Faculty of Natural Science and Technology
Department of Physics

NTNU

Norwegian University of
Science and Technology

MASTER'S THESIS FOR

STUD. TECHN. LARS KLEMET JAKOBSSON

Thesis started:	20.01.2009
Thesis submitted:	30.06.2009

DISCIPLINE: TECHNICAL PHYSICS

Norsk tittel: *"En studie vedrørende effekten av utbrennings-tilt i brenselselementer på effektfordelingsberegninger i lettvannsreaktorkjerner."*

English title: *"A study about the effect of burnup tilt in fuel assemblies on power distribution calculations of light water reactor cores."*

This work has been carried out at IFE, under the supervision of Makoto Tsuiki

---

Trondheim, 30.06.2009

Kåre Olaussen

Responsible supervisor

Professor at Department of Physics

Acknowledgments

This work has been carried out at the Institute of Energy Technology in Halden and I would like to thank my supervisor Makoto Tsuiki for guidance and help on this thesis. I would also like to thank Professor Kåre Olaussen, responsible supervisor at the Department of Physics, for additional help and guidance. William H. Beere has also been to great help giving guidance on the calculations in this thesis.

Abstract

Norsk:

Raske og nøyaktige analyseprogrammer er nødvendig for beregninger av oppførselen til lettvannsreaktorkjerner. Slike programmer er nødvendige for å effektivt drive reaktorkjernen med mindre marginer til begrensende parametre og for å optimalisere måter brenselet blir satt inn i kjernen på.

Metoden "The Variational Nodal Expansion Method" er en ny og raskere metode som kan øke beregningshasigheten for lettvannsreaktorkjerner. Denne methoden anvender variasjonsprinsippet og bruker forhåndsberegnede ekspansjonsfunksjoner og koeffisienter til å beregne oppførselen til reaktorkjernen. Testberegninger er gjort i et to-dimensjonalt system med et program som bruker denne metoden, og resultatene er sammenlignet med resultater fra beregninger med et tregere og mere nøyaktig program. Resultatene viser at metoden er litt unøyaktig for utbrenningsberegninger, og en metode for forbedre dette er beskrevet.

English:

Fast and accurate analysis programs are needed for calculations of the behavior of light water reactor cores. Such programs are needed in order to efficiently operate the reactor core with smaller margins to limiting parameters and to optimize the fuel loading patterns.

The Variational Nodal Expansion Method is a new and faster method that can increase the calculation speed for light water reactor cores. This method utilizes the variational principle using pre-calculated expansion functions and coefficients in order to calculate the behavior of the core. Test calculations are done in a two-dimensional system with a program using this method, and the results are compared to results from calculations done with a slower and more precise program. The results shows that the program is a bit inaccurate for burnup calculations, and a method to improve this is outlined.

Contents

1 Introduction **1**

2 Nuclear reactor technology **3**
2.1 History . . . 3
2.2 Nuclear fission . . . 5
2.3 Nuclear reactors . . . 6
2.3.1 Pressurized Water Reactor . . . 8
2.3.2 Boiling Water Reactor . . . 8
2.4 Burnup . . . 9
2.5 Modeling . . . 11

3 Neutron transport **13**
3.1 Neutron transport equation . . . 13
3.2 Multigroup transport equation . . . 17

4 Programs **21**
4.1 Overview . . . 22
4.2 HELIOS . . . 23
4.3 FCM2D . . . 24
4.3.1 Characteristics method . . . 24
4.3.2 Boundary conditions . . . 25
4.4 VCOEF2D . . . 27
4.4.1 Source expansion function . . . 27
4.4.2 Boundary expansion function . . . 29
4.4.3 Source expansion coefficients . . . 30
4.4.4 Boundary value expansion coefficients . . . 31
4.5 VNEM2D . . . 33
4.6 Summary . . . 35

5 Calculations **37**
5.1 Burnup calculations . . . 37
5.1.1 Single assembly calculations . . . 38
5.1.2 Multi-assembly calculations . . . 39

6 Results **41**
6.1 The effective multiplication factor 41
6.2 Pin powers 44

7 Improvement of VNEM **47**
7.1 Two stages method 47
7.2 Correction of VNEM coefficients 48

8 Conclusion **53**

A HELIOS input data **57**
A.1 AURORA single assembly input 57
A.1.1 Assembly type 1 57
A.1.2 General input data 59
A.1.3 Material properties 61
A.1.4 Pin structure 64
A.1.5 Assembly structure 67
A.1.6 Burnup cases for assembly type 1 85
A.1.7 Assembly type 5 87
A.1.8 Burnup cases for assembly type 5 89
A.2 ZENITH input data 91
A.2.1 Calculation of cross sections 92
A.3 AURORA multi-assembly input 95
A.3.1 Main input 95
A.3.2 System structure 97
A.3.3 Burnup cases for the system 97
A.4 ZENITH multi-assembly input 99
A.4.1 Calculation of pinpowers 100

B VNEM input data **103**
B.1 Selecting cross sections from HELIOS 103
B.2 FCM2D input 104
B.3 VCOEF2D input 105
B.4 VNEM2D input 107

Chapter 1

Introduction

Many countries are dependent upon nuclear power to cover their energy demands, and nuclear power will still be an important energy source for many years to come. Improvement of the nuclear reactor technology is therefore necessary to improve safety and economy, and to reduce the amount of radioactive waste from nuclear power plants. One of the improvements is to introduce more complex fuel composition into the reactor core to prolong the time between fuel replacements and extract more energy from the fuel.

Accurate programs are needed to predict the behavior of the reactor core with different fuel compositions. Currently existing accurate programs are using brute force methods, such as Monte Carlo to calculate the behavior of the reactor core. The calculations done with these programs are impractical for whole core calculations because they require too much computer resources and therefore use too long time to do the calculations. There are also fast programs that predict the behavior of reactor core but they are not very accurate. One of the improvements to be done in nuclear reactor technology is therefore to develop faster accurate programs that can predict the behavior of the reactor core.

A new method, the Variational Nodal Expansion Method (VNEM), is being developed to increase the calculation speed for reactor core calculations. This method is described in chapter 4. The target is to get the method close to as precise as the brute force methods, and at the same time a calculation speed which is much faster. Comparisons of three-dimensional full core VNEM calculations with measured data are done and they show promising results [1].

In this work the VNEM method is used in a two dimensional transport code called VNEM2D. VNEM2D needs pre-calculated expansion functions and coefficients as input. The expansion functions and coefficients are calculated in a code called VCOEF2D. VCOEF2D also needs pre-calculated data, and these data are generated using a brute force program called HELIOS and another program called FCM2D[1]. However, in the VNEM method there are done some approximations that

make the calculated results inaccurate for long time use of fuel with heterogeneous composition. This inaccuracy does mainly occur in the radial direction of the reactor core (which is why we use a 2D system), and it is investigated using HELIOS to do reference calculations and compare this to the results obtained using the VNEM method.

The historical background and an introduction to nuclear reactor technology is written in chapter 2. An understanding of this chapter and the neutron transport chapter 3 is necessary for a basic understanding of the technique used for the VNEM calculations. Chapter 4 explains the basics of the two dimensional version of this technique. An outline of the calculations can be found in chapter 5 and the results from these calculations can be found in chapter 6.

The results shows that the current version of the VNEM method is a bit inaccurate for burnup calculations. A suggested method to improve the accuracy for burnup calculations is outlined in chapter 7.

[1]FCM2D is not used in the production version of the system.

Chapter 2

Nuclear reactor technology

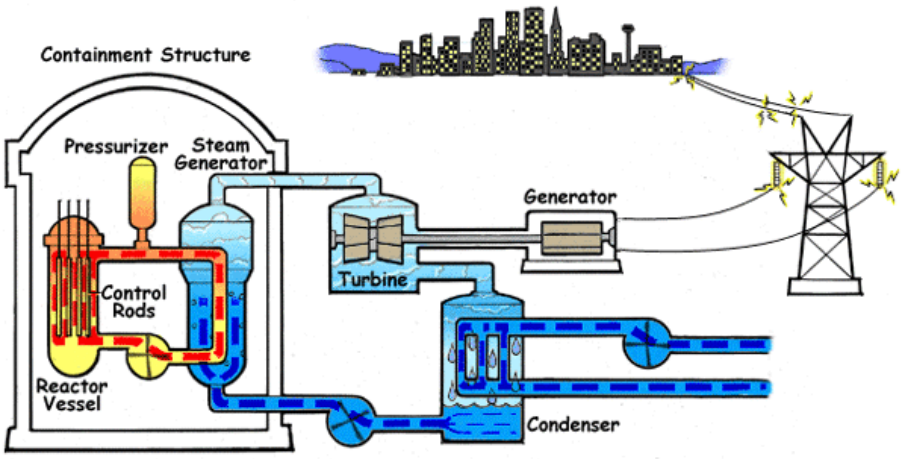

Figure 2.1: A simple drawing of the design of a PWR nuclear power reactor. There are two closed water circuits, and also one open that uses sea- or freshwater to cool the steam in the condenser [2].

2.1 History

The neutron was discovered in 1932 by James Chadwick [3]. The following year Leó Szilárd proposed that if any neutron-driven process released more neutrons than those required starting it, an expanding nuclear chain reaction might result. He filed a patent for his idea of a simple nuclear reactor the following year [4].

In 1938 Otto Hahn and Fritz Strassmann showed that barium was formed after bombarding uranium with neutrons [5]. Lise Meitner and Otto Robert Frisch interpreted these results as being nuclear fission, splitting of uranium into lighter elements after the absorption of a neutron [6]. Frisch confirmed this experimentally afterwards [7]. Otto Hahn and Fritz Strassmann also predicted that additional

neutrons could be released through the fission process [8].

In 1939 Leó Szilárd and Enrico Fermi discovered neutron multiplication in Uranium, proving that nuclear chain reaction was indeed possible [9]. After this discovery, Albert Einstein signed a letter that was sent to President Franklin D. Roosevelt, warning that Nazi Germany might be attempting to build an atomic bomb (The Einstein-Szilárd letter).

Fermi and his team also created the first artificial self-sustaining nuclear chain reaction, called Chicago Pile-1, at the University of Chicago on December 2, 1942. Most of the early reactors were made to produce plutonium for nuclear weapons. The first commercial nuclear power station, Calder Hall in Sellafield, England was opened in 1956.

There have been self-sustaining nuclear chain reactions going on in nature about 2 billion years ago at Oklo in Gabon [10]. The conditions for a natural nuclear reactor were predicted in 1959 by Paul Kuroda [11] and the existence of this phenomenon was discovered in 1972.

Currently 14 percent of the global electricity production comes from nuclear power plants. Nuclear reactors are also used as a neutron source for research and cancer treatment. Some radioactive isotopes used in medicine can only be produced in research reactors. Radioactive isotopes produced in nuclear reactors are also used for research and in industry.

