A Parametric Study on Core Performance of Sodium Fast Reactors Using SERPENT Code

RUBÉN GARCÍA MORENO

Master of Science Thesis
Division of Nuclear Safety
Royal Institute of Technology
Stockholm, Sweden

Abstract

In this thesis, a parametric study was performed for four core designs of Sodium Fast Reactors (SFRs) in two different sizes: the large core of 3600 MWth with oxide or carbide fuel, and the medium core of 1000 MWth with oxide or metallic fuel. The Monte Carlo code SERPENT is chosen in the simulations, and the following global safety parameters are calculated at the beginning and end of cycle:

- core multiplication factor;
- sodium void worth;
- Doppler constant;
- effective delayed neutron fraction;
- control rod worth;
- average nuclide concentrations for end of equilibrium cycle;
- radial power distribution.

The calculated results show that the four cores remain slightly supercritical through the whole equilibrium cycle, with the multiplication factors being close to unity. The sodium void worth increases as the burnup is taking place. The Doppler constants are quite different for the four cores, ranging from -1.2 (EOC of metallic core and BOC of carbide core) to -2.1 (BOC of oxide core). The effective delayed neutron fractions are similar for all the four cores. The control rod worth for the medium cores is much higher than that of the large cores, due the greater ratio of control assemblies to driver assemblies in in the medium cores. The data of average nuclide concentrations at EOC indicate that the transuranics conversion ratios in the large cores are greater than those in the medium cores. Over time the power tends to increase in the inner core and keeps constant or decreasing in the outer core.

The library of cross sections appears to have a significant impact on the simulated results. This implies the importance of having the right data for the cross sections.

Acknowledgements

I would like to thank Dr. Weimin Ma for being my supervisor and for his support, and thank Dr. Tomasz Kozlowski for taking me to do my Master thesis at the Division of Nuclear Power Safety, and remaining in constant contact and helpful despite having left his position at KTH after I started my thesis project. I also want to express my gratitude to Milan Tesinski and Ivan Gajev for their constructive advice and technical help. I give special thanks to the people in the Division of Nuclear Power Safety as well as the people in the Division of Reactor Physics for being so nice. At last, I would like to thank KTH and Helena Legnell for making possible that I learned Swedish.

Table of contents

Abbreviations and definitions

SFR	Sodium-cooled Fast Reactor
GIF	Generation IV International Forum
BOC	Beginning of cycle. The state of the reactor in the moment of starting the burnup cycle
EOC	End of cycle. The state of the reactor after it has been put under burnup for the complete length of a normal cycle
LWR	Light Water Reactor
ABR	Advanced Burner Reactor
ODS	Oxide Strengthened Steel
K-eff	Core multiplication factor
Sodium void worth	Variation in reactivity when comparing the normal core with a core without sodium in its active core (also known as voided core).
Active core	Zone in a fuel assembly that contains the fissile material.
Doppler constant	Variation in reactivity when comparing the normal core with a core with the fuel temperature doubled
Effective delayed neutron fraction	Fraction of neutrons that cause fission that are delayed
Control rod worth	Variation of reactivity between the control rods being withdrawn from the core and totally inserted.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Fast Reactors

The Generation IV International Forum (GIF) is an organization with participation of different countries. Its goal is to research and develop new types of reactors called Generation IV, which are a technological step ahead of their predecessors, the Generation III. Most of the reactors being developed are not expected to work commercially before 2030.

One of the reactors being developed is the Sodium-Cooled Fast Reactor (SFR). The main characteristic of this reactor is that the fissions take place in a fast neutron spectrum, meaning that it doesn't have or need moderator. As a consequence, it cannot have water as coolant, because of its moderating properties. The solution given this case is to use liquid sodium instead of water.

The fast neutron spectrum has lower cross sections compared to the thermal spectrum. This implies two things: first of all, the reactor needs a bigger critical mass, and this translates into more fuel enrichment, which is a problem regarding proliferation; the second consequence is that there are going to be more excess neutrons in the core, and you can use those excess neutrons to "breed" new fuel.

The main idea of a fast reactor is not only to produce energy, but also treat the fuel at the same time. Besides energy production, a fast reactor can work in two fundamental modes: breeding and burning.

In the breeding mode, there is usually critical fuel at the center of the core, and fertile blankets around it. These fertile blankets are usually made of natural or depleted Uranium, which is mainly U-238, and which through some absorptions and decays transforms into the fissile Pu-239.

In the burning mode, the fertile blankets are substituted by reflectors and the fuel at the center is treated depleted fuel. The objective of burning the fuel is mainly to get rid of Pu-239 and Np-237, two isotopes that are very inconvenient in the nuclear waste. Without them, the half life of this waste can be reduced dramatically, from thousands to hundreds of years.

Table 1: Transuranics conversion ratio and modes

Conversion Ratio	Mode	Comments
<1	Burner	Consumes fissile elements; useful for reducing nuclear waste
1	Converter	Consumption equals production
>1	Breeder	Produces new fissile transuranics

The rate at which the fuel transmutes in a fast reactor core can be characterised by a parameter called the Transuranics Conversion Ratio. This parameter tells the ratio of transuranics being produced divided by the transuranics being transformed into fission products. The mode at which a reactor works can be determined by this parameter. This can be seen in table 1.

Fast reactors allow the existence of a closed fuel cycle. Used fuel in LWRs can be separated chemically and treated in a way to be usable by Fast Reactors. The fission products are thrown away, and the rest enters this fuel cycle. Once it enters, in theory, it can be used over and over until it transforms totally into fission products. In figure 1, the fast reactor is substituted by an ABR. Note that not all ABRs have to be fast reactors, but the idea behind its cycle is the same.

Closed cycles, and wasted fuel re-usage in general, have the problem of proliferation. In some countries, like the United States, it is forbidden. In a chemical separation of a burned fuel, there is a possibility that the Plutonium is used for weapons purposes by terrorists. The Plutonium in a burned fuel has some amount of the isotope Pu-240, more if the burnup is long. This isotope makes it very difficult to build nuclear weapons, although not impossible. It is important, therefore, that the burnups are as long as possible before they go into treatment.

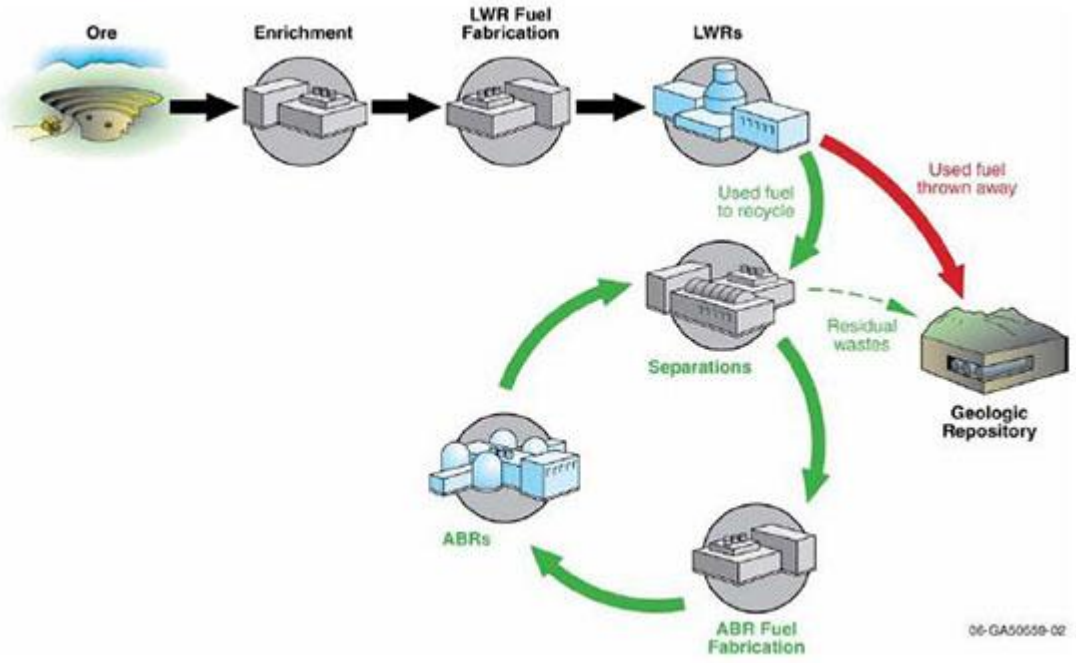

Figure 1: Fuel cycle for Advanced Burner Reactors

1.1.1 Sodium Fast Reactors

Water cannot be used in a sodium fast reactor due to its moderation capabilities. Therefore, different coolant materials have been proposed and sometimes used. Liquid metal is usually the best choice due to its high heat conductivity, but metals usually have a high melting point. For that reason, sodium is one of the best choices because it has a relatively low melting point. Also, the boiling point is higher than the operating temperature of the reactor. For this reason, SFRs don't need to be pressurized; they can work at almost ambient pressure.

The main drawback of sodium is that it reacts spontaneously with water and air, so it has to be designed in a way that this doesn't happen. One safety design is to implement a second sodium

loop between the main core loop and the turbine water loop, so that, in case of contact between sodium and water, it does not damage the main core. It also has to be totally isolated from air.

Another safety concern is that, unlike water, it doesn't moderate, so when it boils, the criticality of the reactor doesn't decrease, but in fact, increases. In other words, the void coefficient of the coolant is positive. This has to be countered by other negative temperature feedbacks.

1.2 Monte Carlo method

A Monte Carlo method is an algorithm that uses random or pseudo-random numbers to solve a deterministic problem or equation. This algorithm generally consists of generating random numbers within a domain, and then calculating a result with a deterministic equation. These types of methods are a good way to make calculations with a big number of coupled degrees of freedom. The reason is that it simplifies the definition of a problem and allows a computer to do all the work, and also because the uncertainty is lower for the same computation time compared to other methods.

The term "simulation" is often used when talking about calculations with Monte Carlo algorithms, but it is not technically correct. A distinction between the term simulation and Monte Carlo has to be made. A simulation is just an imitation of reality that is done in order to see how the experiment behaves. When something is calculated and random numbers are used, it is then a Monte Carlo method. Some authors use the term "Monte Carlo simulation" due to the similarities between them, but in order to differentiate from a plain simulation.

Monte Carlo codes are very suitable for neutron transport calculations, due to the fact that neutrons don't interact with each other. This allows the calculation to be based on one neutron at a time. These types of codes are usually based on cycles, and every cycle consists of a fixed number of neutrons that run one by one.

Each neutron starts in a particular position, and a random number is sampled to determine the direction of movement. Another random number will be used to determine the path length. When the path length is over, the neutron will interact with a nucleus, and more random numbers will be sampled to determine the interaction that it will undergo. This will continue until the interaction makes the neutron disappear. Once that happens, another neutron will appear and start the same process. Every full path of a neutron from creation to disappearance is called a neutron history.

A cycle is a fixed number of neutron histories. Every time a cycle ends, another one begins with the neutrons placed in the positions where the previous cycle created them. Different data is scored between cycles and this data is used at the end of the calculation to get some results.

1.3 Objectives of this thesis

The main objective of this thesis is to obtain some safety-related parameters for four different cores of sodium fast reactors (SFRs). These cores and parameters are defined by a group of experts (Blanchet et al. 2011) working on the SFRs in the Generation IV International Forum (GIF). As a benchmark exercise, calculations with different codes will be performed by different participants in the world.

The SERPENT code is chosen in the present study to characterize the global parameters and feedback coefficients (e.g., K-eff, power distributions, control rob worth, void and Doppler effects) of neutronics.

-

Chapter 2

Core definitions

Calculations have been made for 4 different cores. Two of them are large and the other two are smaller. The information provided is basically the dimensions of each assembly type, the composition of the materials at each zone and the average temperatures of each medium.

The cores have the shape of a hexagonal prism, and so do the subassemblies inside of them. There are four types of subassemblies: fuel, control, absorber and reflector. Fuel subassemblies consist of an outer cladding made of steel and a lattice of round tubes that contain fuel and other materials. The other subassemblies are similar, but contain absorber materials for the absorber and the control subassemblies, and reflective steel for the reflector subassembly. The difference between absorber and control is that the control subassembly has a mobile part that can be inserted or withdrawn from the core.

2.1 Large size cores

The two large cores are very similar in geometry and compositions, but have slight differences that make them two completely different cores. Both cores have a thermal power of 3600 MW. The burnup cycle is 410 days for the oxide core and 500 days for the carbide core.

2.1.1 Carbide core

The carbide core consists of two types of fuel subassemblies and an outer wall of reflectors. The inner core subassemblies cover the center and most of the core surface, and the outer core has only 3 rows. The difference between the inner core and outer core subassemblies is the fuel composition. There are two types of control subassemblies: primary and secondary. The difference is not in their composition, but in their geometry.

The fuel subassembly consists of a hexagonal tube that contains a lattice of round tubes. The hexagonal tube is made of EM10-like steel, and the inner tubes are made of Oxide Strengthened Steel (ODS). The inner tubes contain three types of materials: EM10 steel, helium and (U,Pu)C fuel. The dimensions are presented in figure 3.

