chi>
201 0.58791000000 0.41176000000 0.00033906000 0.00000011761 0.00000000000 0.00000000000 0.00000000000;
202 </Chi>
203 <SigmaS>
204 0.066165900000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000;
205 0.059070000000000 0.240377000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000;
206 0.000283340000000 0.052435000000000 0.183297000000000 0.0000000000000 0.0000000000000 0.0000000000000 0.0000000000000;
207 0.000001462200000 0.000249900000000 0.092397000000000 0.078851100000000 0.000037333000000 0.0000000000000 0.0000000000000;
208 0.000000020642000 0.000019239000000 0.006944600000000 0.170140000000000 0.099737200000000 0.000917260000000 0.0000000000000;
209 0.000000000000000 0.000002987500000 0.001080300000000 0.025881000000000 0.206790000000000 0.316765000000000 0.049792000000000;
210 0.000000000000000 0.000000421400000 0.000205670000000 0.004929700000000 0.024478000000000 0.238770000000000 1.099120000000000;
211 </SigmaS>
212 <NuSigF>
213 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000;
214 </NuSigF>
215 <SigF>
216 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000;
217 </SigF>
218 <Nu>
219 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000 0.00000000000;
220 </Nu>
221 <SigmaA>
222 0.00051132 0.000075801 0.00031572 0.0011582 0.0033975 0.0091878 0.023242;
223 </SigmaA>
224 </mix>
225 </materials>

```

### File A.3: input.settings.xml

```

1 <?xml version='1.0' encoding='utf-8'?>
2 <settings>
3 <dim>2</dim>

```

```
4 <input_files>
5   <geom>input.geom.xml</geom>
6   <mat>input.mat.xml</mat>
7   <out>output_c5g7_2_20_2.xml</out>
8 </input_files>
9 <geometry>
10  <n_ref>2</n_ref>
11 </geometry>
12 <problem type="transport">
13   <approximation type="sn">
14     <sn>20</sn>
15   </approximation>
16 </problem>
17 <algebra>
18   <eig_solver type="PI">
19     <tol>1.e-6</tol>
20     <max_it>500</max_it>
21   </eig_solver>
22   <inner_solver type="Krylov">
23     <tol>1.e-7</tol>
24     <max_it>400</max_it>
25   </inner_solver>
26 </algebra>
27 <fe_settings>
28   <degree>2</degree>
29 </fe_settings>
30 </settings>
```