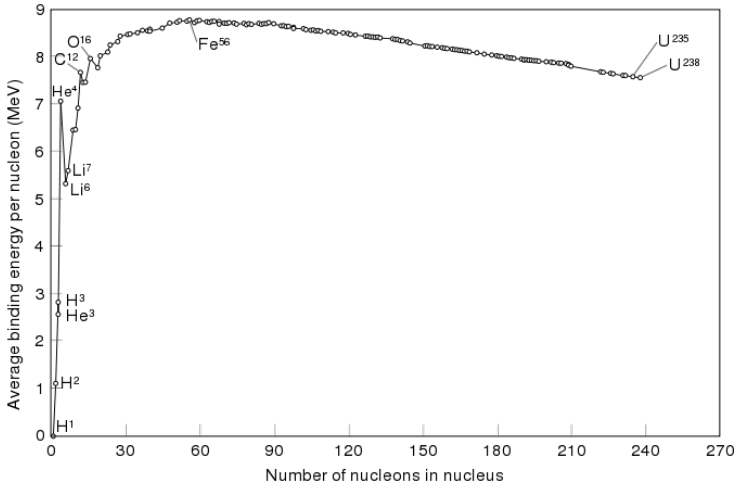

Figure 2.2: The binding energy curve. Energy will be released if a heavy element such as uranium-235 is split into two lighter elements through fission. Light elements such as hydrogen can for example be combined into helium and release energy through fusion [12].

2.2 Nuclear fission

Nuclear fission is a process in which a heavy element is split into two lighter elements, and it is the main process that releases energy in a nuclear reactor. As we can see in figure 2.2 the binding energy per nucleon is lower for intermediate elements than for heavy elements, which means that energy will be released if heavy elements are split into two lighter elements. Nuclear fission can occur spontaneously for certain isotopes of heavy elements such as uranium and thorium.

Free neutrons are often produced when heavy elements undergo fission. These free neutrons can be absorbed in the nucleus of atoms through neutron capture. Some elements become unstable once they have absorbed one neutron and undergo fission immediately afterwards. This process is called induced fission and is essential in a nuclear reactor. The most abundant element that undergoes induced fission in nature is uranium-235. Elements that can undergo induced fission are said to be fissile.

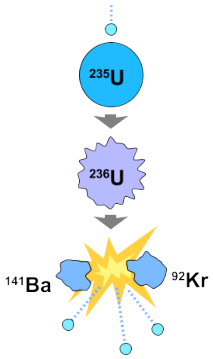

Figure 2.3: Example of a fission event. Uranium-235 absorbs a neutron and splits into barium-141 and krypton-92, and three new neutrons are released [13].

Two or three neutrons are usually released through the fission of an element. These neutrons can induce a new fission event in which more free neutrons are produced. This chain reaction is what keeps a nuclear reactor running, and it also makes nuclear fission bombs explode and triggers fusion bombs. In a nuclear reactor there are too few neutrons released from spontaneous fission in the fuel to start the chain reaction, and an independent neutron source is therefore placed in the reactor to start the chain reaction.

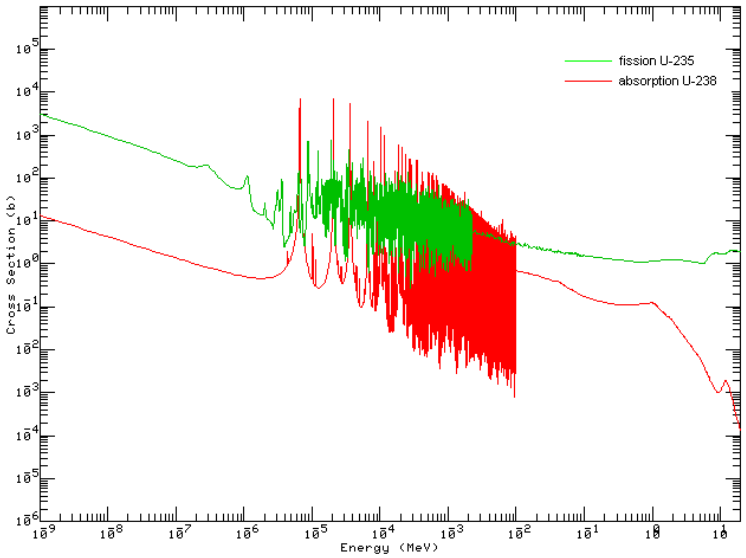

Figure 2.4: Fission cross section for U-235 and absorption cross section for U-238. The cross sections have a lot of resonance peaks for intermediate energies [14].

2.3 Nuclear reactors

The neutrons released through fission have an energy of about 2 MeV. In figure 2.4 we see that the fission cross section for uranium-235 is higher for lower energies. This means that the neutrons need to be slowed down in order to get many of them to be absorbed in uranium-235. We say that they need to be moderated.

Water is very good for the process of moderating the neutrons and is therefore used in very many nuclear reactors. We say that water is a good moderator. This is because the hydrogen in water is very light, and the neutrons can therefore loose very much speed/energy through scattering on the hydrogen nucleus.

There are also three other things that also can happen to the high energy/fast neutrons in addition to scattering;

- The fast neutrons can be absorbed and split a heavy element due to its high energy (fast fission).

- The neutrons can be absorbed without any fission occurring.

- The neutrons can escape form the reactor.

We want to slow down the neutrons in a short time because the absorption cross section for U-238 is high for intermediate energies (the red curve in figure 2.4). The neutrons that have been slowed down are called thermal neutrons and they have an energy of about 0.025eV.

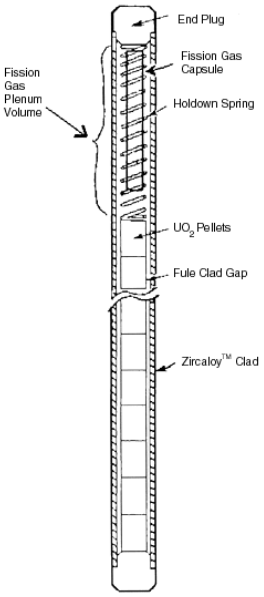

Figure 2.5: Typical design of a fuel rod. Zirconium-alloy tube filled with helium and containing cylindrical pellets of uranium [15].

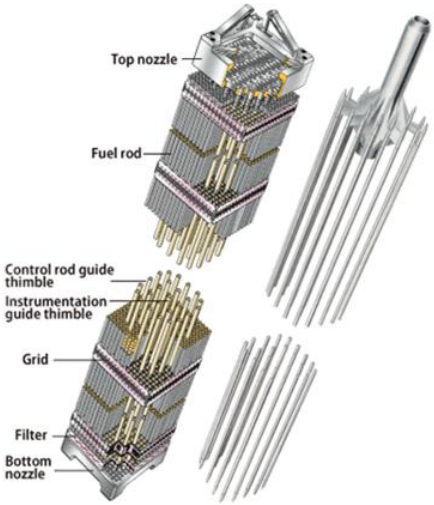

Figure 2.6: Fuel assembly and control rod cluster. The control rod cluster is inserted into the fuel assembly and can be adjusted to regulate the power of the reactor [16].

2.3.1 Pressurized Water Reactor

There are several different types of nuclear reactors. The most common type of power producing reactors is the Pressurized Water Reactor (PWR). The fuel for PWR reactors is uranium oxide which is processed into ceramic cylindrical pellets. Metallic uranium is not used because it has a lower melting point. The uranium oxide pellets have a diameter and a length about one centimeter, and are usually stacked inside hollow metal rods; from now on called fuel rods. These rods are about four meters long and consist of an alloy of Zirconium mixed with other metals. Zirconium has a very low absorption cross-section of thermal neutrons and do therefore not prevent the neutrons from getting back into the fuel where they can split more uranium-235.

Quadratic bundles with about 17x17 fuel rods are put together; this is from now on called a fuel assembly or just an assembly. There are also a number of control rods that can be adjusted to be a certain length into the assembly to control the power level, see figure 2.6. About 200 fuel assemblies are put together within a cylindrical vessel, and makes up the core of the reactor. Water runs through the core to cool the fuel and moderate the neutrons. The heated water is used to generate steam in a secondary water circuit where the steam runs a steam turbine which drives an electric generator. Figure 2.1 is a drawing of how modern PWR nuclear power plants are designed.

2.3.2 Boiling Water Reactor

The second most common type of nuclear reactors is in many ways similar to the PWR. The biggest difference is that the steam is generated in the reactor core in stead of being generated in a secondary water circuit. This kind of a reactor is called a Boiling Water Reactor (BWR) since the water is boiling in the reactor core. The design of the assemblies and the control rods are also a bit different. The assemblies have metal walls and internal water tubes, and the control rods are cross shaped blades that are inserted from below in between a group of four assemblies.

The programs that are used to do calculations for PWR reactors can also be used for the BWR reactors because of the similarity, but it is harder to do precise calculations due to the less predictable moderation ability of the boiling water.

Uranium in nature contains about 0.7 percent uranium-235. This is a too low concentration to run a PWR or BWR reactor, and the natural uranium is therefore enriched so that it contains between 2 and 4 percent uranium-235 before it is used as fuel. In uranium with below 1% uranium-235 too many neutrons are absorbed in uranium-238 without a new fission event occurring when light water is used as moderator.

Am235 15 m EC,α	**Am236** EC,α	**Am237** 73.0 m 5/2(-) EC,α	**Am238** 98 m 1+ EC,α	**Am239** 11.9 h (5/2)- EC,α	**Am240** 50.8 h (3-) EC,α	**Am241** 432.2 y 5/2- α,sf	**Am242** 16.02 h 1- EC,β- *	**Am243** 7370 y 5/2- α,sf	**Am244** 10.1 h (6-) β- *
Pu234 8.8 h 0+ EC,α	**Pu235** 25.3 m (5/2+) EC,α	**Pu236** 2.858 y 0+ α,sf	**Pu237** 45.2 d 7/2- EC,α *	**Pu238** 87.7 y 0+ α,sf	**Pu239** 24110 y 1/2+ α,sf	**Pu240** 6563 y 0+ α,sf	**Pu241** 14.35 y 5/2+ β-,α,sf,...	**Pu242** 3.733E+5 y 0+ α,sf	**Pu243** 4.956 h 7/2+ β-
Np233 36.2 m (5/2+) EC,α	**Np234** 4.4 d (0+) EC	**Np235** 396.1 d 5/2+ EC,α	**Np236** 1.54E5 y (6-) EC,β-,α,... *	**Np237** 2.144E+6 y 5/2+ α,sf	**Np238** 2.117 d 2+ β-	**Np239** 2.3565 d 5/2+ β-	**Np240** 61.9 m (5+) β- *	**Np241** 13.9 m (5/2+) β-	**Np242** 5.5 m (6) β- *
U232 68.9 y 0+ α	**U233** 1.592E+5 y 5/2+ α,sf	**U234** 2.455E+5 y 0+ α,n,sf,... 0.0055	**U235** 7.038E+8 y 7/2- α,^{20}Ne,sf,... * 0.7200	**U236** 2.342E7 y 0+ α,sf	**U237** 6.75 d 1/2+ β-	**U238** 4.468E+9 y 0+ α,β-β-,sf,... 99.2745	**U239** 23.45 m 5/2+ β-	**U240** 14.1 h 0+ β-	**U241**
Pa231 32760 y 3/2- α,sf	**Pa232** 1.31 d (2-) EC,β-	**Pa233** 26.967 d 3/2- β-	**Pa234** 6.70 h 4+ β-,sf *	**Pa235** 24.5 m (3/2-) β-	**Pa236** 9.1 m 1(-) β-	**Pa237** 8.7 m (1/2+) β-	**Pa238** 2.3 m (3-) β-,sf	**Pa239**	**Pa240**
Th230 7.538E+4 y 0+ α,sf	**Th231** 25.52 h 5/2+ β-,α	**Th232** 1.405E10 y 0+ α,sf 100	**Th233** 22.3 m 1/2+ β-	**Th234** 24.10 d 0+ β-	**Th235** 7.1 m (1/2+) β-	**Th236** 37.5 m 0+ β-	**Th237** 5.0 m β-	**Th238**	

Figure 2.7: Table of some heavy isotopes with increasing number of neutrons to the right and increasing number of protons upwards [17].