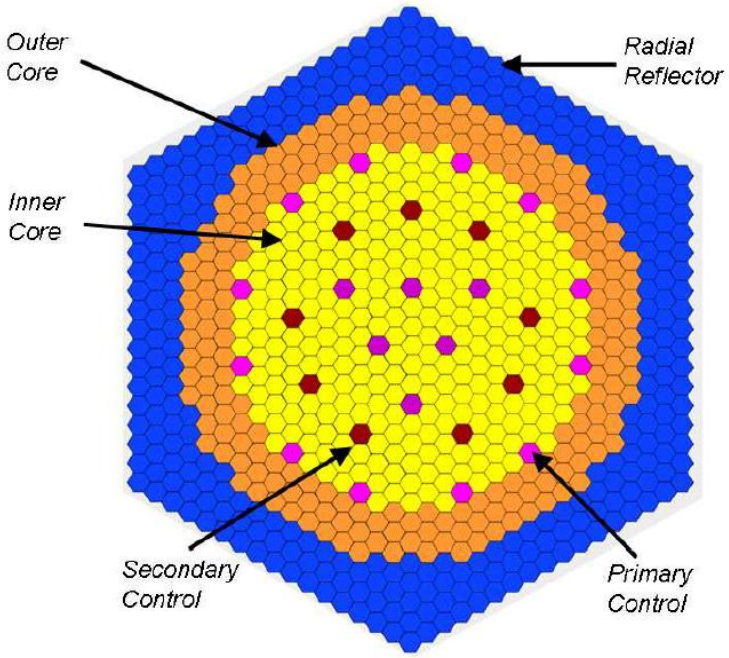

Figure 2: Large carbide core

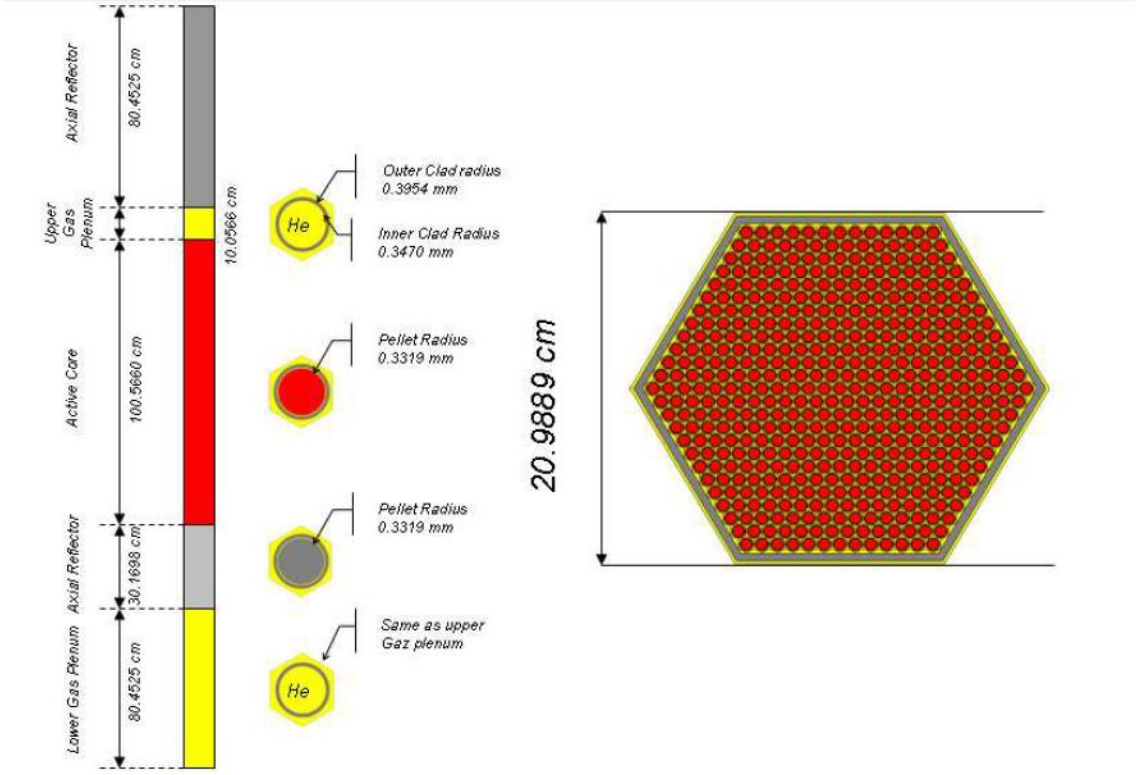

Figure 3: Large carbide core fuel assembly

2.1.2 Oxide core

The oxide core has a very similar configuration when compared to the carbide core. There are the same types of subassemblies, but these are configured in a different way. The outer core occupies 1 more row, and the position of the control subassemblies is different.

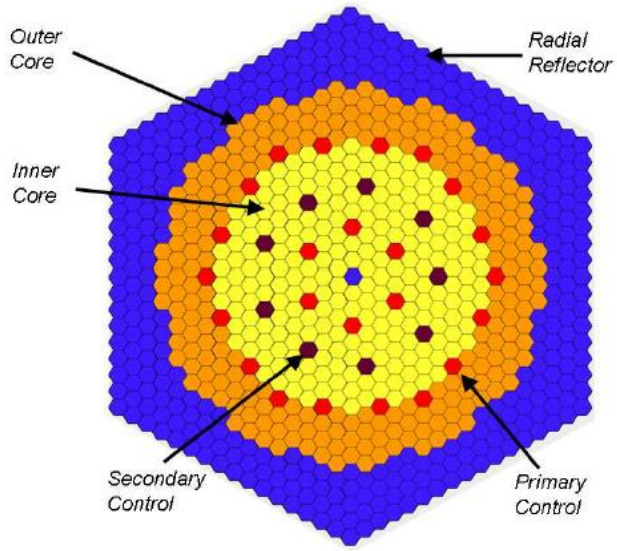

Figure 4: Large oxide core

The fuel subassembly has the same configuration than the carbide core, but the dimensions are different and there are fewer pins in the tube lattice. The fuel is made of $(U,Pu)O_2$.

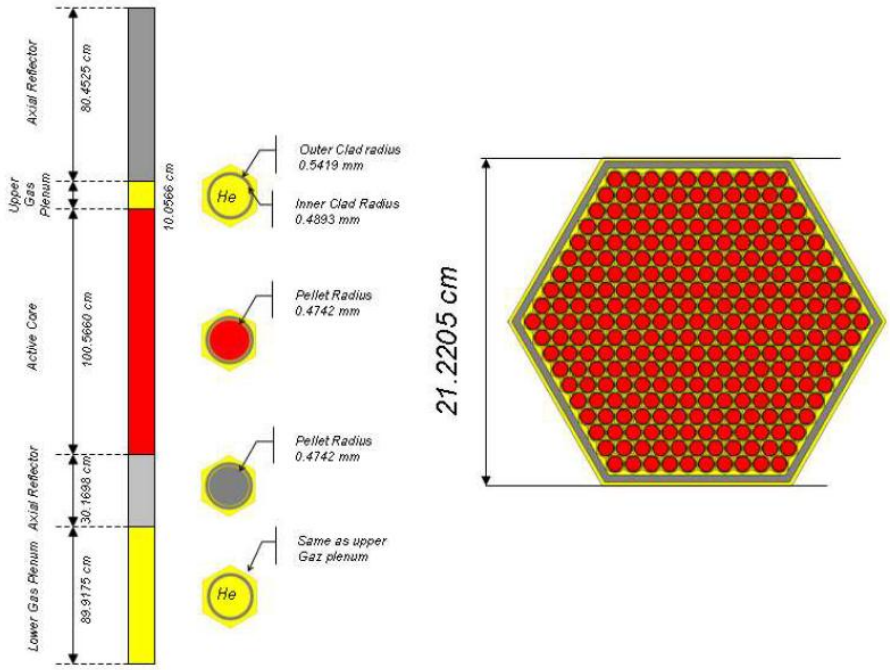

Figure 5: Large oxide core fuel assembly

2.1.3 Control rod and radial reflector design

There are two types of control subassemblies: primary and secondary. The primary has a hexagonal shaped inner cladding that contains the lattice, while in the secondary it is round-shaped. Both cores use the same type of subassemblies, but each one has slight differences. The absorber height is the same for both systems, and is 100.56 cm, the same height than the active core. The rest is an empty duct filled with sodium. In the normal core operation, the absorbers remain at the top of the active core zone, and when they are inserted, they occupy the same space than the active core in their respective duct.

In figure 6, the dimensions for the primary control are presented. The numbers at the left of the slash belong to the oxide core, and at the right to the carbide core.

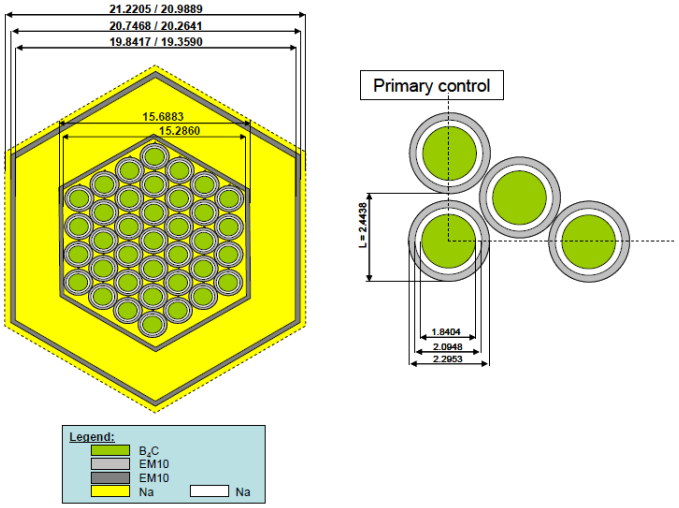

Figure 6: Primary control subassembly

The same applies for the secondary control. The dimensions can be seen in figure 7.

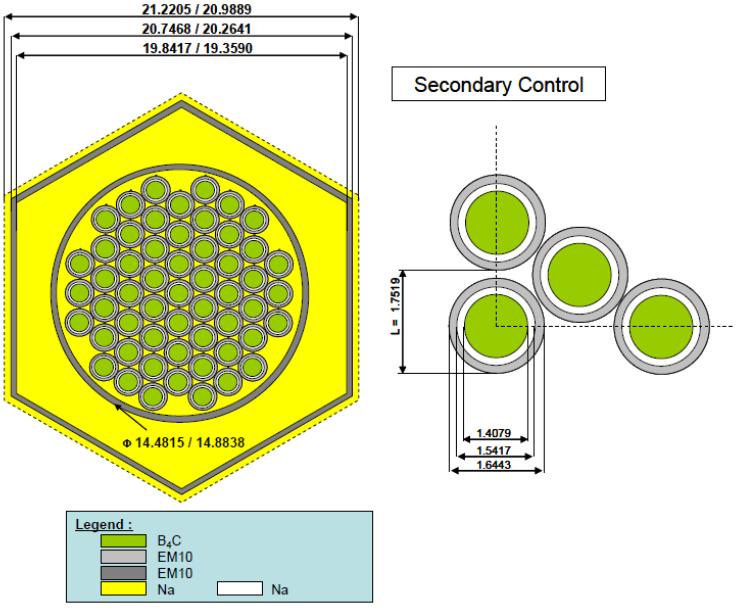

Figure 7: Secondary control subassembly

For the radial reflector, a homogeneous medium made of EM10 is supposed. The volume fractions are 26% for sodium and 74% for EM10.

2.1.4 Material description

2.1.4.1 Fuel materials

The fuel composition is a mixture of depleted Uranium, Plutonium, Neptunium, Americium and Curium, plus some Molybdenum to account for the fission products. The oxide core has all that mixed with Oxygen, and the carbide core is mixed with Carbon.

2.1.4.2 Structure, coolant and absorber materials

Both cores use the same material compositions for everything that is not fuel. The ducts are made of EM10 steel, which is mainly steel with some Chrome, Silicium, Titanium and other materials. The cladding is made of ODS steel, that is made of steel, Chrome, Phosphor and other materials. The coolant is made of Sodium. The control rods are made of a mixture between Carbon and Boron. The primary system has natural Boron, and the secondary is enriched.

2.2 Medium size cores

The two medium size cores are design concepts for ABRs (Advanced Burner Reactors). The idea behind those designs is to allow the interchange of fuel drivers between those two. For that reason, most materials and dimensions are the same for both cores. Their power is 1000 MWth and the burnup cycle is 328.5 days.

2.2.1 Metallic core

The metallic core is composed of two types of fuel subassemblies: inner and outer. The inner subassemblies occupy most of the center area of the core, and the outer subassemblies occupy 3 rows that surround the inner core. Surrounding the outer core there are two rows of reflector, and around it there is one row of shield. There are two types of control rods: primary and secondary.