2.4 Burnup

Burnup is a measure of how many atoms that has underwent fission during a certain period of time. In a reactor that is operational the amount of uranium-235 decreases, but at the same time plutonium-239 and 241 are formed from uranium-238 through neutron capture. Plutonium-239 and 241 are fissile like uranium-235, and do therefore contribute more and more to the reactor effect as the fuel gets older, where we by older mean for how long the fuel has been inside a reactor that is operational. Fission of plutonium-239 and 241 does not, however, replace the decrease in effect due to depletion of uranium-235 in the light water reactors. In figure 2.7 we can see how different heavy isotopes decay, and how long their half-lives are. Uranium-239 formed by neutron capture in uranium-238 will beta-decay into neptunium-239. Neptunium-239 will subsequently beta-decay into plutonium-239;

$$\mathrm{n} + {}^{238}\mathrm{U} \rightarrow {}^{239}\mathrm{U} \xrightarrow{\beta^-} {}^{239}\mathrm{Np} \xrightarrow{\beta^-} {}^{239}\mathrm{Pu}. \tag{2.1}$$

It can be mentioned that thorium-232 also can be used in the same way as uranium-238 inside a reactor. Uranium-233 is fissile and it is formed from thorium-232 through neutron capture in the same way as plutonium-239 is formed from uranium-238. This is called the thorium cycle;

$$\mathrm{n} + {}^{232}\mathrm{Th} \rightarrow {}^{233}\mathrm{Th} \xrightarrow{\beta^-} {}^{233}\mathrm{Pa} \xrightarrow{\beta^-} {}^{233}\mathrm{U}. \tag{2.2}$$

Both uranium-238 and thorium-232 are called fertile isotopes, which mean that they can be converted into fissile material.

Some of the fission products generated from fission will also affect the power of the reactor. In figure 2.8 we can see the amount of the different elements that are formed after fission of uranium-235. Other fissile isotopes have a similar distribution. Xenon-135 and samarium-149 are among the produced isotopes, and they have a large neutron absorption cross section. They will therefore reduce the effect of the reactor and has to be taken into account, and many other isotopes do also have to be taken into account when doing precise calculations.

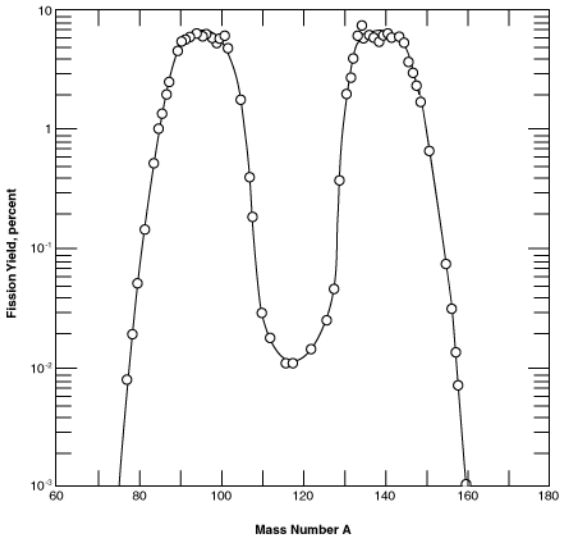

Figure 2.8: Distribution of fission products formed by fission of U-235 [18].

A typical PWR can operate with a total thermal power of about 3000MW. The reactor core is typically loaded with about 80 tons of uranium oxide. One third of the assemblies are usually replaced every 18 to 24 months. The energy produced is represented as GigaWatt-days per Ton (GWd/T). Thus after running the reactor for 24 months the energy produced will be

$$E = \frac{2 \cdot 3000\text{MW} \cdot 365\text{d}}{80\text{T}} = 27375\text{MWd/T} = 27.375\text{GWd/T}. \tag{2.3}$$

Ideally a nuclear power reactor should run all the time, but that is not possible because the amount of uranium-235 will eventually become so low that it is impossible to maintain a chain reaction. At that point the fuel will need to be replaced with fresh fuel, and the reactor has to be shut down while changing the fuel. One way to increase the interval between fuel changes could be to enrich the uranium to more than 4 % uranium-235, but as the enrichment of uranium-235 is increased, it becomes difficult to shut down the reactor at the beginning of a burnup cycle. Therefore burnable poisons are used in stead of some of the control rods in some of the fuel assemblies. Boron or gadolinium is generally used, and they will gradually be removed as they absorb neutrons and are changed into isotopes that absorb

much less neutrons. This makes the assemblies with burnable poisons contribute more and more to the effect of the reactor as the effect from assemblies without burnable poisons decrease. Therefore one can keep the reactor running for a longer time without changing fuel when using burnable poisons. Boron are also added to the water when the fuel is fresh so reduce the power of the reactor, and the amount of boron are then reduced as the fuel get older.

In summary the things that happen when the fuel burns are;

- Fissile isotopes decrease
- Fertile isotopes become fissile isotopes
- Fission products are generated
- Burnable poisons decrease because of neutron absorption

2.5 Modeling

The neutrons have a distribution in space and energy which evolves in time. A neutron flux is defined to describe these properties of the neutrons. Energy is mainly released through fission which is induced by neutrons. The neutron flux is therefore the most important factor to calculate in order to find the power level inside a reactor core. The differential equation that needs to be solved to find the neutron flux is called the neutron transport equation. This equation is outlined in the next chapter.

Chapter 3

Neutron transport

The behavior of a nuclear reactor is mainly governed by the free neutrons in the reactor core. So in order to predict how the reactor will behave, we need to calculate the number of neutrons, how they are distributed in space and their energy.

The neutrons can be described as point (not quantum) particles to good approximation. The wavelength λ of a neutron is

$$\lambda = \frac{\hbar}{p}, \tag{3.1}$$

where $\hbar$ is Planck's constant divided by 2π and p is the momentum of the particle. With a neutron energy of 0.025eV we get

$$\lambda = \frac{4.14 \times 10^{-15}\text{eVs}}{2\pi\sqrt{2m_0 E}} = 2.88 \times 10^{-11}\text{m}, \tag{3.2}$$

which is much less than the distance between atoms in a solid. 0.025eV is the most probable energy of a neutron in thermodynamic equilibrium at 290 K within the reactor core, so there is only a negligible number of neutrons having such a low energy that the point particle description is seriously wrong. The neutrons also have spin and a magnetic moment that could affect their behavior, but in practical situations this seems to be negligible.

3.1 Neutron transport equation

We have a neutron angular density

$$N(\mathbf{r}, \mathbf{\Omega}, E, t) \left[\frac{\text{neutrons}}{\text{cm}^3 \cdot \text{sr} \cdot \text{eV}}\right], \tag{3.3}$$

which is defined as the number of neutrons in the position $\mathbf{r}$ having the velocity direction $\mathbf{\Omega}$ and energy E at the time t.

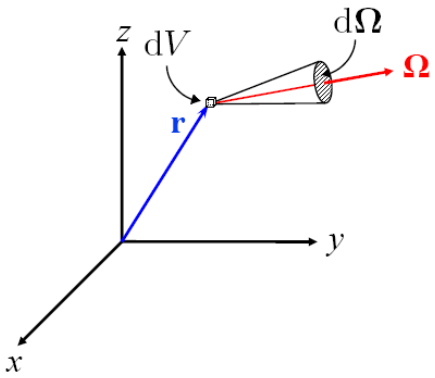

Figure 3.1: Volume element $\mathrm{d}V$ and directional element $\mathrm{d}\mathbf{\Omega}$.

The macroscopic cross section for a certain type of interaction, x, with the neutrons is

$$\Sigma_x(\mathbf{r}, E) = \sum_n \mathrm{N}_n \cdot \sigma_{x,n}(\mathbf{r}, E), \tag{3.4}$$

where N_n is the number of atoms of a certain element per unit volume and $\sigma_{x,n}(\mathbf{r}, E)$ is the microscopic cross section of type x for that element. For neutrons the total cross section is the sum of four different partial cross sections

$$\Sigma(\mathbf{r}, E) = \Sigma_s(\mathbf{r}, E) + \Sigma_{s'}(\mathbf{r}, E) + \Sigma_c(\mathbf{r}, E) + \Sigma_f(\mathbf{r}, E), \tag{3.5}$$

where $\Sigma_s(\mathbf{r}, E)$ and $\Sigma_{s'}(\mathbf{r}, E)$ are the elastic and inelastic scattering cross sections. $\Sigma_c(\mathbf{r}, E)$ is the neutron capture cross section and $\Sigma_f(\mathbf{r}, E)$ is the cross section for fission.

Now we define a differential cross section

$$\Sigma_x(\mathbf{r}, E') f_x(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E), \tag{3.6}$$

where $f_x(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E)\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E$ is the probability that if a neutron of direction $\mathbf{\Omega}'$ and energy E' has a collision, there will emerge from the collision a neutron in the direction interval $\mathrm{d}\mathbf{\Omega}$ about $\mathbf{\Omega}$ with energy in $\mathrm{d}E$ about E.

The total probability of neutron transfer from $\mathbf{\Omega}', E'$ to $\mathbf{\Omega}, E$ is

$$\begin{aligned} \Sigma(\mathbf{r}, E') f(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E) &= \frac{1}{4\pi} \Sigma_f(\mathbf{r}, E') \nu(\mathbf{r}, E' \to E) \\ &+ \sum_{x \neq f} \Sigma_x(\mathbf{r}, E') f_x(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E), \end{aligned} \tag{3.7}$$

where

$$f_f(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E) = \frac{1}{4\pi} \nu(\mathbf{r}, E' \to E), \tag{3.8}$$

because we can assume that the neutrons from fission are emitted isotropically in the lab system. And $\nu(\mathbf{r}, E' \to E)$ is normalized so that

$$\int\int \nu(\mathbf{r}, E' \to E)\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E = \bar{\nu}(\mathbf{r}, E'), \tag{3.9}$$

where $\bar{\nu}(\mathbf{r}, E')$ is the average number of neutrons released by a fission at $\mathbf{r}$ caused by a neutron with energy E'. Since scattering does not change the number of neutrons, f_s and $f_{s'}$ are normalized to one;

$$\int\int f_s(\mathbf{r}, E' \to E)\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E = 1, \tag{3.10}$$

and similarly for $f_{s'}$. The probability of transfer for neutron capture, f_c, is of course zero since it removes the free neutron.

If we have a group of neutrons and follow it for a time Δt, there will remain

$$N(\mathbf{r}, \mathbf{\Omega}, E, t)[1 - \Sigma(\mathbf{r}, E)v\Delta t]\ \mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E \tag{3.11}$$

neutrons in this group, where v is the speed of the neutrons. We regard neutrons that undergo a collision as lost from this group, and the ramaining neutrons will arrive at the position $\mathbf{r} + \mathbf{\Omega}v\Delta t$ at the time $t + \Delta t$. Some neutrons may enter this group at the same time.

The number of neutrons entering the group after collisions is

$$\left[\int\int \Sigma(\mathbf{r}, E')f(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E)v'N(\mathbf{r}, \mathbf{\Omega}, E, t)\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E'\right]\ \mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E\ \Delta t, \tag{3.12}$$

and the number of neutrons entering the group from sources

$$Q(\mathbf{r}, \mathbf{\Omega}, E, t)\ \mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E\ \Delta t. \tag{3.13}$$

To simplify the equations we define

$$\Sigma \equiv \Sigma(\mathbf{r}, E),$$

$$\Sigma' f \equiv \Sigma(\mathbf{r}, E')f(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E)$$

and

$$Q \equiv Q(\mathbf{r}, \mathbf{\Omega}, E, t).$$

Then by adding equation (3.11), (3.12) and (3.13) we get that the number of neutrons at the position $\mathbf{r} + \mathbf{\Omega}v\Delta t$ at the time $t + \Delta t$ is

$$\begin{aligned} & N(\mathbf{r} + \mathbf{\Omega}v\Delta t, \mathbf{\Omega}, E, t + \Delta t)\ \mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E \\ = \ & N(\mathbf{r}, \mathbf{\Omega}, E, t)[1 - \Sigma v\Delta t]\ \mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E \\ + \ & \left[\int\int \Sigma' f v' N(\mathbf{r}, \mathbf{\Omega}, E, t)\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E'\right]\ \mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E\ \Delta t \\ + \ & Q\ \mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E\ \Delta t. \end{aligned} \tag{3.14}$$

If we divide this equation by $\mathrm{d}V\ \mathrm{d}\mathbf{\Omega}\ \mathrm{d}E\ \Delta t$, and let $\Delta t \to 0$ we get

$$\lim_{\Delta t\to\infty} \frac{N(\mathbf{r}+\mathbf{\Omega}v\Delta t, \mathbf{\Omega}, E, t+\Delta t) - N(\mathbf{r}, \mathbf{\Omega}, E, t)}{\Delta t} + \Sigma v N(\mathbf{r}, \mathbf{\Omega}, E, t)$$

$$= \int\int \Sigma' f v' N(\mathbf{r}, \mathbf{\Omega}, E, t)\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E' + Q \tag{3.15}$$

Then by subtracting and adding $N(\mathbf{r}, \mathbf{\Omega}, E, t+\Delta t)$ in the numerator in the first term we get

$$\lim_{\Delta t\to\infty} \frac{N(\mathbf{r}+\mathbf{\Omega}v\Delta t, \mathbf{\Omega}, E, t+\Delta t) - N(\mathbf{r}, \mathbf{\Omega}, E, t)}{\Delta t}$$

$$\begin{aligned} &= \lim_{\Delta t\to\infty} \frac{N(\mathbf{r}+\mathbf{\Omega}v\Delta t, \mathbf{\Omega}, E, t+\Delta t) - N(\mathbf{r}, \mathbf{\Omega}, E, t+\Delta t)}{\Delta t} \\ &+ \lim_{\Delta t\to\infty} \frac{N(\mathbf{r}, \mathbf{\Omega}, E, t+\Delta t) - N(\mathbf{r}, \mathbf{\Omega}, E, t)}{\Delta t} \\ &= \frac{\partial}{\partial t} N(\mathbf{r}, \mathbf{\Omega}, E, t) + v\mathbf{\Omega}\cdot\nabla N(\mathbf{r}, \mathbf{\Omega}, E, t) \end{aligned} \tag{3.16}$$

The neutron flux, Φ is defined as

$$\Phi = N(\mathbf{r}, \mathbf{\Omega}, E, t)\cdot v, \tag{3.17}$$

and hence we get the neutron transport equation

$$\frac{1}{v}\cdot\frac{\partial\Phi}{\partial t} + \mathbf{\Omega}\cdot\nabla\Phi + \Sigma\Phi = \int\int \Sigma' f \Phi'\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E' + Q. \tag{3.18}$$

A system containing fissile elements can be regarded as being either subcritical, critical or supercritical. If we have an initial population of neutrons that decrease with time if there is no external or internal neutron source, we say that the system is subcritical. If the neutron population increase, we say that the system is supercritical. The system is critical if the initial number of neutrons can be maintained as time evolves without any sources.

We are interested in a system that is critical. To obtain the criticality condition we introduce auxiliary eigenvalues. We choose to replace $\nu(\mathbf{r}, E' \to E)$ with $\nu(\mathbf{r}, E' \to E)/k$, and k can be varied to obtain the criticality condition $k = k_{eff}$. For a critical system the neutron flux will be time independent, i.e. $\partial\Phi/\partial t = 0$. The source term are also taken into account through k_{eff}[19], and hence we get the equation

$$\mathbf{\Omega}\cdot\nabla\Phi + \Sigma\Phi = \frac{1}{k_{eff}}\int\int \frac{1}{4\pi}\nu\Sigma^f\Phi'\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E' + \int\int \Sigma^s f^s \Phi'\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E', \tag{3.19}$$

where we have defined

$$\nu\Sigma^f \equiv \Sigma_f(\mathbf{r}, E')\nu(\mathbf{r}, E' \to E) \tag{3.20}$$

and

$$\begin{aligned}\Sigma^s f^s &\equiv \sum_{x\neq f} \Sigma_x(\mathbf{r},E')f_x(\mathbf{r};\mathbf{\Omega}',E'\rightarrow\mathbf{\Omega},E)\\ &= \Sigma_s(\mathbf{r},E')f_s(\mathbf{r};\mathbf{\Omega}',E'\rightarrow\mathbf{\Omega},E)\\ &+ \Sigma_{s'}(\mathbf{r},E')f_{s'}(\mathbf{r};\mathbf{\Omega}',E'\rightarrow\mathbf{\Omega},E),\end{aligned} \tag{3.21}$$

since absorption removes the neutron. Equation (3.19) is the neutron transport equation for a critical system.

3.2 Multigroup transport equation

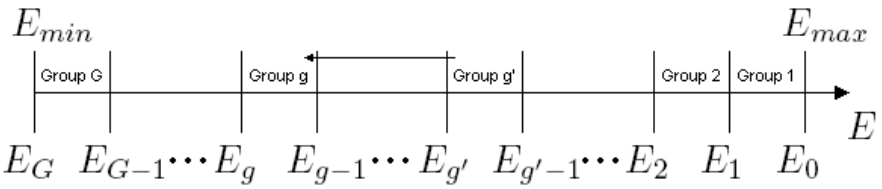

Figure 3.2: Neutrons in an energy group g' losing energy through scattering and ending up un another group g.

To solve the neutron transport equation numerically we use a multigroup method. We divide the energy of the neutrons into energy groups where the neutron flux for each group becomes

$$\Phi_g(\mathbf{r},\mathbf{\Omega}) = \int_{E_g}^{E_{g-1}} \Phi(\mathbf{r},\mathbf{\Omega},E)\ \mathrm{d}E. \tag{3.22}$$

The scalar neutron flux is

$$\Phi_g(\mathbf{r},E) = \int_{4\pi} \Phi(\mathbf{r},\mathbf{\Omega},E)\ \mathrm{d}\mathbf{\Omega}, \tag{3.23}$$

and the scalar neutron flux for each energy group is

$$\Phi_g(\mathbf{r}) = \int_{4\pi}\int_{E_g}^{E_{g-1}} \Phi(\mathbf{r},\mathbf{\Omega},E)\ \mathrm{d}E\ \mathrm{d}\mathbf{\Omega} = \int_{4\pi} \Phi_g(\mathbf{r},\mathbf{\Omega})\ \mathrm{d}\mathbf{\Omega}. \tag{3.24}$$

The transport equation is integrated over one energy group to find the group wise neutron flux

$$\begin{aligned}&\int_{E_g}^{E_{g-1}} \mathbf{\Omega}\cdot\nabla\Phi\ \mathrm{d}E + \int_{E_g}^{E_{g-1}} \Sigma\Phi\ \mathrm{d}E = \frac{1}{k_{eff}}\int\int_{E_g}^{E_{g-1}}\int \frac{1}{4\pi}\nu\Sigma^f\Phi'\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E\ \mathrm{d}E'\\ &+ \int\int_{E_g}^{E_{g-1}}\int \Sigma^s f\Phi'\ \mathrm{d}\mathbf{\Omega}'\ \mathrm{d}E\ \mathrm{d}E',\end{aligned} \tag{3.25}$$

We do also assume isotropic scattering to simplify the calculations. This means that

$$f_s(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E) = \frac{1}{4\pi} f_s(\mathbf{r}, E' \to E) = 0 \tag{3.26}$$

since the energy is conserved for elastic scattering, and

$$f_s(\mathbf{r}; \mathbf{\Omega}', E' \to \mathbf{\Omega}, E) = \frac{1}{4\pi} f_s(\mathbf{r}, E' \to E). \tag{3.27}$$

Hence we get

$$\begin{aligned}
&\int \int_{E_g}^{E_{g-1}} \int \Sigma^s f^s \Phi' \, \mathrm{d}\mathbf{\Omega}' \, \mathrm{d}E \, \mathrm{d}E' \\
= \; & \frac{1}{4\pi} \int \int_{E_g}^{E_{g-1}} \int \Sigma_{s'}(\mathbf{r}, E') f_{s'}(\mathbf{r}, E' \to E) \Phi(\mathbf{r}, \mathbf{\Omega}', E') \, \mathrm{d}\mathbf{\Omega}' \, \mathrm{d}E \, \mathrm{d}E' \\
= \; & \frac{1}{4\pi} \sum_{g'} \int_{E_{g'}}^{E_{g'-1}} \Sigma_{s'}(\mathbf{r}, E') f_{g,s'}(\mathbf{r}, E') \Phi(\mathbf{r}, E') \, \mathrm{d}E' \\
= \; & \frac{1}{4\pi} \sum_{g'} \Sigma^s_{g' \to g}(\mathbf{r}) \Phi_{g'}(\mathbf{r}).
\end{aligned} \tag{3.28}$$