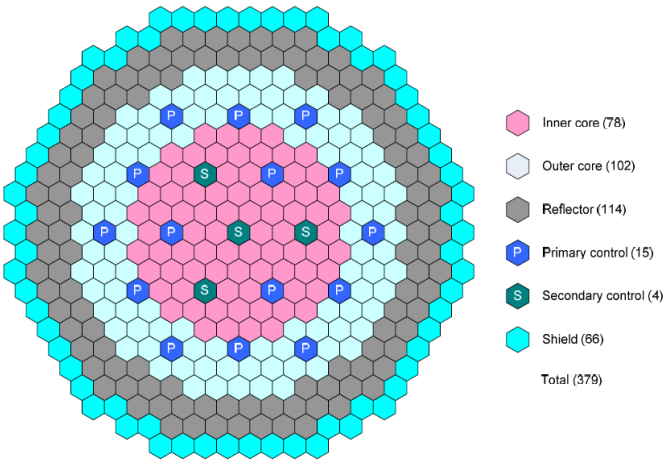

Figure 8: Metallic core

The dimensions of the different types of assemblies can be seen in figures 9 – 12.

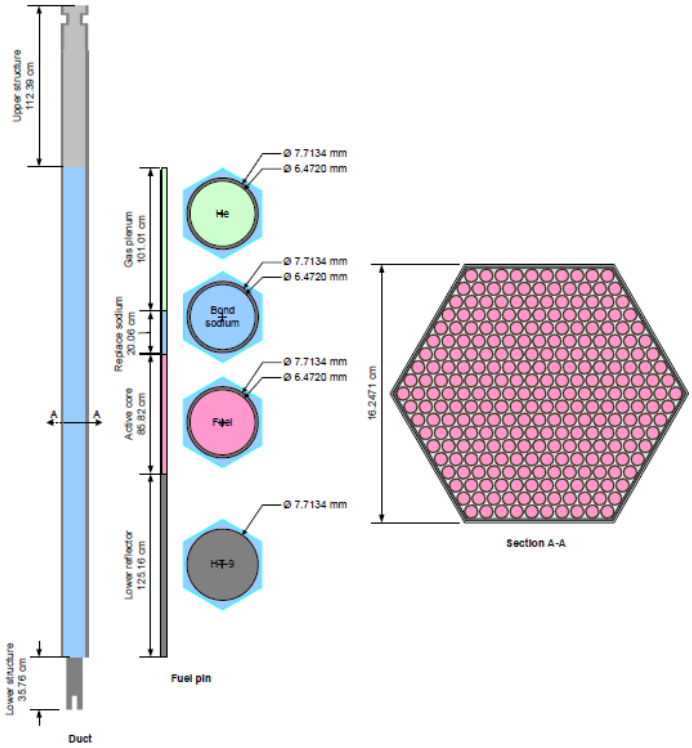

Figure 9: Metallic core fuel subassembly

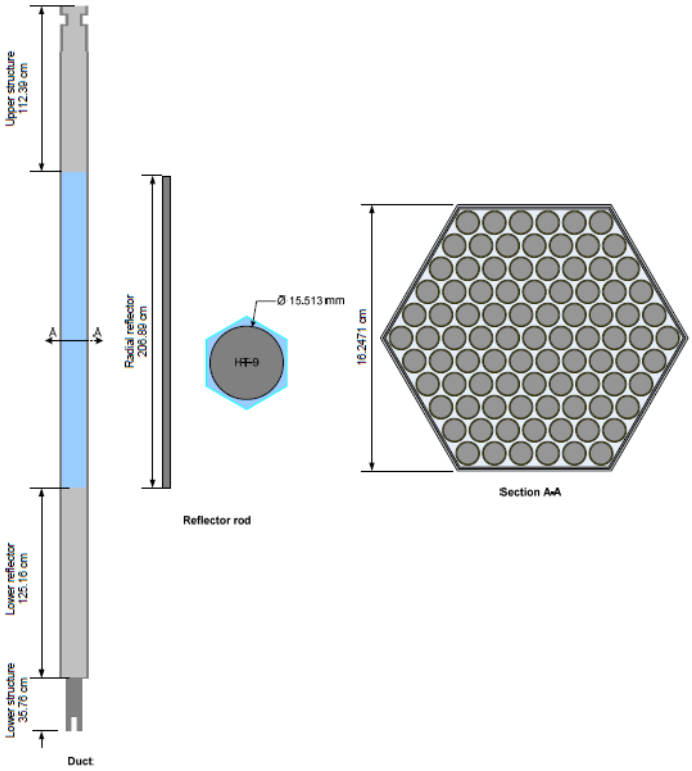

Figure 10: Metallic core reflector subassembly

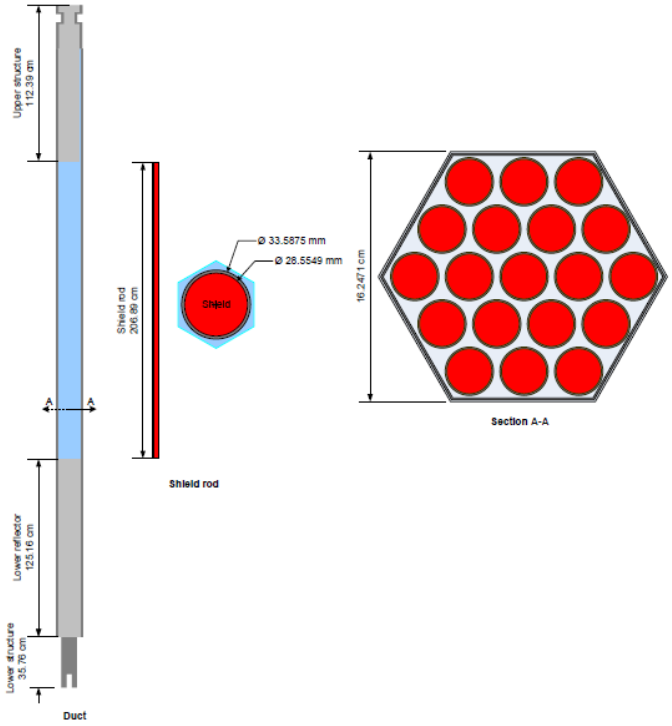

Figure 11: Metallic core shield subassembly

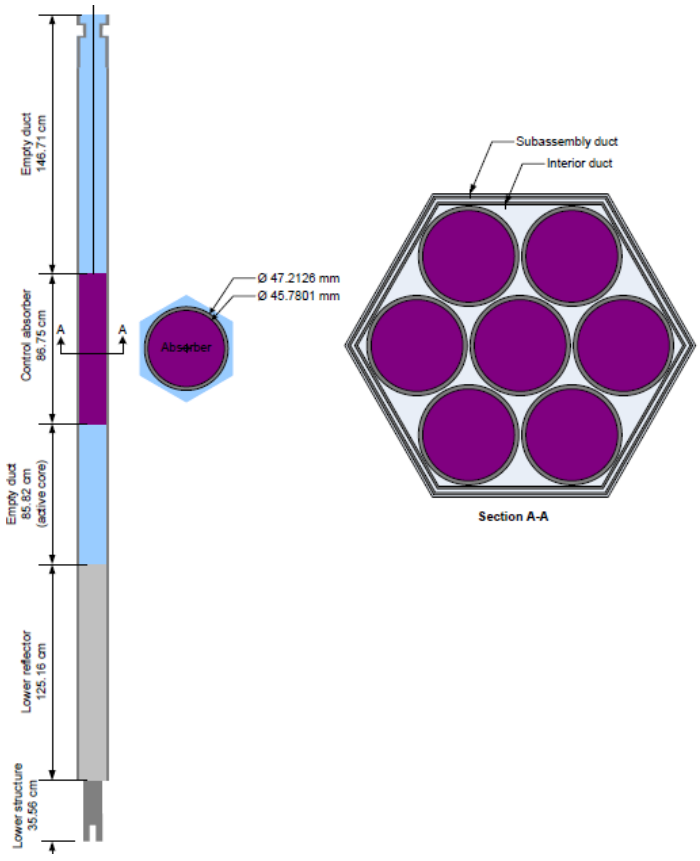

Figure 12: Metallic core control subassembly

2.2.1.1 Material description for the metallic core

There is a specific material used for the lower structure of the subassemblies, which is a mixture of SS-316 and sodium, the rest of the structural materials are HT-9. Both steels contain Chrome, Nickel and other metals. The coolant is Sodium. The control rods have a mixture of Carbon and enriched Boron, and for the radial shields it's the same but with natural Boron.

For the fuel, the composition is very similar to that of the large cores, but the main component is Zirconium.

2.2.2 Oxide core

As was said previously, both cores use the same materials except for the fuel. The only difference in this core is that, instead of two fuel zones, there are three. The inner core is smaller, letting space for three rows of the middle core, and one row for the outer core. The rest of assemblies have the same position as with the metallic core.

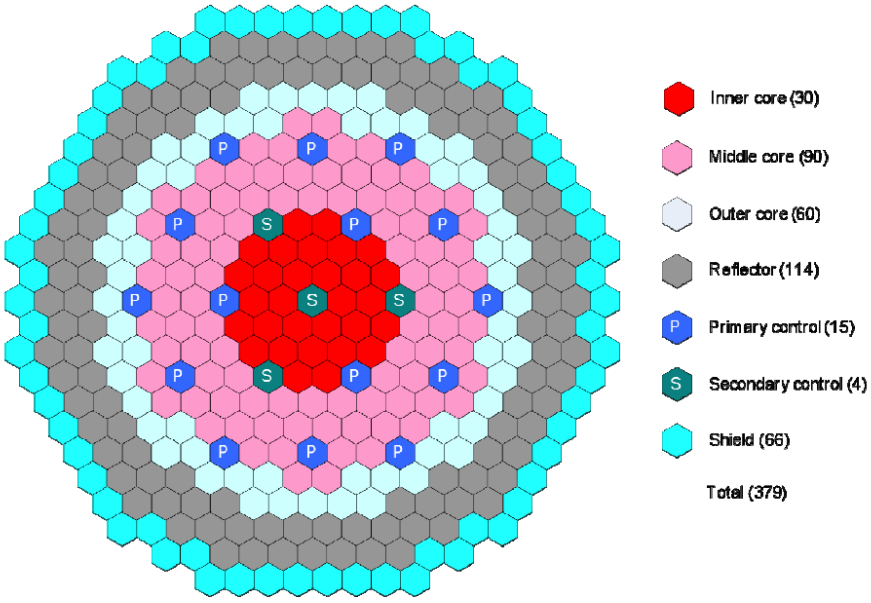

Figure 13: Medium oxide core

Figures 14 - 17 illustrate the different types of subassemblies.

2.2.2.1 Material compositions

The structural materials, the coolant and the absorbers have the exact same material compositions as the metallic core. For the fuel, the composition is very similar, substituting Zirconium for Oxygen.

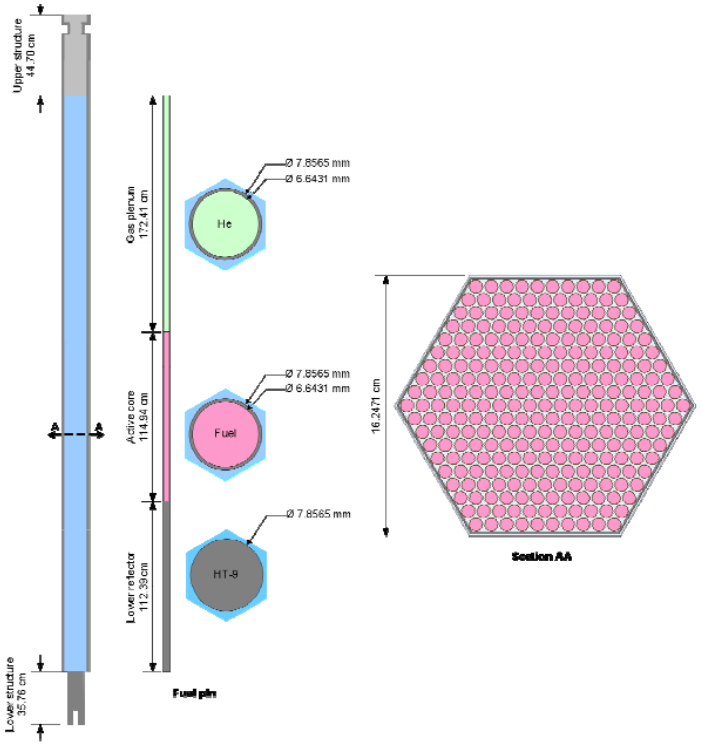

Figure 14: Medium oxide core fuel subassembly

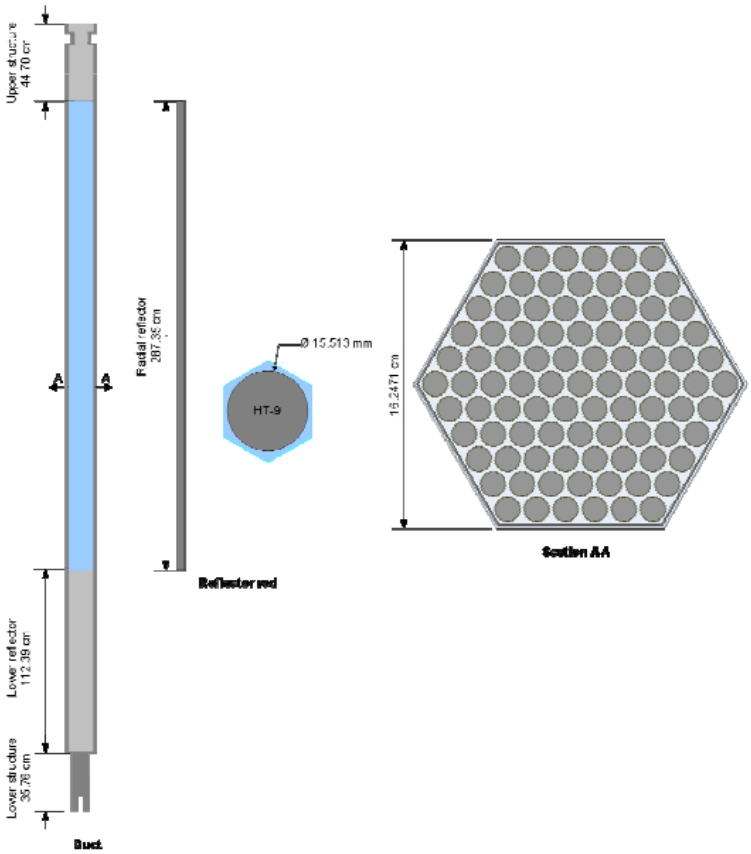

Figure 15: Medium oxide core reflector subassembly

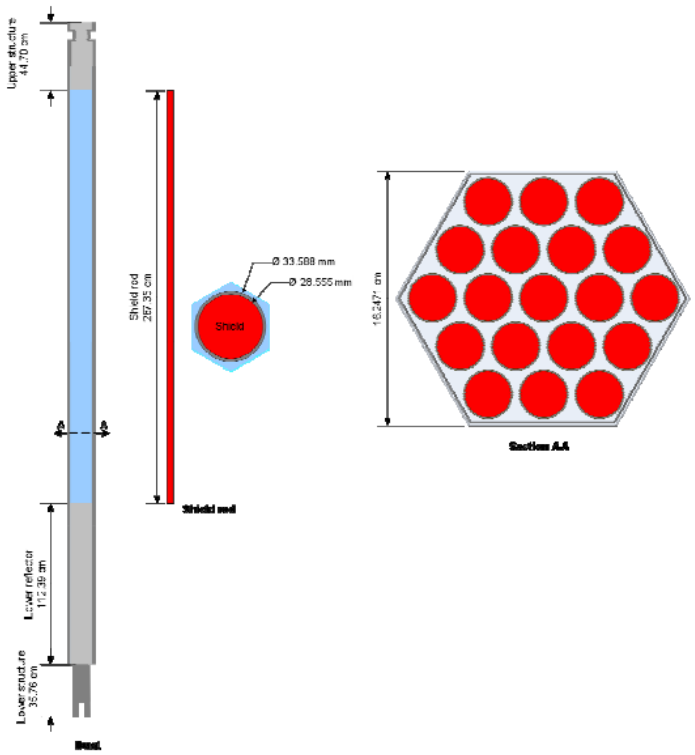

Figure 16: Medium oxide core shield subassembly

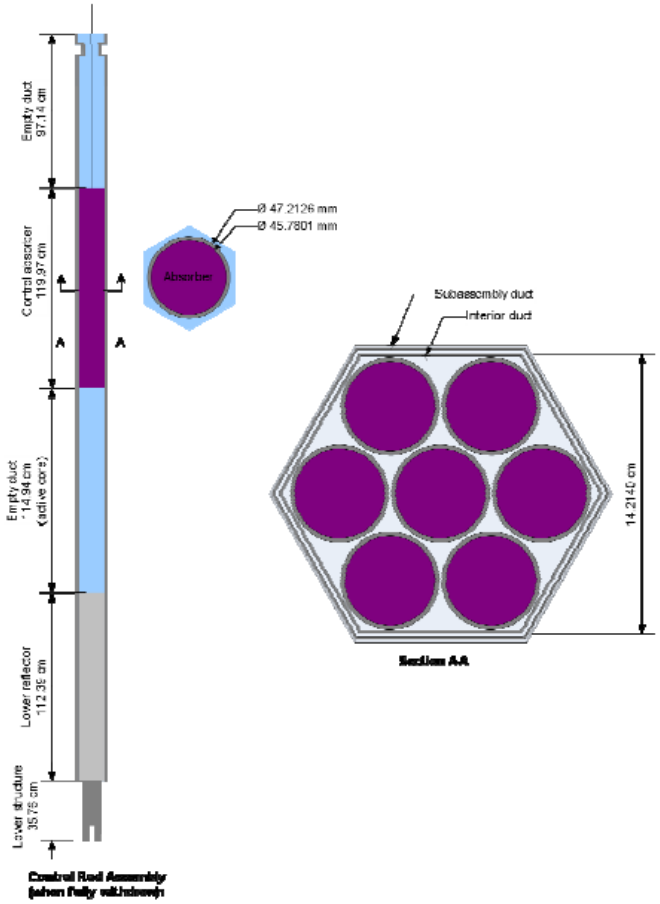

Figure 17: Medium oxide core control subassembly

Chapter 3

Serpent

Serpent is a Monte Carlo code used to calculate different parameters in neutron transport problems. It was developed and is supported by Jaakko Leppänen, doctor of science from the Helsinki University of Technology (this program was, in fact, part of his PhD dissertation).

Serpent is very similar to the code MCNP. It also specialises in constant group generation. Serpent has been through a lot of changes since its release, and now is capable of things that it wasn't at the beginning, i.e. burnup calculations.

The Serpent program runs in Linux operative systems. It has the capability to use parallel calculations, which is running the same input in different computers or different cores in the same computer. This is possible due to the linearity of the algorithm used in monte carlo calculations.

Serpent doesn't have a user visual interface. Everything runs from the command line. To run a simulation, an input file has to be made. Then this input is run through a command in the terminal, and at the end of the calculation some output files appear. Apart from this input files, Serpent also needs library files. Those library files usually come with the Serpent distribution, but can also be found on the internet.

3.1 Input file

Most of the work when doing a calculation is to build an input file with all the information about the core and the characteristics of the simulation. A lot of things have to be included in this input file. The most important and time-consuming things are the geometry and the materials information. There has to be also information about the number of neutrons and the number of cycles. Other information is optional, like the output files wanted or some calculation options for optimising the simulation.

Figure 18 shows the beginning of one of the input files for the Medium Size Metallic Core. Everything that is behind a % sign is not read by Serpent, it is just there as information for a human reader.

The inputs work with "cards". A "card" is a command that tells Serpent what the following information is going to be. In the example it is written for example "set title" followed by a string of characters. This tells Serpent that we want the title of the simulation to be that string (the title will be shown when simulating). Other cards that can be seen are "surf", that defines a surface for the geometry, and "cell", that defines a cell. The meaning of those will be discussed later.

```
% Medium size metallic core withdrawed rods%

set title "Metallic medium core burnup"

%Geometry

surf 1000 pz 0.0  %planes
surf 1001 pz 35.76
surf 1002 pz 160.92
surf 1003 pz 246.74
surf 1004 pz 266.8
surf 1005 pz 333.49
surf 1006 pz 367.81
surf 1007 pz 480.2

surf 1010 pz 178.08 %planes dividing fuel
surf 1011 pz 195.25
surf 1012 pz 212.41
surf 1013 pz 229.58

surf 1014 hexxc 0.0 0.0 0.9 %surface for pin exterior
surf 1015 cyl 0.0 0.0 0.38567 %exterior pin cladding
surf 1016 cyl 0.0 0.0 0.3236 %interior pin cladding

surf 1100 cyl 0.0 0.0 5.2567 %cylinder for lower structure

surf 1200 hexxc 0.0 0.0 6.7104 %hexagonal cylinders for claddings
surf 1201 hexxc 0.0 0.0 7.107
surf 1202 hexxc 0.0 0.0 7.5095 %Outer cladding, inner surface
surf 1203 hexxc 0.0 0.0 7.9061 %Outer cladding, outer surface
surf 1210 hexyc 0.0 0.0 180 %hexagonal cylinder for main core

%Inner core assembly (universe 1)

cell 1 1 fill 12 1001 -1006 -1202 %Lattice
cell 2 1 HT 1001 -1007 1202 -1203 %Cladding
cell 3 1 fill 100 1006 -1007 -1202 %Upper structure
```

Figure 18: Serpent input example

3.2 Building the geometry

Serpent is a universe-based code. A universe is defined as a part of the geometry that includes other universes inside of it. To build a repetitive geometry, a small universe will be defined. Then, inside the "main" universe, this small one can be included in a repetitive way. Like said before, universes are hierarchic, which means that there is a "parent" universe, and inside of it there are other universes, and at the same time, inside of those there are others, etc. Different options can be used to define universes.

3.2.1 Lattices

A lattice is a grid of universes. A lattice is defined by a numerical map. Figure 19 illustrates the geometry that can be built with the lattice card:

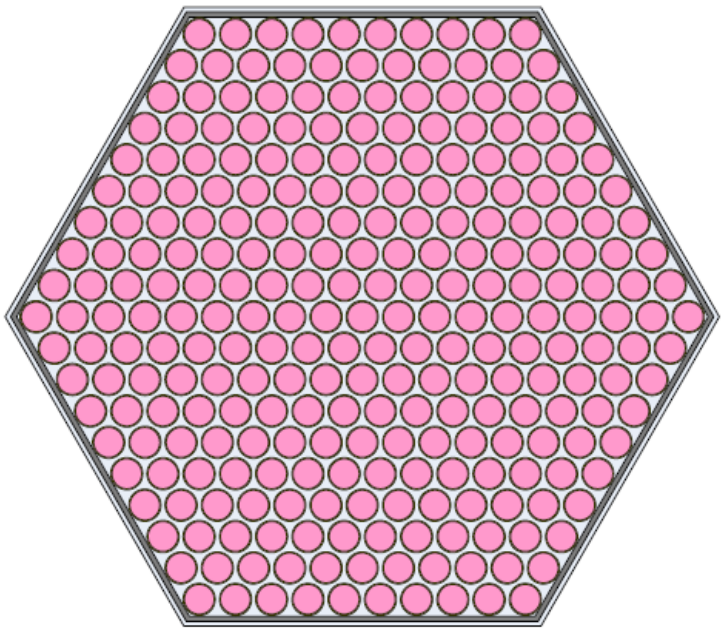

Figure 19: Lattice in the assembly

It is a hexagonal lattice with 10 pin at each side, 271 pins in total. The way a lattice like this is built in the input is in a way like figure 20.

```
lat 12 3 0.0 0.0 23 23 0.8966 %hexagonal y-lattice for inner core assembly

7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
 7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
  7  7  7  7  7  7  7  7  7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 7  7
   7  7  7  7  7  7  7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 7  7
    7  7  7  7  7  7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 7  7
     7  7  7  7  7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
      7  7  7  7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
       7  7  7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
        7  7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
         7  7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
          7  7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
           7  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
            7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7
             7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7  7
              7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7  7  7
               7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7  7  7  7
                7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7  7  7  7  7
                 7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7  7  7  7  7  7
                  7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 20 7  7  7  7  7  7  7  7
                   7  7  20 20 20 20 20 20 20 20 20 20 20 20 20 7  7  7  7  7  7  7  7  7
                    7  7  20 20 20 20 20 20 20 20 20 20 20 20 7  7  7  7  7  7  7  7  7  7
                     7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
                      7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
```

Figure 20: Lattice example in Serpent

"Lat" is the card for lattice building. The number 12 is the universe number that is given to that lattice. The following numbers are the type, position and pin to pin distance. Below that, a 23x23 number square can be seen. This square is displaced with one space character at every line so that the real shape of the hexagon can be seen, although this is not necessary. Universe 20 has

been defined previously as a fuel pin, and universe 7 is a universe filled with sodium. Note that the outer wall of cladding is not present here. To build the cladding, a cell will be needed.

3.2.2 Materials

To set the materials, the card "mat" is used. It has to be followed by a name, then information about the isotopic composition. This composition can be given in different units. In this benchmark, the units used are atoms/barn-cm, which is the atomic density. An example of a material card is given in figure 21.

```
mat Fuel 0 tmp 807.15
92234.06c 1.1369E-06
92235.06c 3.0421E-05
92236.06c 2.4896E-06
92238.06c 1.9613E-02
93237.06c 4.6686E-05
94236.06c 4.9700E-10
94238.06c 1.1695E-04
94239.06c 2.2076E-03
94240.06c 1.3244E-03
94241.06c 1.9375E-04
94242.06c 2.9277E-04
95241.06c 1.0791E-04
95342.06c 9.2989E-06
95243.06c 1.0017E-04
96242.06c 5.6250E-06
96243.06c 5.4321E-07
96244.06c 6.7240E-05
96245.06c 1.7397E-05
96246.06c 9.2285E-06
40000.06c 7.2802E-03
42000.06c 9.2873E-04
```

Figure 21: Material card in Serpent example

Fuel is the name, 0 indicates that the total density is the sum of the isotopic densities. "tmp 807.15" indicates the temperature of this material during operation. The following list indicates each isotope followed by the atomic density. The isotopes names are written in the way that the library needs them.