Here $f_{g,s'}(\mathbf{r}, E')$ is the probability that a neutron with energy E' is transferred to energy group g after a collision, and

$$\Sigma^s_{g' \to g}(\mathbf{r}) = \frac{\int_{E_{g'}}^{E_{g'-1}} \Sigma_{s'}(\mathbf{r}, E') f_{g,s'}(\mathbf{r}, E') \Phi(\mathbf{r}, E') \, \mathrm{d}E'}{\Phi_{g'}(\mathbf{r})} \tag{3.29}$$

is the cross section for a neutron to be transferred from energy group g' to g. In the same way we also have

$$\begin{aligned}
&\int \int_{E_g}^{E_{g-1}} \int \nu \Sigma^f \Phi' \, \mathrm{d}\mathbf{\Omega}' \, \mathrm{d}E \, \mathrm{d}E' \\
= \; & \int \int_{E_g}^{E_{g-1}} \int \Sigma^f(\mathbf{r}, E') \nu(\mathbf{r}, E' \to E) \Phi(\mathbf{r}, \mathbf{\Omega}', E') \, \mathrm{d}\mathbf{\Omega}' \, \mathrm{d}E \, \mathrm{d}E' \\
= \; & \sum_{g'} \int_{E_{g'}}^{E_{g'-1}} \Sigma^f(\mathbf{r}, E') \nu_g(\mathbf{r}, E') \Phi(\mathbf{r}, E') \, \mathrm{d}E' \\
= \; & \chi_g(\mathbf{r}) \sum_{g'} \nu \Sigma^f_{g'}(\mathbf{r}) \Phi_{g'}(\mathbf{r}),
\end{aligned} \tag{3.30}$$

where χ_g is the fission spectrum factor for energy group g and $\nu_g(\mathbf{r}, E')$ is the probability that a neutron from fission will be within energy group g when fission is caused by a neutron with energy E'.

$$\nu \Sigma^f_{g'} = \frac{\int_{E_{g'}}^{E_{g'-1}} \Sigma^f(\mathbf{r}, E') \nu_g(\mathbf{r}, E') \Phi(\mathbf{r}, E') \, \mathrm{d}E'}{\Phi_{g'}(\mathbf{r})} \tag{3.31}$$

is the fission neutron generation cross section for a neutron in energy group g'. The transport equation can then be written as

$$\mathbf{\Omega} \cdot \nabla \Phi_g(\mathbf{r}, \mathbf{\Omega}) + \Sigma_g(\mathbf{r}, \mathbf{\Omega}) \Phi_g(\mathbf{r}, \mathbf{\Omega}) = \frac{S_g(\mathbf{r})}{4\pi}, \tag{3.32}$$

where

$$S_g(\mathbf{r}) = \sum_{g'} \Sigma^s_{g' \to g}(\mathbf{r}) \Phi_{g'}(\mathbf{r}) + \frac{\chi_g(\mathbf{r})}{k_{eff}} \sum_{g'} \nu \Sigma^f_{g'}(\mathbf{r}) \Phi_{g'}(\mathbf{r}) \tag{3.33}$$

and

$$\Sigma_g(\mathbf{r}, \mathbf{\Omega}) = \frac{\int_{E_g}^{E_{g-1}} \Sigma(\mathbf{r}, \mathbf{\Omega}, E) \Phi(\mathbf{r}, \mathbf{\Omega}, E) \, \mathrm{d}E}{\Phi_g(\mathbf{r}, \mathbf{\Omega})}. \tag{3.34}$$

This multigroup transport equation could be solved exactly for a reactor core by inserting the neutron cross sections and the geometrical distribution for the materials in the reactor and use brute force numerical methods (e.g. Monte Carlo or collision probability). This however, proves to be impossible. The main problem is that there is no currently existing supercomputer solving this equation for a reactor core in a short enough time to be of any practical use.

Equation (3.32) is solved with various methods in the next chapter. The Variational Nodal Expansion Method (VNEM) is one of the methods used, and it is a new and faster method that can be used for whole core calculations.

Chapter 4

Programs

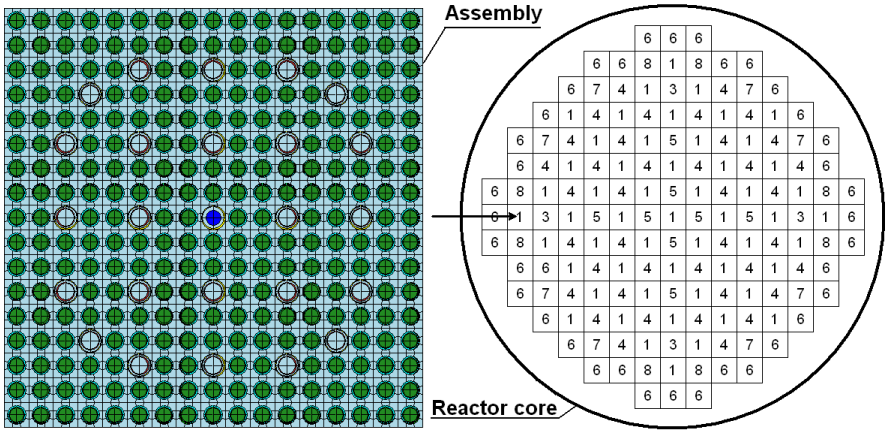

Figure 4.1: 2-D geometry of assembly and reactor vessel seen from above. The assemblies are stacked beside each other within the reactor vessel, where the numbers represent different kinds of assembly types (see table 4.1). The green disks in the assembly are uranium fuel and the blue disk in the center is a tube which can be used to measure the effect within the assembly with a detector. The remaining circles are guide tubes for the control rods and the grey background color represents water with boron.

Initially we have a reactor core with uranium fuel, water and supporting construction materials. All these materials have different cross sections for interacting with the neutrons, and experimentally determined cross sections for the different materials are looked up from a library when doing calculations. The programs use these cross sections, the geometry of the core and neutron transport methods to calculate the neutron flux, the power level and the criticality of the reactor core. We use a fuel loading pattern from the Swedish nuclear power plant "Ringhals 3" as an example of a reactor core.

Assembly type	Enrichment	Number of rods with burnable poisons
0	reflector	0
1	2.11% U-235	0
2	2.60% U-235	0
3	”	12
4	”	16
5	”	20
6	3.10% U-235	0
7	”	12
8	”	16

Table 4.1: Properties of different assembly types.

4.1 Overview

VNEM is a new faster method that can be used for whole core calculations. It needs pre-calculated neutron angular flux and cross sections for single assembly systems with reflective boundary condition. HELIOS is an external program that is used to calculate the burnup of the fuel and the cross sections for the burnup levels we are interested in. FCM2D is then used to calculate the neutron angular flux using these cross sections.

The calculations are done in a two-dimensional system in the radial direction of the reactor core since the inaccuracy is expected to mainly occur in the radial direction, and the data flow goes like this:

- HELIOS: 2D single assembly burnup calculations for the different assemblies. Cross sections are transferred to FCM2D.
- FCM2D: 2D single assembly calculation of neutron angular flux for a cell homogenized assembly.
- VCOEF2D: Using the single assembly cross sections calculated in HELIOS and the angular flux calculated in FCM2D, VCOEF2D calculates the VNEM expansion functions and the VNEM coefficients (see section 4.4).
- VNEM2D: Using the assembly-wise VNEM coefficients and expansion functions calculated in VCOEF2D, VNEM2D calculates k_{eff}, the scalar flux and the power distribution for a multi-assembly system.

The details of each program and calculation step are described in the following sections.

The core is octant symmetric, so we only need to do calculations for one quadrant of the core with reflective boundary condition at the symmetry lines. And we use reflector assemblies as a model of the outer boundary of the core, see figure 4.2

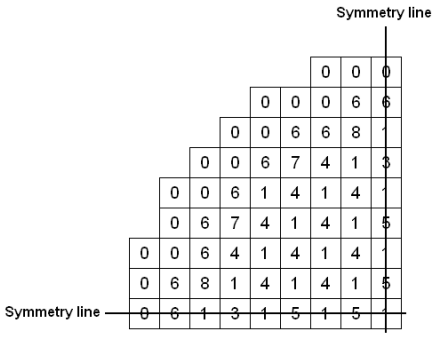

Figure 4.2: One quadrant of the core with symmetry lines.

4.2 HELIOS

HELIOS uses a current coupling and collision probability transport method which only can be used in 2-dimensional geometry[20]. This transport method requires so much computer resources that it will use a very long time to do a calculation for a whole core such as the one in figure 4.1.

The user must specify the geometry of the system, how it should be divided into space meshes and boundary conditions. The size of the space meshes are dependent on the calculation speed and accuracy required. Smaller space meshes make the calculations more accurate but also slower, and vice a versa with larger space meshes.

HELIOS can then do a burnup calculation with user specified burnup steps where the user chooses which data to save. Both the properties of the system and how the results should be saved are specified in an input processor called AURORA.

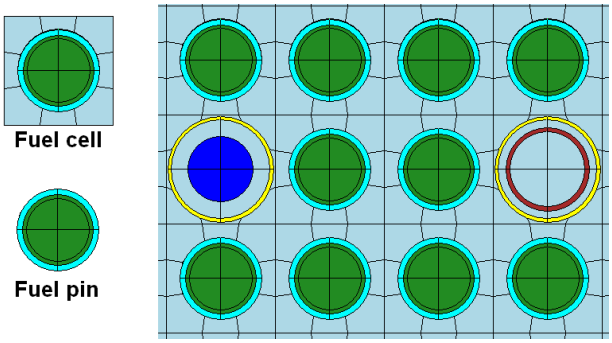

Figure 4.3: A close up look at one part of the assembly shows better how it is divided into space meshes before calculations are done in HELIOS. The black lines indicate the borders of the meshes, the turquoise circles represent the zirconium-alloy tubes that encapsulate the fuel, and the red and the yellow circles represent other kinds of zirconium-alloys.

HELIOS is used to do detailed two dimensional burnup calculations for one single assembly. Reflective boundary condition is assumed when doing the calculations. This is the same as assuming that the quadratic assembly surrounded infinitely by the same assembly as itself, and the reflective boundary condition is therefore also called infinite lattice boundary condition. The 2-D geometry of the assembly and the space meshes can be seen in figure 4.1 and 4.3.

The cross sections obtained for each burnup step are homogenized over each fuel cell, where the quadratic regions surrounding each fuel pin in figure 4.3 are called fuel cells.

We use an output processor, ZENITH, to select the fuel cell homogenized cross sections from the burnup calculations and save them into a data file in text format. Another program (hermes.exe) can then convert the text file with the cross sections into files in a format that can be used as input in FCM2D.

See the HELIOS manual[20] for a closer explanation on how HELIOS works and how input and output are specified.