3.2.3 Options

Some options are important in a basic input build. The most common is the population option. A typical population setting looks like this:

set pop 5000 1000 100

When setting the population, 3 different numbers are to be set. First, there is the neutron population. The 5000 indicates that at each cycles, 5000 neutrons are going to be simulated. Serpent runs every cycle with the same number of neutrons, whatever the multiplication factor is.

Usually the number of neutrons increases or decreases at every cycle, so Serpent adds or destroys neutrons in order to maintain always the same number.

The number 1000 indicates the number of active cycles, and the number 100, the inactive ones. Both are exactly the same, but the inactive ones don't count for scores. The inactive cycles are used to distribute neutrons through the geometry so that the calculations are more accurate.

Other options are used as well. One of them is the plotting capability of serpent. It can plot sections of the geometry. A typical plot command looks like this:

plot 3 7000 7000 170

Where 3 is the x-y plane, the two 7000 are the resolution of the picture and 170 is the z height of the section. Figure 22 illustrates the output of this plot:

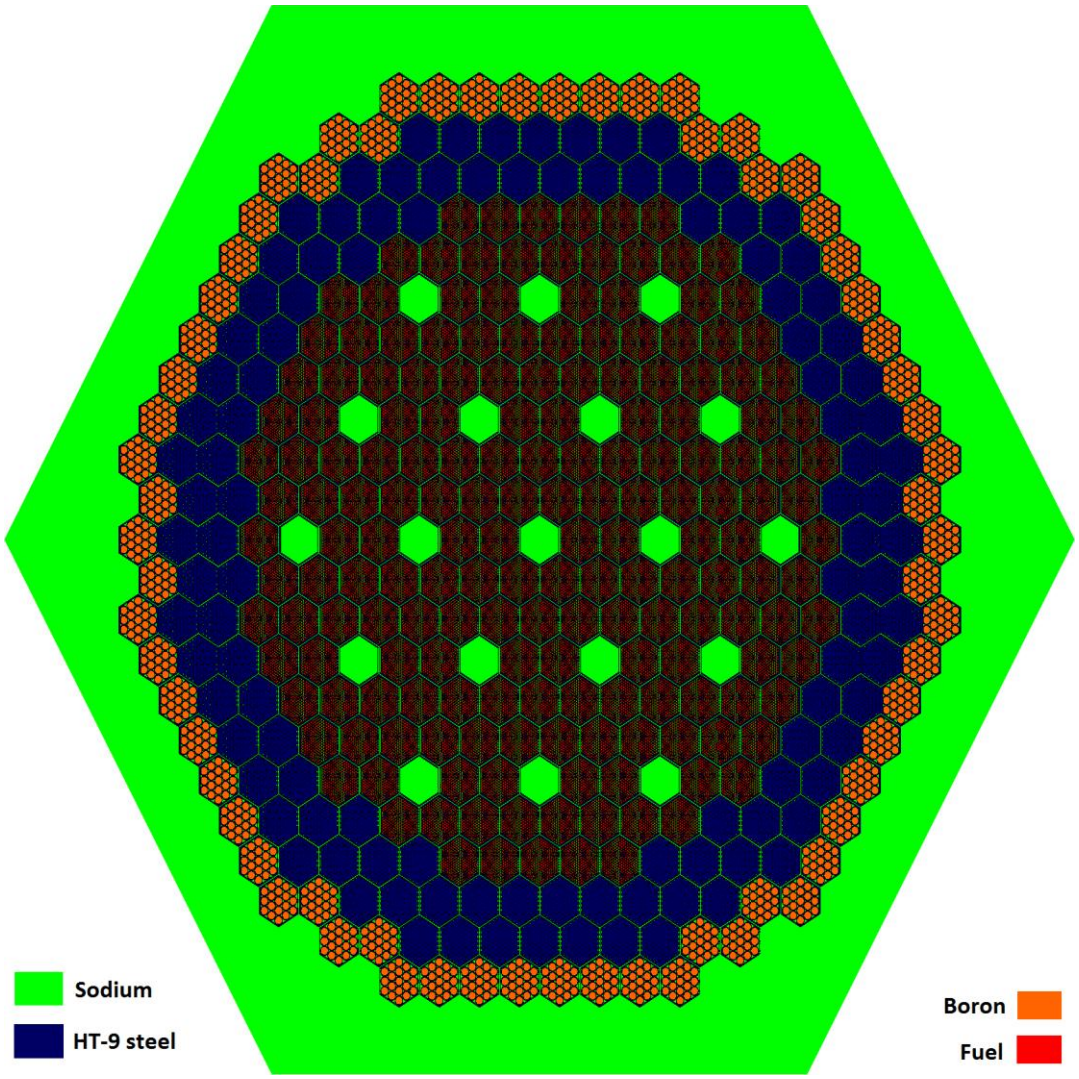

Figure 22: Serpent plot example

The full metallic medium core can be seen, with a section in the active core. A different colour has been given to each material

Another important option is the burnup, which means that serpent will repeat a simulation a given number of times, burning the fuel at each so that the composition changes. This pretends to emulate the real working conditions of a reactor over time. The following example shows a typical burnup:

dep daytot

1
5
10
15
20
25
30
40
50
60
80
100
120
140
180
220
260
300
328.5

"Dep" is the card. Daytot means that the numbers given are the total amount of time spent in the core. This burnup takes a total of 328.5 days, which is the time used for the medium size cores. It is important to note that a burnup calculation is very time-consuming, because is like repeating a steady state simulation over and over. For this reason, a limited number of steps have to be taken.

3.3 Running Serpent

When running Serpent, a command must be executed, followed by the input file that is to be calculated. If not specified otherwise, the program will be run in only one core. The Serpent program must be compiled previously to be able to run simulations. Also, the executable file can be added to the executables directory in Linux. Once this is done, the command to execute an input looks like this:

sss input

where input must be replaced by the name of the file. Once this is done, a series of data will be shown on screen until the calculation is over.

The execution of the program will first calculate some materials and geometry parameters. Then, it will start with the inactive cycles, and once it is done, it will start with the active cycles. These 3 steps can be seen in figures 23 – 25.

```
Reading input file "Test1"...

Processing geometry...
OK.

Reading directory files...
OK.

Calculating isotope fractions...
OK.

Reading data from ACE files:
Isotope  11023.06c (Na-23)...
Isotope  26000.06c (Fe-nat)...
Isotope  28000.06c (Ni-nat)...
Isotope  24000.06c (Cr-nat)...
Isotope  25055.06c (Mn-55)...
Isotope  42000.06c (Mo-nat)...
Isotope  92234.06c (U-234)...
Isotope  92235.06c (U-235)...
Isotope  92236.06c (U-236)...
Isotope  92238.06c (U-238)...
Isotope  93237.06c (Np-237)...
```

Figure 23: Serpent initial calculations example

```
material Fueli5...
material Fuelo1...
material Fuelo2...
material Fuelo3...
material Fuelo4...
material Fuelo5...
material Richboron...
material Naturboron...
OK.

Starting the transport calculation cycle...

Sampling initial source...
OK.

Inactive cycle   1 /  10: k-eff = 0.99120 (DT thresh = 0.9000)
Inactive cycle   2 /  10: k-eff = 1.03204 (DT thresh = 0.9000)
Inactive cycle   3 /  10: k-eff = 1.06052 (DT thresh = 0.9000)
Inactive cycle   4 /  10: k-eff = 1.06879 (DT thresh = 0.9000)
Inactive cycle   5 /  10: k-eff = 1.03331 (DT thresh = 0.9000)
Inactive cycle   6 /  10: k-eff = 1.00975 (DT thresh = 0.9000)
Inactive cycle   7 /  10: k-eff = 1.02429 (DT thresh = 0.9000)
Inactive cycle   8 /  10: k-eff = 1.07735 (DT thresh = 0.9000)
```

Figure 24: Serpent inactive cycle calculations example

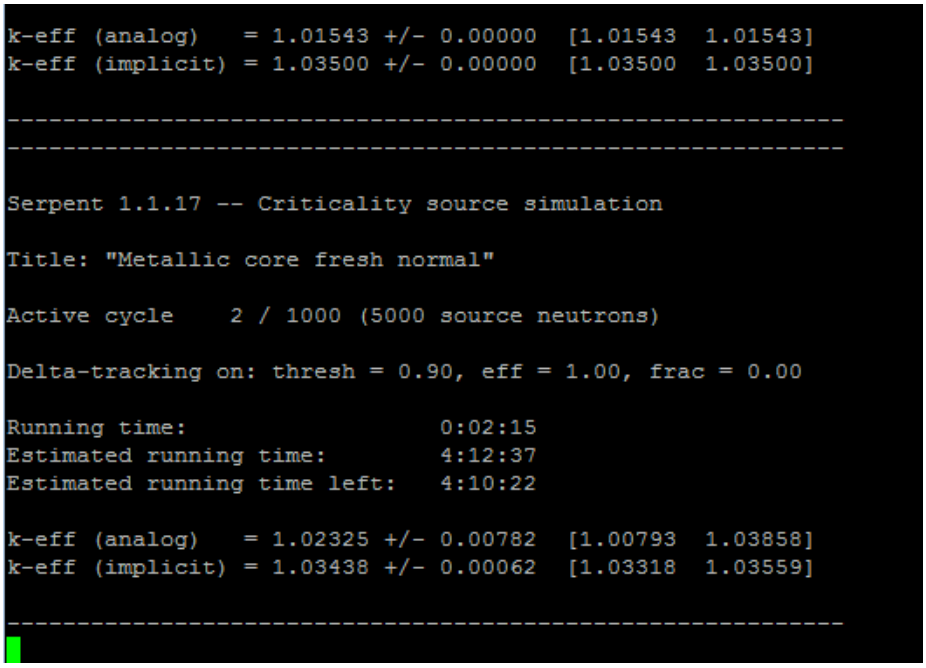

Figure 25: Serpent active cycles calculation example

Once the calculation is over, a message will be shown telling that the calculation has finished, and will return to the command line of the terminal.

To run more than one simulation, they have to be run in the background so that the command line is always available. To do that, the command "nohup" can be used. In the command line it has to look like this:

nohup sss input &

This way, the simulations will be run in the background and the information will be saved in a file in the same folder.

3.4 Output files

Serpent creates different output files at the end of the calculation. The most important one is called "input_res.m", where "input" is the name of the input file. This file gives detailed information about the calculation, like the multiplication factor, the delayed neutron fraction, fission fraction, six factor formula, burnup information, etc. Figure 26 illustrates a typical output file.

In the picture, ANA_KEFF and IMP_KEFF are two ways of calculating the multiplication factor. The value is approximately 1.023, and the number that follows is the standard deviation.

There is other important information that comes in other files. Most important one might be the isotopic composition at the end of a burnup, which has to be called specially in the input file.

```
ENTROPY_Z                 (idx, [1:   2]) = [  4.08096E-01 0.00102 ];
ENTROPY_TOT               (idx, [1:   2]) = [  6.03263E-01 0.00054 ];

% Fission source centre:

SOURCE_X0                 (idx, [1:   2]) = [  1.48513E+00 0.12875 ];
SOURCE_Y0                 (idx, [1:   2]) = [ -3.20579E+00 0.07000 ];
SOURCE_Z0                 (idx, [1:   2]) = [  2.02219E+02 0.00032 ];

% Criticality eigenvalues:

ANA_KEFF                  (idx, [1:   2]) = [  1.04006E+00 0.00200 ];
IMP_KEFF                  (idx, [1:   2]) = [  1.03862E+00 0.00106 ];
COL_KEFF                  (idx, [1:   2]) = [  1.03950E+00 0.00122 ];
ABS_KEFF                  (idx, [1:   2]) = [  1.03862E+00 0.00106 ];
ABS_KINF                  (idx, [1:   2]) = [  1.05582E+00 0.00106 ];
ABS_GC_KEFF               (idx, [1:   2]) = [  1.03862E+00 0.00106 ];
ABS_GC_KINF               (idx, [1:   2]) = [  1.05582E+00 0.00106 ];
IMPL_ALPHA_EIG            (idx, [1:   2]) = [  1.57101E+04 0.03201 ];
FIXED_ALPHA_EIG           (idx, [1:   2]) = [  0.00000E+00 0.00000 ];
GEOM_ALBEDO               (idx, [1:   2]) = [  1.00000E+00 0.00000 ];

% Normalized total reaction rates:

TOT_POWER                 (idx, [1:   2]) = [  9.99999E+08 0.00000 ];
TOT_GENRATE               (idx, [1:   2]) = [  8.86537E+19 0.00015 ];
TOT_FISSRATE              (idx, [1:   2]) = [  2.99597E+19 0.00000 ];
```

Figure 26: Serpent output file example

Chapter 4

Method

The computer used to make the calculations is a 4 core workstation. The cores are Intel Xeon 5130 with a power of 2 Ghz each one. With this machine, the mean time for a steady state calculation is about 2 hours, and for a burnup calculation is about 3 days. The Serpent version used has been 1.1.17.