4.3 FCM2D

FCM2D is a characteristics transport code that solves the neutron transport equation with rectangular geometry of the core, appropriate boundary conditions and cross sections. The rectangular core is composed of a number of assemblies where each assembly is divided into regular space meshes.

FCM2D uses cell homogenized cross sections from HELIOS single assembly burnup to calculate the angular flux for the cell homogenized assemblies, where we still use reflective boundary condition. This flux is needed as input for VCOEF2D. Output from FCM2D can also be compared with output from VNEM2D to check the accuracy of VNEM transport method.

4.3.1 Characteristics method

Calculation tracks are generated in a way used in CACTUS module of WIMS[21]. The tracks are used to integrate the transport equation, and the same set of tracks can be used for all the assemblies in the core.

The transport equation along a track takes the form

$$\frac{\partial \Phi_g(s, \mathbf{\Omega}_i)}{\partial s} + \Sigma_g \Phi_g(s, \mathbf{\Omega}_i) = \frac{S_g(s)}{4\pi}, \tag{4.1}$$

where s is the one-dimensional coordinate along the track i and $\mathbf{\Omega}_i$ is the solid angle of track i. The track intercepts the border of a mesh at two places. We denote the flux $\Phi_{g,in}(\mathbf{\Omega}_i)$ at the first place where the track intercepts and $\Phi_{g,out}(\mathbf{\Omega}_i)$ at the second place. By integrating equation (4.1) along the track inside the mesh we get

$$\Phi_{g,out}(\mathbf{\Omega}_i) = \Phi_{g,in}(\mathbf{\Omega}_i)\mathrm{e}^{-\Sigma_g l} + \frac{S_g(1 - \mathrm{e}^{-\Sigma_g l})}{4\pi\Sigma_g}, \tag{4.2}$$

where we have assumed that the cross sections and the source term are spatially uniform and l is the length of the track within the mesh.

Using these approximations equation (4.1) can be approximated by

$$\frac{\Phi_{g,in}(\mathbf{\Omega}_i) - \Phi_{g,out}(\mathbf{\Omega}_i)}{l} + \Sigma_g \Phi_{g,avg}(\mathbf{\Omega}_i) = \frac{S_g}{4\pi}. \tag{4.3}$$

By rewriting and setting that $\Phi_{g,in}(\mathbf{\Omega}_i) - \Phi_{g,out}(\mathbf{\Omega}_i) = D$ we get

$$\Phi_{g,avg}(\mathbf{\Omega}_i) = \frac{S_g}{4\pi\Sigma_g} + \frac{D}{l\Sigma_g}. \tag{4.4}$$

Then the average angular flux within a mesh becomes

$$\Phi_{g,mesh}(\mathbf{\Omega}_i) = \sum_{j \in mesh} \frac{\Phi_{g,avg,j}(\mathbf{\Omega}_i) V_{i,j}}{V}, \tag{4.5}$$

where we sum the flux for all j tracks within the mesh in the direction $\mathbf{\Omega}_i$. V is the mesh volume and the volume $V_{i,j}$ is defined by $V_{i,j} = a \cdot l \cdot p$, where p is the spacing between the tracks in direction $\mathbf{\Omega}_i$, and a is a normalization factor to make

$$\sum_{j \in mesh} V_{i,j} = V. \tag{4.6}$$

The scalar flux can be calculated by integrating the mesh average angular flux over the whole solid angle. And by using a standard method of source iterations [19] we get the solution of the transport equation for the whole system.

This characteristics method is also used by VCOEF to calculate the VNEM expansion functions.

4.3.2 Boundary conditions

The following boundary conditions can be given at each of four outer interfaces of the core:

1. Reflective boundary condition:

$$\Phi_g(\mathbf{r}, \mathbf{\Omega}) = \Phi_g(\mathbf{r}, \mathbf{\Omega}'), \tag{4.7}$$

   where $\mathbf{\Omega}'$ is the mirror reflected direction of $\mathbf{\Omega}$ at the position $\mathbf{r}$.

2. No incoming neutrons boundary condition:

$$\Phi_g(\mathbf{r}, \mathbf{\Omega}) = 0, \tag{4.8}$$

   if $\mathbf{\Omega}$ is in incoming direction to the core.

3. Zero even parity component boundary condition:

$$\Phi_g(\mathbf{r}, \mathbf{\Omega}) + \Phi_g(\mathbf{r}, -\mathbf{\Omega}) = 0, \tag{4.9}$$

   which is an extended zero flux boundary condition.

FCM2D program flow

- Read input data (core geometry, composition, cross sections, track specification, iteration control, etc.).
- Generate characteristics/tracks.
- Set initial guess for scalar fluxes, and angular fluxes at in/outlet of tracks.
- Initialize old fission source by space mesh.
- By energy group and for each source iteration:
  - Calculate group neutron source by space mesh.
  - Inner iteration: Integrate transport equation along tracks, and iterate until angular fluxes at in/outlet converge.
  - Calculate scalar flux by mesh from angular fluxes.
  - Calculate new fission term.
  - Calculate k_{eff} from Rayleigh quotient of new to old fission source terms.
  - Calculate convergence error from the error-ratio of new to old fission sources.
- Calculate nodal/pin powers if source iteration has converged.

4.4 VCOEF2D

The neutron flux is expanded into spherical harmonics;

$$\Phi_g(\mathbf{r},\mathbf{\Omega}) = \frac{1}{4\pi}\sum_{l=0}(2l+1)\sum_{-l}^{l} Y_{l,m}(\mathbf{\Omega})\phi_{l,m,g}(\mathbf{r}), \tag{4.10}$$

where $\phi_{l,m,g}(\mathbf{r})$ is the flux moment.

In the three-dimensional version of the VNEM method one assembly is divided into several vertical regions, where each region is called a node. In the two dimensional version which we are using here, a node and an assembly is the same.

We approximate the flux moment within a node by doing an expansion;

$$\begin{aligned}\phi_{l,m,g}(\mathbf{r}) \equiv & \sum_{nr} F^{Q}_{nr,g}\varphi^{Q}_{nr,l,m,g}(\mathbf{r}) \\ & + \sum_{pp,il,im,sr} F^{CB}_{pp,il,im,sr,g}\varphi^{CB}_{pp,il,im,sr,l,m,g}(\mathbf{r}) \\ & + \sum_{pp,il,im,sr} F^{SB}_{pp,il,im,sr,g}\varphi^{SB}_{pp,il,im,sr,l,m,g}(\mathbf{r}),\end{aligned} \tag{4.11}$$

where l, m, il and im is even, and;

nr = degree of 2-dimensional Legendre polynomial

pp = index of interface of the two dimensional node

il = l-index of the boundary value

im = m-index of the boundary value

sr = degree of 1-dimensional Legendre polynomial

$F^{Q}_{nr,g}$ is the source expansion coefficient, and $F^{CB}_{pp,il,im,sr,g}$ and $F^{SB}_{pp,il,im,sr,g}$ are the boundary value expansion coefficients. $\varphi^{Q}_{nr,l,m,g}$ is the source expansion function, and $\varphi^{CB}_{pp,il,im,sr,l,m,g}$ and $\varphi^{SB}_{pp,il,im,sr,l,m,g}$ are the boundary expansion functions. To solve this numerically, the sum in the expansion has to be finite. Hence the expansion becomes an approximation. By using the variational principle (Ritz method) we can minimize the error.

4.4.1 Source expansion function

VCOEF2D uses the single assembly scalar flux calculated by FCM2D and the cell homogenized cross sections from HELIOS to generate the expansion functions and calculate the VNEM coefficients (see section 4.4.3 and 4.4.4 for the latter). The

source expansion function is obtained by solving the transport equation within an assembly with the infinite lattice source term multiplied with Legendre polynomials.

$$\mathbf{\Omega} \cdot \nabla \Phi_{nr,g}(\mathbf{r}, \mathbf{\Omega}) + \Sigma_g \Phi_{nr,g}(\mathbf{r}, \mathbf{\Omega}) = \frac{S_g^{\infty}(\mathbf{r}) P_{nr}(\mathbf{r})}{4\pi}, \tag{4.12}$$

where

$$S_g^{\infty}(\mathbf{r}) = \sum_{g'} \Sigma^s_{g' \to g}(\mathbf{r}) \Phi^{\infty}_{g'}(\mathbf{r}) + \frac{\chi_g(\mathbf{r})}{k_{eff}} \sum_{g'} \nu \Sigma^f_{g'}(\mathbf{r}) \Phi^{\infty}_{g'}(\mathbf{r}). \tag{4.13}$$

It is one of the fundamental assumptions in VNEM that the ratio S_g/S_g^{∞} can be expanded into Legendre polynomials. Φ_g^{∞} is single assembly scalar flux calculated in FCM2D. $P_{nr}(\mathbf{r})$ is the 2-dimensional Legendre polynomial of degree nr defined

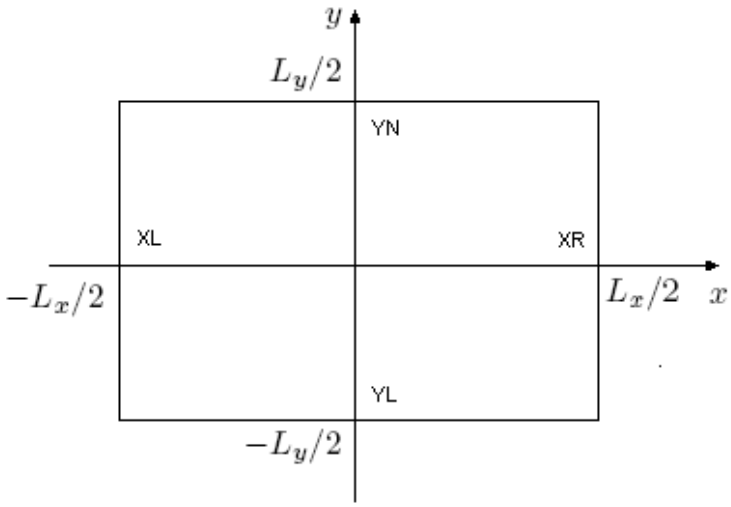

Figure 4.4: A node. In these calculations a node and an assembly is the same. In a 3-D system a node is a vertical region of the assembly.

within a node. From figure 4.4 we see that the one dimensional Legendre polynomials can be defined by

$$P_0(x) = 1, \tag{4.14}$$

$$P_1(x) = 2x/L_x, \tag{4.15}$$

and

$$P_2(x) = \frac{3}{2}\left(\frac{2x}{L_x}\right)^2 - \frac{1}{2}, \tag{4.16}$$

where the origin is taken at the center of the node and L_x is the width of the node. $P_{sr}(y)$ is defined in the same way, and the 2-D Legendre polynomials can be defined by;

$$P_0(\mathbf{r}) = P_0(x) P_0(y), \tag{4.17}$$

$$P_1(\mathbf{r}) = P_1(x)P_0(y), \tag{4.18}$$
$$P_2(\mathbf{r}) = P_0(x)P_1(y), \tag{4.19}$$
$$P_3(\mathbf{r}) = P_1(x)P_1(y), \tag{4.20}$$
$$P_4(\mathbf{r}) = P_2(x)P_0(y), \tag{4.21}$$

and

$$P_5(\mathbf{r}) = P_0(x)P_2(y). \tag{4.22}$$

The real boundary expansion function, $\Phi_{pp,il,im,g}$, for each of the four interfaces of a node;

pp = XL(west), XR(east), YL(south) and YR(north)

VCOEF2D solves equation (4.12) within an assembly by the method of characteristics described in section 4.3.1 with the zero even parity component boundary condition (4.9) at the assembly boundaries. Then the expansion functions are calculated by expanding $\Phi_{nr,g}(\mathbf{r}, \mathbf{\Omega})$ into spherical harmonics:

$$\varphi^{Q}_{nr,l,m,g}(\mathbf{r}) = \int_{4\pi} \mathrm{d}\mathbf{\Omega} \mathrm{Y}^{*}_{l,m}(\mathbf{\Omega})\Phi_{nr,g}(\mathbf{r}, \mathbf{\Omega}), \tag{4.23}$$

where $\mathrm{Y}^{*}_{l,m}(\mathbf{\Omega})$ is the complex conjugate of the spherical harmonics of order l, m. The zero even parity component boundary condition can now be expressed by

$$\varphi^{Q}_{nr,l,m,g}(\mathbf{r}) = 0 \tag{4.24}$$

at the assembly boundaries.