4.1 Input file

All cores have been built mostly the same way. Lattice universes have been used to build the main core, as well as each subassembly. To build the subassembly pins, a mix has been used between pin universes and cell universes, which has saved a lot of space and time.

Different options have been used for different purposes. The option “egrid” has been used to limit the memory usage of the computer, which exceeded the maximum, and also allowed to make different calculations in parallel. The option “set bc” has been used in order to set the boundary conditions so that the neutrons going away from the core are set to leak and are lost.

For every core, 9 different input files have been used. First, there is a file for the burnup of the core, then the rest are steady state. The burnup had the objective of obtaining the isotope composition of the fuel at the end of equilibrium cycle (EOC). Of the 8 remaining, they are mirrored 4 and 4, so that each one has a replica for BOC and for EOC. There was then 4 types of steady state simulation. One was the normal conditions simulation to obtain most of the data, then one with the fuel temperature doubled to obtain the Doppler coefficient, one with the active core’s sodium voided in order to obtain the void coefficient, and one more with the control rods introduced totally in order to obtain the rod worth.

Each simulation has been performed twice, one for each library. The two libraries used have been ENDF/B-VII and Jeff3.1.1. The reason for using two different libraries was to contrast the results so that they don’t get biased because of the library used. There is no library which use guarantees that the calculations will be perfectly accurate. Serpent sometimes has to deal with problems regarding values not present for some energy, or negative energies, or some other type of errors. It is good then to use different libraries.

4.2 Validation of results

The results given by Serpent are mainly mean values with their respective standard deviation. Their accuracy depends mostly on the number of neutron histories performed, and this is regulated by setting the population in the input. With the option “set pop”, which is mandatory for every calculation, one can set the number of neutrons in every cycle, as well as the number of

active and inactive cycles. Note that the number of histories is the product of neutrons by number of cycles.

When performing a validity test, different calculations are run with increasing number of neutron histories, in order to watch a possible convergence in mean values and choose one number of histories that is reasonable in terms of calculation time. A very big number of neutron histories will give very accurate results, but the calculation time would be too long.

The main problem when setting the number of histories is that it is a combination of three values, as said before. In order to get good convergence results, different tests have been performed to each of the variables.

The value used to look for convergence was the k-eff, or multiplication factor. There are 2 different k-effs at each calculation, the analogue k-eff and the implicit k-eff. Each one is calculated in a different way by Serpent, and results are usually slightly different. To make convergence tests, both values have been used. For the results section, only the analogue one was used. Both values are supposed to be statistically the same, and they are usually within 70% of confidence interval, although sometimes they show some bias.

Figure 27 shows a test with 2000 cycles. There is an oscillation on the first 400 cycles, and then it starts to converge. One can say that at 750 cycles the convergence starts to be optimal. To have a safe margin, 1000 cycles has been chosen as a fixed value for all the calculations in this benchmark.

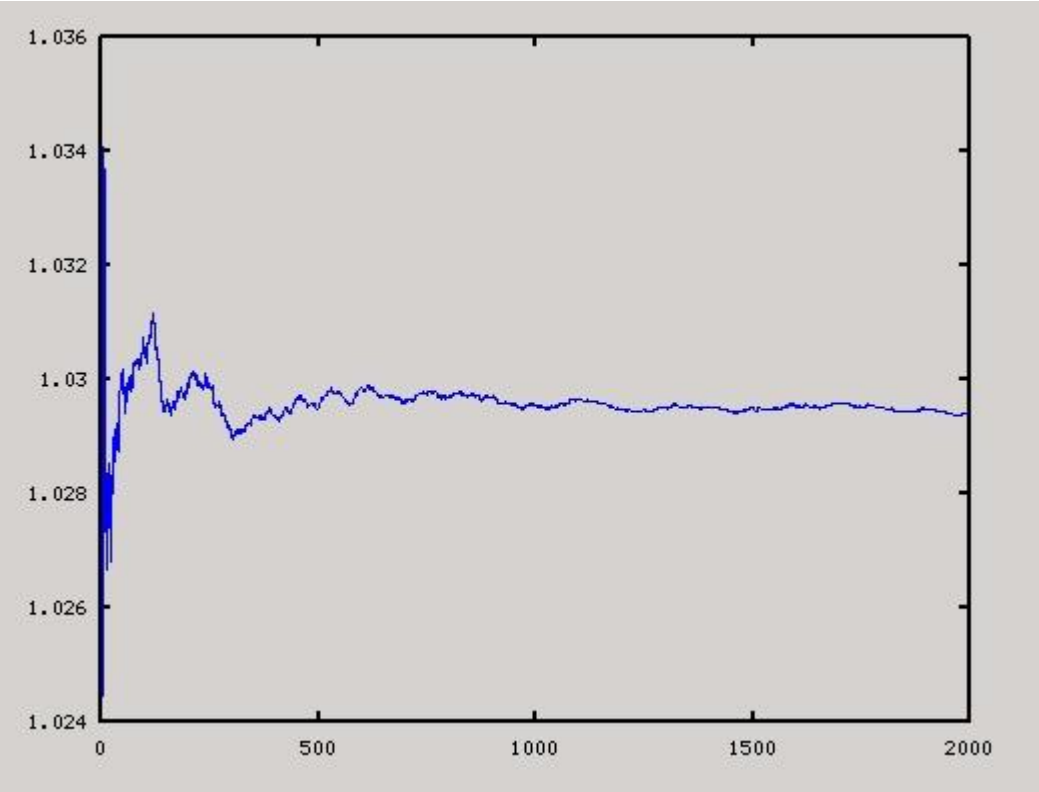

Figure 27: Active cycles convergence test

To choose the number of inactive cycles, the parameter observed was the Shannon Entropy. Figure 28 is a graph representing the evolution of the Shannon Entropy of the neutrons over 1000 cycles.

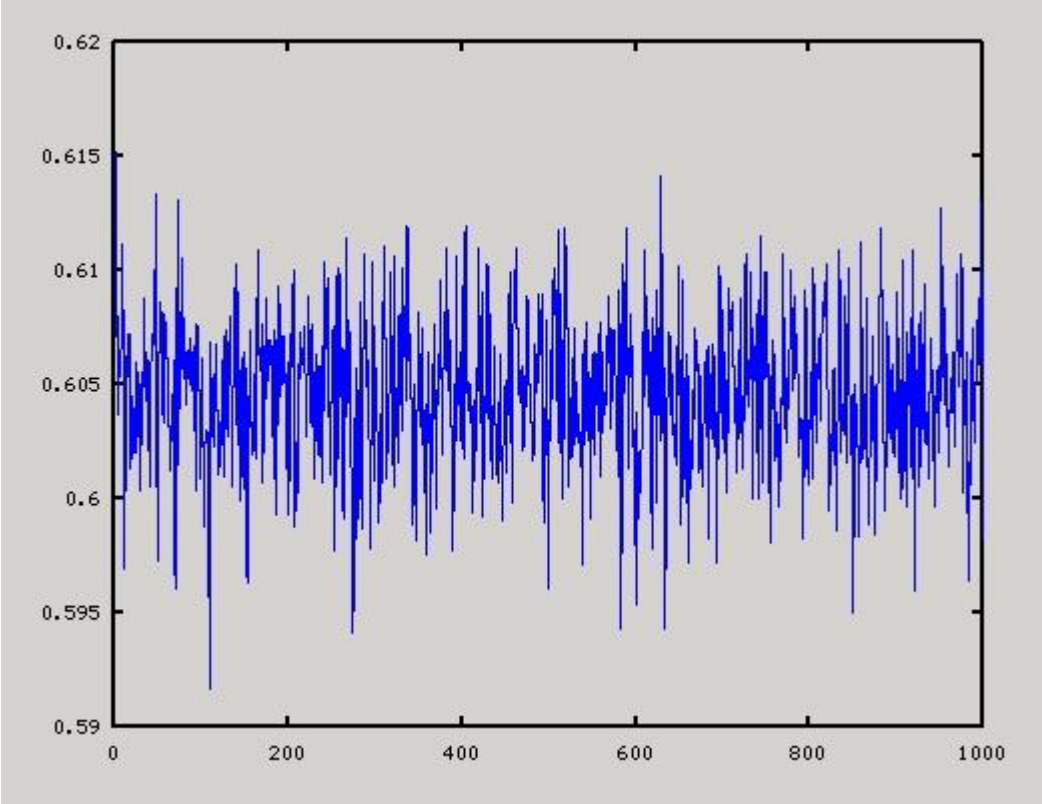

Figure 28: Inactive cycles convergence test

There is a tendency to convergence, but the noise is constant and doesn't seem to diminish. There is only a small bias at the starting cycles, and for that reason, a safe number of 100 inactive cycles has been chosen.

To choose a good neutronic population value, a series of tests with increasing number of neutrons and fixed number of cycles have been performed. The results are shown in figures 29 and 30, one for the analogue k-eff and one for the implicit k-eff.

In these figures, the standard deviation can be seen as dots of different shapes above and beyond the average value for each neutron population number. The standard deviation is written as 2*stdev, meaning two times the standard deviation, or 95% confidence interval. The graphs show an oscillation at the beginning followed by a fast recovery and a tendency to converge. The final choice has been 5000 neutrons, due to the fact that increasing it doesn't show any important improvement in the results.

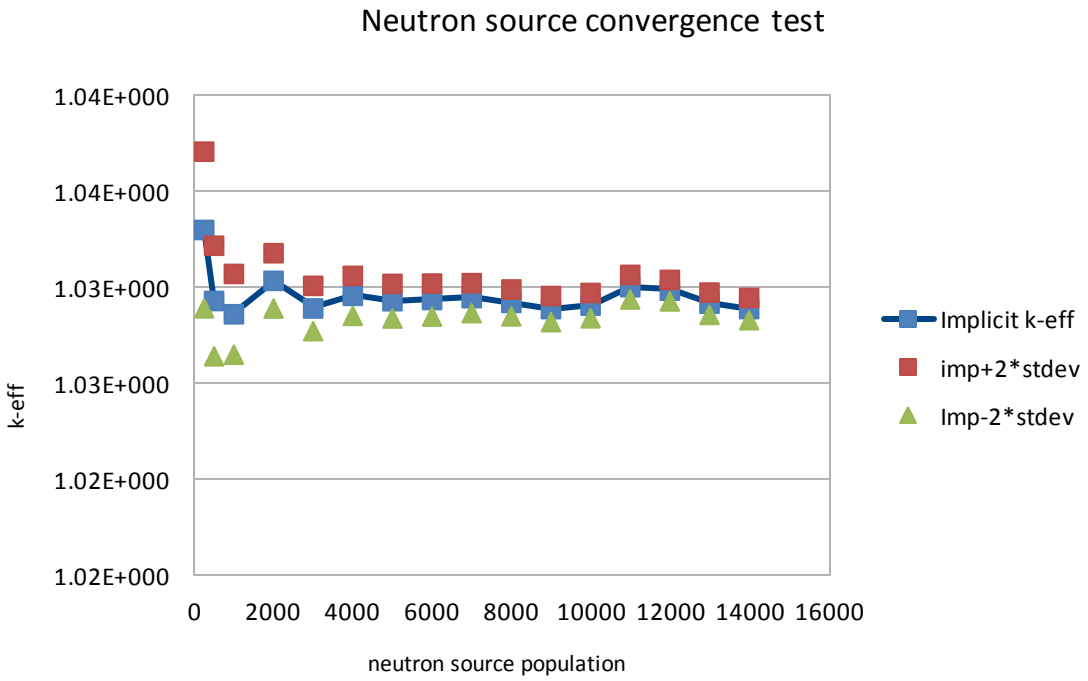

Figure 29: Neutron source convergence test, implicit k-eff

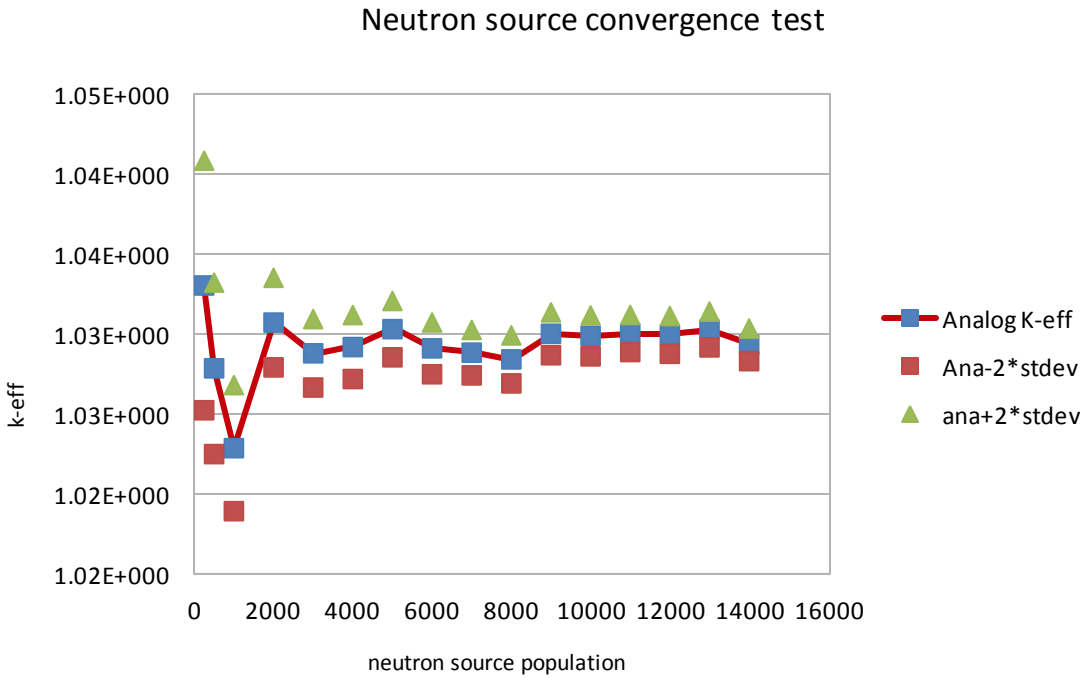

Figure 30: Neutron source convergence test, analogue k-eff

Chapter 5

Results and discussion

The Generation IV International Forum defines a list of parameters required as results for the calculations of the 4 SFR cores. The following is a list of all these parameters:

- Core multiplication factor (k-eff)
- Sodium void worth
- Doppler constant
- Effective delayed neutron fraction
- Average nuclide concentrations for end of equilibrium cycle
- Radial power distribution (integrated over Z for active core length only)
- Control rod worth for all rods totally inserted

The units used in the tables are mostly given in dollars, as a request from the OECD.

5.1 Core multiplication factor

ENDFB/V-VII library

Table 2:Core multiplication factor for ENDF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	1.0170	1.0398	1.0334	1.0242
EOC	1.0262	1.0430	1.0098	1.0063

JEFF-3.1.1 library

Table 3: Core multiplication library for JEFF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	1.0199	1.0452	1.0440	1.0311
EOC	1.0304	1.0451	1.0193	1.0119

From these values, it can be seen that the 4 cores remain slightly supercritical through the whole equilibrium cycle. The values are all reasonable from a design perspective, so it can be assumed that Serpent has the capability to make calculations with results not very biased. It can also be seen that values given by JEFF-3.1.1 library are slightly higher than ENDF/B-VII values.

Given that these cores have the possibility to breed fuel, the multiplication factor increasing from BOC to EOC doesn't seem unreasonable. It can be seen that, for the carbide core, the criticality increases over time. For the oxide core, it remains on the same value, and the medium cores have a decreasing value. It can be guessed that, for the large cores, the transuranics conversion ratio is going to be positive or close to one. For the medium size cores, on the other hand, it will be negative. The transuranics conversion ratio will be discussed in further sections.

5.2 Sodium Void Worth ($)

ENDFB/V-VII library

Table 4: Sodium void worth for ENDF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	3.0932	3.0593	5.2443	5.1300
EOC	4.4481	3.1733	5.5677	5.0085

JEFF-3.1.1 library

Table 5: Sodium void worth for JEFF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	3.8340	2.6461	4.5088	4.3783
EOC	4.3332	3.4979	4.8886	4.8308

With a first look at the data one can tell that in general, the void coefficient increases as the burnup takes place. This can be due to different variables.

One of the reasons could be the americium. As americium increases in a core, so does the void coefficient. The reason is that, because of the disappearance of sodium, there is less moderation (sodium has a small moderation effect) and the spectrum hardens. The americium has a particular cross section that increases as neutron energy increases, so the reactor becomes more critical when there are more fast neutrons.

Americium is not the only reasons why the void coefficient can increase. There are two main components in this phenomenon: spectrum hardening and leakage. The spectrum hardening is what has been discussed in the previous paragraph, but there is also a lot of neutron leakage due to the geometry of the reactor. Smaller reactors tend to have more leakage than the bigger ones. This leakage can be complex to determine and it can change from BOC to EOC, so with only the americium change one cannot say for sure if the void coefficient is going to increase or not.

5.3 Doppler constant (K_D)

ENDFB/V-VII library

Table 6: Doppler constant for ENDF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	-2.3796	-1.7139	-0.6232	-1.5162
EOC	-1.9987	-2.3317	-1.0017	-1.8175

JEFF-3.1.1 library

Table 7: Doppler constant for JEFF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	-1.2075	-2.0835	-1.1969	-2.0553
EOC	-1.8287	-1.7955	-1.1197	-2.0570

The Doppler coefficient, in general, decreases over time because of the decrease of Uranium-238 and increase in Plutonium-240, and also because of the deformation by mechanical creep of the claddings.

In this case, one cannot say whether the Doppler coefficient increases or decreases. The reason is that there is a lot of variability due to the values being close between them and close to zero. The standard deviation is high due to the error propagation, being around 0.5 for a 75% confidence interval.

5.4 Effective delayed neutron fraction

ENDFB/V-VII library

Table 8: Effective delayed neutron fraction for ENDF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	0.0037	0.0036	0.0033	0.0032
EOC	0.0036	0.0035	0.0033	0.0032

JEFF-3.1.1 library

Table 9: Effective delayed neutron fraction for JEFF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	0.0039	0.0037	0.0034	0.0033
EOC	0.0037	0.0036	0.0035	0.0033

The multiplication factor showed signs that the large cores had a higher transuranics conversion ratio than the medium ones, and so does the effective delayed neutron fraction. Typically, cores with higher conversion ratio (and thus, lower TRU/HM enrichment) have a higher delayed neutron fraction.

5.5 Control rod worth

ENDFB/V-VII library

Table 10: Control rod worth for ENDF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	-10.2431	-13.5809	-53.8215	-56.8273
EOC	-12.3313	-14.8624	-56.5861	-59.0391

JEFF-3.1.1 library

Table 11: Control rod worth for JEFF library

	Large Core (3600 MWth)		Medium Core (1000 MWth)	
	Carbide Core	Oxide Core	Metallic Core	Oxide Core
BOC	-9.3764	-13.4720	-52.2721	-56.0059
EOC	-11.3039	-14.5612	-52.6170	-56.8567

The results show that the control rod worth for the medium cores is higher than that of the large cores. This is mainly because the ratio of control assemblies vs. driver assemblies is higher in the medium cores. The results show also a slight tendency to increase in the control rod worth at the end of cycle.

5.6 Average nuclide concentrations for end of equilibrium cycle

Table 13 shows the average concentration of isotopes in each core, in atoms/barn-cm units, as well as the total mass of each element and the transuranics conversion ratio calculated with those values.

Table 12: Average nuclide masses for EOC

		Large Core (3600 MWth)		Medium Core (1000 MWth)	
		Carbide Core	Oxide Core	Metallic Core	Oxide Core
Isotopic number density (atoms/barn-cm)	U234	2.5218E-06	2.5785E-06	2.2540E-06	2.5835E-06
	U235	3.2056E-05	2.6448E-05	2.4910E-05	1.7598E-05
	U236	5.4927E-06	5.3237E-06	3.2119E-06	3.4640E-06
	U238	2.0406E-02	1.7857E-02	1.8306E-02	1.4567E-02
	Np237	6.8320E-06	6.4729E-06	6.9146E-05	3.3481E-05
	Pu238	9.4650E-05	9.0555E-05	1.3728E-04	1.2051E-04
	Pu239	2.0714E-03	1.9718E-03	2.4407E-03	1.9349E-03
	Pu240	1.1135E-03	1.0880E-03	1.5138E-03	1.3992E-03
	Pu241	2.2015E-04	2.1028E-04	2.3029E-04	2.2181E-04
	Pu242	3.5197E-04	3.3926E-04	3.4633E-04	3.0840E-04
	Am241	4.7909E-05	4.6929E-05	1.3761E-04	1.0937E-04
	Am242g	1.4895E-08	1.4324E-08	0.0000E+00	4.4908E-08
	Am242m	1.5209E-06	1.6358E-06	9.8784E-06	1.7672E-06
	Am243	3.2289E-05	3.4522E-05	1.1341E-04	9.9565E-05
	Cm242	2.6600E-06	2.6992E-06	8.8652E-06	1.0181E-05
	Cm243	7.6390E-07	1.4749E-07	5.5844E-07	7.8753E-07
	Cm244	5.3449E-06	7.0278E-06	7.6167E-05	7.8720E-05
	Cm245	7.2689E-07	4.4130E-07	1.7600E-05	2.0725E-05
	Cm246	6.3135E-08	1.9019E-08	9.9274E-06	1.3553E-05
mass (kg)	U	63974	61671	10091	10767
	Pu	12136	12843	2587	3342
	Am	260	291	146	194
	Cm	31	36	64	100
Transuranics Conversion Ratio		1.14	0.82	0.74	0.75

It was discussed earlier that the results indicated that the transuranics conversion ratio is higher for the large cores than for the medium cores and it can be confirmed comparing the composition of the fuel at the beginning and end of cycle.

In the core description file provided by the Generation IV International Forum (GIF), they said that the transuranics conversion ratio of the medium cores was around 0.7, and that can be confirmed with those results. The working mode of each core thus is breeder for the carbide core, converter/breeder for the large oxide core and burner for the medium cores.

5.7 Radial power distribution

The following series of pictures show the distribution of the thermal power in each core for beginning and end of cycle. It can be seen that there is a tendency to have the power output increased on the inner core and constant or decreased on the outer core. This depends a lot on the transuranics conversion ratio at each type of fuel. For example in the large carbide core, the TRU.CR is 1.27 for the inner core and 1.01 for the outer core. This means that the outer core will be more critical at the beginning, while the inner core will breed Plutonium and will be more critical at the end. For all the cores, the TRU.CR is higher for the inner subassemblies than for the outer ones.