4.4.2 Boundary expansion function

The real boundary value expansion function is then found by solving the transport equation within an assembly with zero sources,

$$\mathbf{\Omega} \cdot \nabla \Phi_{nr,il,ir,sm,g}(\mathbf{r}, \mathbf{\Omega}) + \Sigma_g \Phi_{nr,il,im,sr,g}(\mathbf{r}, \mathbf{\Omega}) = 0, \tag{4.25}$$

by the method of characteristics with the boundary condition expressed in terms of the boundary value expansion function;

$$\varphi^{CB}_{pp,il,im,l,m,sr,g}(\mathbf{r}) = \delta_{pp,qq}\delta_{il,l}\delta_{im,m}P_{sr}(\mathbf{s}_{pp}) \tag{4.26}$$

on the node interfaces qq, where $P_{sr}(\mathbf{s}_{pp})$ is the one dimensional Legendre polynomial defined on node interface pp, and $\mathbf{s}_{pp}$ is the coordinate taken along node interface pp. And we find the real boundary value expansion function by expanding the solution of equation (4.25) into spherical harmonics:

$$\varphi^{CB}_{pp,il,im,l,m,sr,g}(\mathbf{r}) = \int_{4\pi} \mathrm{d}\mathbf{\Omega} \mathrm{Y}^{*}_{l,m}(\mathbf{\Omega})\Phi_{pp,il,im,g}(\mathbf{r}, \mathbf{\Omega}), \tag{4.27}$$

The imaginary boundary value expansion function, $\varphi^{SB}_{pp,il,im,l,m,sr,g}(\mathbf{r})$, is found in the same way but with the boundary condition

$$\varphi^{SB}_{pp,il,im,l,m,sr,g}(\mathbf{r}) = -\mathrm{i}\delta_{pp,qq}\delta_{il,l}\delta_{im,m}P_{sr}(\mathbf{s}_{pp}). \tag{4.28}$$

4.4.3 Source expansion coefficients

The source expansion coefficients, $F^Q_{nr,g}$, are determined by using the variational principle.
A functional $F[\phi]$ is defined:

$$F[\phi] = \sum_g \sum_{l=0}^{L} \sum_{m=0}^{l} F^c_{l,m,g}[\phi^c_{l,m,g}(\mathbf{r})] + \sum_g \sum_{l=2}^{L} \sum_{m=2}^{l} F^s_{l,m,g}[\phi^s_{l,m,g}(\mathbf{r})], \tag{4.29}$$

where $\phi_{l,m,g}$ is a complex number which can be written as

$$\phi_{l,m,g}(\mathbf{r}) = \phi^c_{l,m,g}(\mathbf{r}) - \mathrm{i}\phi^s_{l,m,g}(\mathbf{r}). \tag{4.30}$$

Now we define

$$\begin{aligned} F^i_{l,m,g}[\phi^i_{l,m,g}(\mathbf{r})] \equiv & \int_{core} \mathrm{d}\mathbf{r}\ aabb1_{l,m,g}(\mathbf{r}) \left[\left(\frac{\partial}{\partial x} \phi^i_{l,m,g}(\mathbf{r}) \right)^2 + \left(\frac{\partial}{\partial y} \phi^i_{l,m,g}(\mathbf{r}) \right)^2 \right] \\ + & \int_{core} \mathrm{d}\mathbf{r}\ aabb1_{l,m,g} \left[\Sigma_g(\mathbf{r}) \phi^i_{l,m,g}(\mathbf{r})^2 - 2Q^i_{l,m,g}(\mathbf{r}) \right], \end{aligned} \tag{4.31}$$

where $i = s$ or c indicating the real or imaginary part of $\phi_{l,m,g}(\mathbf{r})$

We also have that

$$aabb1_{l,m,g}(\mathbf{r}) = \frac{1}{\Sigma_g(\mathbf{r})} (\alpha_{l,m+1}\alpha_{l+1,m} + \beta_{l,m-1}\beta_{l+1,m} + \beta_{l,m+1}\beta_{l-1,m} + \alpha_{l,m-1}\alpha_{l-1,m}), \tag{4.32}$$

where

$$\alpha_{l,m} = \frac{[(l+m)(l+m+1)]^{\frac{1}{2}}}{2(2l+1)} \tag{4.33}$$

and

$$\beta_{l,m} = \alpha_{l,-m}. \tag{4.34}$$

The source expansion coefficients, $F^Q_{nr,g}$, are then determined by substituting equation (4.11) into equation (4.29), differentiate with respect to the source expansion coefficient and set the result to be zero. The variational principle guarantees that we get the most accurate solution by doing this. We get

$$\begin{aligned} & \sum_{nr'} NN_{nr,nr',g} F^Q_{nr',g} - \sum_{nr',g'} NNS_{nr,g,nr',g'} F^Q_{nr',g'} \\ = & \sum_{pp,ir,g'} \left[NCS_{pp,nr,g,ir,g'} F^{CB}_{pp,ir,g'} + NSS_{pp,nr,g,ir,g'} F^{SB}_{pp,ir,g'} \right] \\ + & \frac{1}{k_{eff}} \sum_{pp,ir,g'} \left[NCF_{pp,nr,g,ir,g'} F^{CB}_{pp,ir,g'} + NSF_{pp,nr,g,ir,g'} F^{SB}_{pp,ir,g'} \right] \\ + & \frac{1}{k_{eff}} \sum_{nr',g'} NNF_{nr,g,nr',g'} F^S_{nr',g'}, \end{aligned} \tag{4.35}$$

where index $ir = il, im, sr$, and this equation is eventually solved in VNEM2D for a multi-assembly system. The VNEM coefficients $NN_{nr,nr',g}$, $NNS_{nr,g,nr',g'}$, $NCS_{pp,nr,g,ir,g'}$, $NSS_{pp,nr,g,ir,g'}$, $NCF_{pp,nr,g,ir,g'}$, $NSF_{pp,nr,g,ir,g'}$ and $NNF_{nr,g,nr',g'}$ are calculated in VCOEF2D using the single assembly cell homogenized neutron scalar fluxes and cross sections.

4.4.4 Boundary value expansion coefficients

The boundary value expansion coefficients, $F^{CB}_{pp,ir,g}$ and $F^{SB}_{pp,ir,g}$, are determined from the continuity condition of the flux moment $\varphi_{l,m,g}(\mathbf{r})$ and the current moment $\mathbf{J}_{l,m,g}(\mathbf{r})$. $\varphi_{l,m,g}(\mathbf{r})$ and $\mathbf{n}\cdot\mathbf{J}_{l,m,g}(\mathbf{r})$ must be continuous at the node interface for even l, where $\mathbf{J}_{l,m,g}$ is the current moment and $\mathbf{n}$ is the unit vector normal to the node interface.

The continuity of $\phi_{l,m,g}(\mathbf{r})$ at the node interface can be satisfied by setting boundary value expansion coefficients $F^{CB}_{pp,ir,g}$ of one node and $F^{SB}_{pp',ir,g}$ of another node to be the same at their common interface $pp = pp'$.

To satisfy the continuity of $\mathbf{n}\cdot\mathbf{J}_{l,m,g}(\mathbf{r})$, we first write it by a linear combination of the gradient of the flux moments

$$\mathbf{n}\cdot\mathbf{J}_{l,m,g}(\mathbf{r}) = \sum_{l'=l-2}^{l+2}\sum_{m'=m-2}^{m+2}\alpha_{l',m',g}(\mathbf{r})\mathbf{n}\cdot\nabla\phi_{l',m',g}(\mathbf{r}), \tag{4.36}$$

where $\alpha_{l',m',g}(\mathbf{r})$ is a constant depending on the total cross section at $\mathbf{r}$. We have the continuity requirement

$$\begin{aligned}&\sum_{l'=l-2}^{l+2}\sum_{m'=m-2}^{m+2}\alpha_{l',m',g}(\mathbf{s}_{pp})\mathbf{n}\cdot\nabla\phi_{l',m',g}(\mathbf{s}_{pp})|_{\text{node 1}}\\ =&\sum_{l'=l-2}^{l+2}\sum_{m'=m-2}^{m+2}\alpha_{l',m',g}(\mathbf{s}_{pp'})\mathbf{n}\cdot\nabla\phi_{l',m',g}(\mathbf{s}_{pp'})|_{\text{node 2}}\end{aligned} \tag{4.37}$$

on their common interface. We have a finite number of coefficients in the moment expansion and it is therefore impossible to satisfy equation (4.37) at all points of the interface. Instead we request the continuity of

$$\begin{aligned}&\sum_{l'=l-2}^{l+2}\sum_{m'=m-2}^{m+2}\int_{pp}\mathrm{d}\mathbf{s}P_{sr}(\mathbf{s}_{pp})\alpha_{l',m',g}(\mathbf{s}_{pp})\mathbf{n}\cdot\nabla\phi_{l',m',g}(\mathbf{s}_{pp})|_{\text{node 1}}\\ =&\sum_{l'=l-2}^{l+2}\sum_{m'=m-2}^{m+2}\int_{pp'}\mathrm{d}\mathbf{s}P_{sr}(\mathbf{s}_{pp'})\alpha_{l',m',g}(\mathbf{s}_{pp'})\mathbf{n}\cdot\nabla\phi_{l',m',g}(\mathbf{s}_{pp'})|_{\text{node 2}}.\end{aligned} \tag{4.38}$$