Large carbide core

BOC

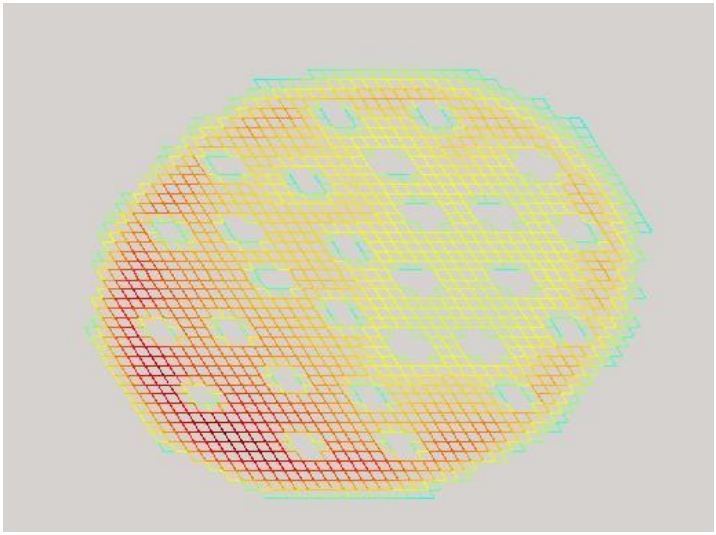

Figure 31: Power distribution for the large carbide core, BOC

EOC

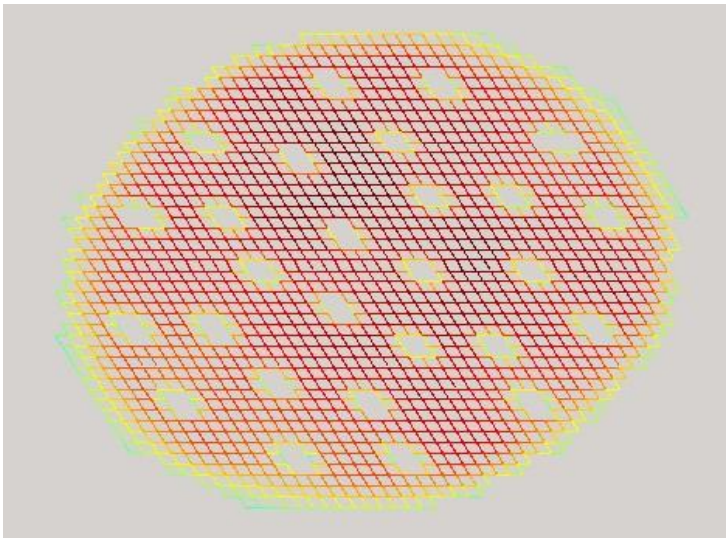

Figure 32: Power distribution for the large carbide core, EOC

Large oxide core

BOC

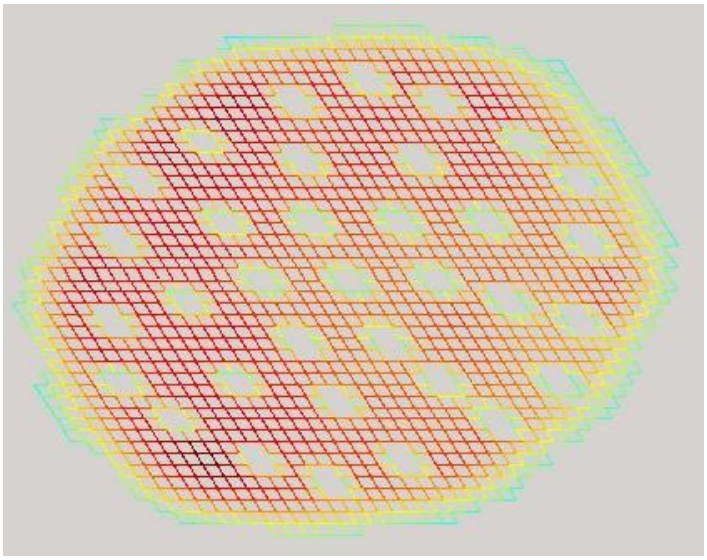

Figure 33: Power distribution for the large oxide core, BOC

EOC

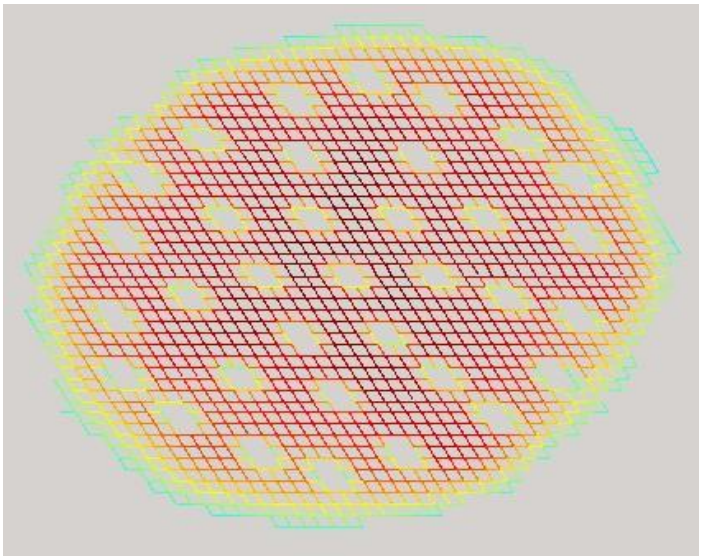

Figure 34: Power distribution for the large oxide core, EOC

Medium metallic core

BOC

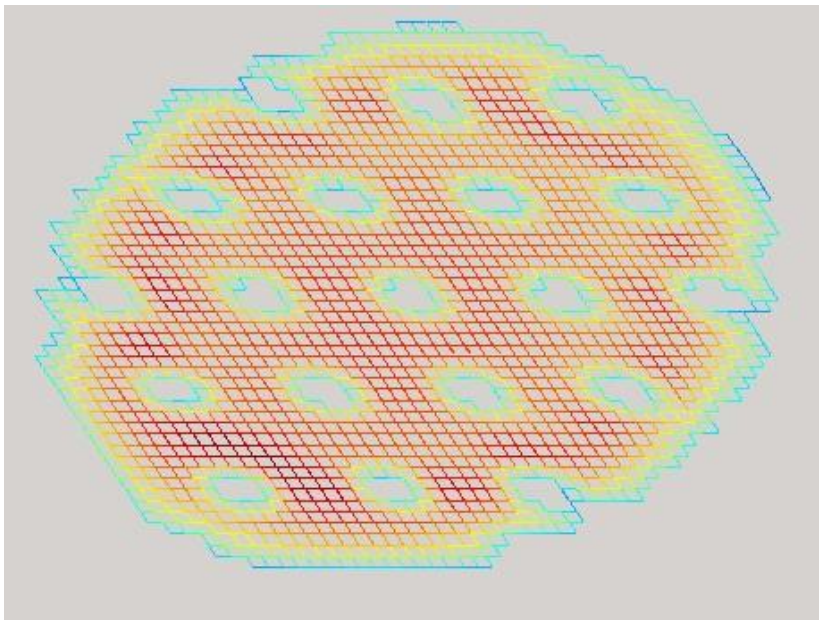

Figure 35: Power distribution for the metallic core, BOC

EOC

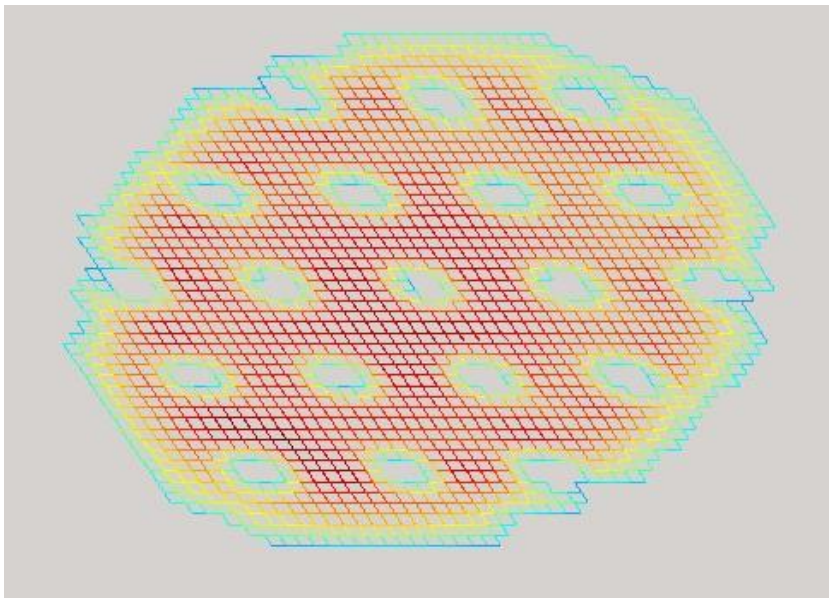

Figure 36: Power distribution for the metallic core, EOC

Medium oxide core

BOC

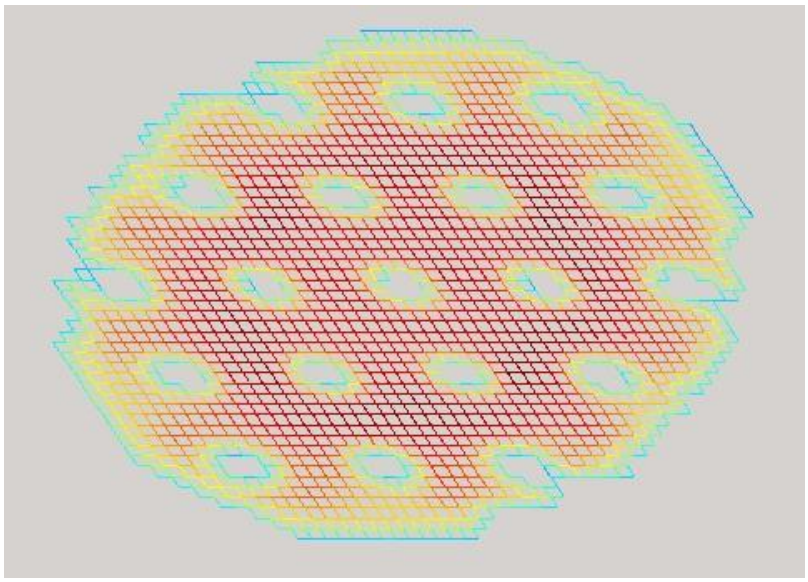

Figure 37: Power distribution for the medium oxide core, BOC

EOC

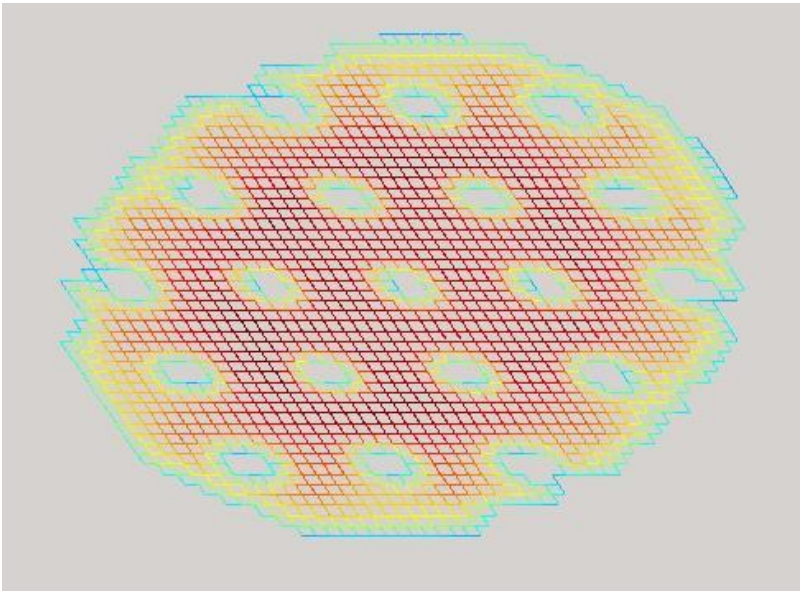

Figure 38: Power distribution for the medium oxide core, EOC

Conclusions

This thesis documented a parametric study performed for four core designs of Sodium Fast Reactors (SFRs): the large core of 3600 MWth with oxide or carbide fuel, and the medium core of 1000 MWth with oxide or metallic fuel. The SERPENT code is chosen in simulations to obtain the global safety parameters: core multiplication factor, sodium void worth, Doppler constant, effective delayed neutron fraction, control rod worth, average nuclide concentrations for end of equilibrium cycle and radial power distribution.

The results obtained show that the four cores remain slightly supercritical through the whole equilibrium cycle. The multiplication factor remains constant at the end of cycle for the large cores, and is reduced in the medium cores. This reflects from the fact that the transuranics conversion ratio is higher for the large cores. The effective delayed neutron fraction is very similar among all cores and doesn't show a significant change through the equilibrium cycle. The sodium void worth is higher for the medium cores due to having more Americium in proportion, and it shows a small increase at the end of cycle. The Doppler coefficient has a value of approximately -2 for all cores except for the metallic core, which has a value of -1. The control rod worth is higher for the medium cores due to them having more control rods in proportion than large cores. The power distribution shows that the fresh cores have the inner fuel colder than the outer fuel, and that the inner fuel becomes hotter at the end of cycle.

The metallic core shows the worst results of all in terms of safety, with a low Doppler coefficient and a high sodium void worth. This can be countered with the reactivity insertion of the control rods, which is significant. On the other side, the large oxide core shows better results, with a higher Doppler constant versus sodium void worth ratio, but at the expense of less control rod worth than the medium cores.

The cross section library choice seems to have a big impact on the results. This is very significant in the Doppler coefficient, with a very high statistical dispersion of results. The JEFF library has a tendency to cause SERPENT to deliver higher multiplication factors than with the ENDF.

Bibliography

[1] Blanchet, D., Buiron, L., Stauff, N. Kim, T.K., and Taiwo, T., Sodium Fast Reactor - Core Definitions, version 1.2, AEN – WPRS report September 19, 2011.

[2] Leppänen, J., 2007c. "*Development of a New Monte Carlo Reactor Physics Code."* D.Sc. Thesis, Helsinki University of Technology (2007). VTT Publications 640.

[3] "*Sodium as a Fast Reactor Coolant*" presented by Thomas H. Fanning. Nuclear Engineering Division. U.S. Department of Energy. U.S. Nuclear Regulatory Commission. Topical Seminar Series on Sodium Fast Reactors. May 3, 2007

[4] Wallenius, J. "*Safety of fast spectrum Gen-IV reactors*" Course on GenIV reactors, KTH

[5] W. S. Yang, 2011. "*Fast Reactor Physics and Computational Methods*" Purdue University School of Nuclear Engineering

[6] Petrelli, G., 2006. "*Nuclear Safety*"

[7] Reuss, P. , 2008. "*Nuclear Physics"*

[8] W. P. Poenitz, 1981, "Most *Important Fission products in a Fast Reactor*"

[9] Fuel Cycles and Advanced Burner Reactors:

http://www.gg.uwyo.edu/content/lecture/energy/nuclearpower/fuel/fuel_cycles/advanced/summary.asp?type=ss&color=993300&Callnumber=14276

[10] Transmutation in a reactor: http://www.ricin.com/nuke/bg/reactor.html

[11] Transuranics: http://www.ead.anl.gov/pub/doc/transuranics.pdf

[12] Nuclear Waste Information: http://www.pnas.org/content/96/7/3432.full

[13] Thompson, A., 2008. "Nuclear Fuel for Advanced Burner Reactors"

[14] Callahan, J.2007. "*Advanced Reactor, Fuel Cycle and Energy Products*" Workshop for Universities

[15] Info on fast breeders: http://hyperphysics.phy-astr.gsu.edu/hbase/nucene/fasbre.html

[16] Ikeda, K., Sekimoto, H., 2011. "Progress in Nuclear Energy"

[17] Steven J. Piet, Samuel E. Bays, Edward A. Hoffman, 2010. "*Description of Transmutation Library for Fuel Cycle System Analyses*" Idaho National Laboratory

[18] R. Ferrer, M. Asgari, S. Bays, B. Forget, 2007. "Computational Neutronics Methods and Transmutation Performance Analyses for Fast Reactors" Idaho National Laboratory