By substituting equation (4.11) into equation (4.38) we get a set of linear equations

in x-direction on the form

$$\begin{aligned}
&\sum_{qq,ir'} CC_{pp,ir,qq,ir',g} F^{CB}_{qq,ir,g} + \sum_{qq,ir'} CS_{pp,ir,qq,ir',g} F^{SB}_{qq,ir,g}|_{\text{node 1}} \\
+ &\sum_{qq,ir'} CC_{pp',ir,qq,ir',g} F^{CB}_{qq,ir,g} + \sum_{qq,ir'} CS_{pp',ir,qq,ir',g} F^{SB}_{qq,ir,g}|_{\text{node 2}} \\
= &\sum_{nr} CN_{pp,ir,nr,g} F^{CB}_{qq,ir,g} F^{Q}_{nr,g}|_{\text{node 1}} \\
+ &\sum_{nr} CN_{pp',ir,nr,g} F^{CB}_{qq,ir,g} F^{Q}_{nr,g}|_{\text{node 2}},
\end{aligned} \tag{4.39}$$

for the real part, and

$$\begin{aligned}
&\sum_{qq,ir'} SC_{pp,ir,qq,ir',g} F^{CB}_{qq,ir,g} + \sum_{qq,ir'} SS_{pp,ir,qq,ir',g} F^{SB}_{qq,ir,g}|_{\text{node 1}} \\
+ &\sum_{qq,ir'} SC_{pp',ir,qq,ir',g} F^{CB}_{qq,ir,g} + \sum_{qq,ir'} SS_{pp',ir,qq,ir',g} F^{SB}_{qq,ir,g}|_{\text{node 2}} \\
= &\sum_{nr} SN_{pp,ir,nr,g} F^{CB}_{qq,ir,g} F^{Q}_{nr,g}|_{\text{node 1}} \\
+ &\sum_{nr} SN_{pp',ir,nr,g} F^{CB}_{qq,ir,g} F^{Q}_{nr,g}|_{\text{node 2}}
\end{aligned} \tag{4.40}$$

for the imaginary part. We also get similar equations in y-direction. These equations together with equation (4.35) are solved in VNEM2D for a multi-assembly system. The VNEM coefficients $CC_{pp,ir,qq,ir',g}$, $CS_{pp,ir,qq,ir',g}$, $CN_{pp,ir,nr,g}$, $SC_{pp,ir,qq,ir',g}$, $SS_{pp',ir,qq,ir',g}$, $SN_{pp',ir,nr,g}$ and similar coefficients in y-direction are also calculated in VCOEF2D using the single assembly cell homogenized scalar fluxes and cross sections.

4.5 VNEM2D

Using the VNEM coefficients and expansion functions for each burnup step obtained from VCOEF2D, the VNEM2D code solves equations defined in the previous section for the expansion coefficients $F^{Q}_{nr,g}$, $F^{CB}_{pp,il,im,sr,g}$ and $F^{SB}_{pp,il,im,sr,g}$ for a rectangular multi-assembly system.

VNEM2D does this by a normal method of source iterations with Gauss-Seidel inner iterations and finds the effective multiplication factor (k_{eff}). The expansion coefficients are used to calculate the neutron scalar flux and the fission power density for a multi-assembly system.

There are two options for the boundary condition on the outer edges of the system; Either the zero even parity component boundary condition, or reflective boundary condition.

The dominating term in the scalar flux is $\phi_{0,0,g}$, so the fission power density $\mathrm{P}(\mathbf{r})$ becomes

$$\begin{aligned}
\mathrm{P}(\mathbf{r}) &= \sum_{g} \Sigma_{f,g}(\mathbf{r})\phi_{g}(\mathbf{r}) \\
&= \sum_{nr,g} F^{Q}_{nr,g}\Sigma_{f,g}(\mathbf{r})\varphi^{Q}_{nr,0,0,g}(\mathbf{r}) \\
&+ \sum_{pp,il,im,sr,g} F^{CB}_{pp,il,im,sr,g}\Sigma_{f,g}(\mathbf{r})\varphi^{CB}_{pp,il,im,sr,0,0,g}(\mathbf{r}) \\
&+ \sum_{pp,il,im,sr,g} F^{SB}_{pp,il,im,sr,g}\Sigma_{f,g}(\mathbf{r})\varphi^{SB}_{pp,il,im,sr,0,0,g}(\mathbf{r}),
\end{aligned} \tag{4.41}$$

where we have used the expansion defined in equation (4.11). We define an operator for spatial averaging:

$$\langle f(\mathbf{r})\rangle_{\mathrm{nreg}} \equiv \frac{\int_{\mathrm{nreg}} f(\mathbf{r})\mathrm{d}x\mathrm{d}y}{\int_{\mathrm{nreg}} \mathrm{d}x\mathrm{d}y}. \tag{4.42}$$

The pin powers and the assembly(nodal) powers can then be calculated by;

$$\begin{aligned}
P_{\mathrm{nreg}} &= \sum_{nr,g} F^{Q}_{nr,g}\left\langle \Sigma_{f,g}(\mathbf{r})\varphi^{Q}_{nr,0,0,g}(\mathbf{r})\right\rangle \\
&+ \sum_{pp,il,im,sr,g} F^{CB}_{pp,il,im,sr,g}\left\langle \Sigma_{f,g}(\mathbf{r})\varphi^{CB}_{pp,il,im,sr,0,0,g}(\mathbf{r})\right\rangle \\
&+ \sum_{pp,il,im,sr,g} F^{SB}_{pp,il,im,sr,g}\left\langle \Sigma_{f,g}(\mathbf{r})\varphi^{SB}_{pp,il,im,sr,0,0,g}(\mathbf{r})\right\rangle .
\end{aligned} \tag{4.43}$$

For nreg = fuel cell, and nreg = whole assembly. The scalar flux can be obtained in the same way by excluding the cross sections from equation (4.41) and (4.43).

VNEM program flow

- Read input data (core geometry, iteration control, etc.).
- Read VNEM coefficients generated by VCOEF2D code.
- Set initial guess for source expansion coefficient values.
- Set initial guess for boundary expansion coefficient values.
- Initialize old fission source term.
- By energy group and for each source iteration:
  - Calculate source expansion coefficients.
  - Gauss-Seidel inner iteration [22]: Calculate boundary expansion coefficients.
  - Calculate new fission source term.
  - Calculate k_{eff} from Rayleigh quotient of the new to old source terms.
  - Calculate convergence error from the error-ratio of new to old fission source terms.
- Calculate nodal/pin powers if source iteration has converged.

4.6 Summary

The 2D geometrical model of the core consists of a number of assemblies, and these assemblies are divided into space meshes. A reasonable number of space meshes are used to do the calculations in HELIOS, where single assembly burnup calculations are done with reflective boundary condition. We start out with fresh fuel and then the calculations are done in steps with increasing burnup.

The cross sections for each burnup step from the single assembly HELIOS burnup calculations are homogenized for each fuel cell and stored in energy groups. Using these cross sections FCM2D recalculates the neutron scalar flux for the cell homogenized assembly using a characteristics method.

In VCOEF2D the g-th group neutron flux from FCM2D is expanded into a sum, and the source expansion coefficients are determined by the variational principle. The boundary value expansion coefficients are determined by the continuity condition of the neutron angular current moments at the radial node interface. The flux continuity condition is satisfied by using the same boundary value coefficients at the common interface of radially neighboring nodes. To determine the boundary value coefficients, we request the continuity of the spatial moments in x-direction over a surface perpendicular to the x-axis of a node, and similarly in the y-direction.

For each burnup step VNEM2D uses the VNEM coefficients to calculate the expansion coefficients $F^{Q}_{nr,g}$, $F^{CB}_{pp,il,im,sr,g}$ and $F^{SB}_{pp,il,im,sr,g}$ and the effective multiplication factor k_{eff} for a multi-assembly system. The scalar flux and power level are calculated using the expansion coefficients.

It is obvious that the burnup calculations done in HELIOS are approximate for a multi-assembly system. The intra assembly neutron flux for a heterogeneous multi-assembly whole core system will be different from the single assembly flux. The burnup will therefore also be different for a whole core system. Calculations are done to investigate this approximation, and these calculations are explained in the next chapter.

Chapter 5

Calculations

The cross sections in the different assemblies in the core are not identical. Therefore the actual flux distribution is tilted from that obtained by the infinite lattice assumption and therefore the burnup is tilted. We would like to apply the actual boundary condition at the assembly boundary in the single assembly calculations. However, it is impossible to know the actual boundary condition before solving the whole core problem. The burnup tilt is investigated by comparing VNEM multi-assembly calculations with HELIOS reference calculations.

5.1 Burnup calculations

The following calculations are only done for a small part of the core since full core calculations in HELIOS may take a very long time.

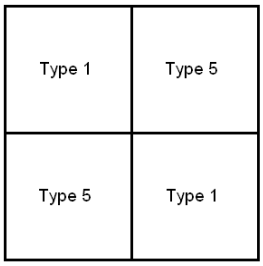

Figure 5.1: A small multi-assembly system from the central part of the core. Type 1 is uranium enriched to 2.11% uranium-235 and type 5 is uranium enriched to 2.60% uranium-235 and with 20 rods with burnable poisons.

We do calculations for a small part close to the center of the core. The system can be seen in figure 5.1. We look at a two-dimensional system in the radial direction of the core since the tilt mainly occurs in the radial direction. Assembly type 1 and type 5 have different enrichment of uranium-235 and assembly type 5 do also have twenty rods with burnable poisons. This resembles a part of the Ringhals core where

the difference between the assemblies are large, and this is therefore one of the parts of the core where the burnup tilt is most likely to be seen.

5.1.1 Single assembly calculations

Single assembly burnup calculations are done in HELIOS for both of the assemblies with reflective boundary condition in order to calculate the input for FCM2D and the VNEM calculations. The burnup calculations are done in steps of 2 GWd/T from zero to 30 GWd/T with zero, 400, 800 and 1400 ppm boron concentration in the moderator water. The temperature of the moderator water is set to be 557.06 Kelvin and the density of the water is set to be 0.7576 g/cm^3 corresponding to this temperature and a pressure of 15.7 MPa. For each burnup step, branch-off calculations are done with 400 ppm boron at 4% of full power with equilibrium Xenon, because the prensent version of VNEM2D does not include thermal hydraulic- and Xenon effects.

The branch-off calculations are done in this way:

We use the cross sections of the burnt fuel for a certain burnup step, change the boron concentration to 400 ppm boron in the moderator water and change the power level to 4% of full power. Then we calculate the properties of this system.

We obtain energy-groupwise cross sections from the branch off calculations for each burnup step, where the neutron energy is condensed into five groups in the following way:

Group	Upper boundary		Lower boundary
1	∞	-	1 MeV
2	1 MeV	-	9.119 keV
3	9.119 keV	-	3.9279 eV
4	3.9279 eV	-	0.27052 eV
5	0.27052 eV	-	0 eV

These cross sections are homogenized for each fuel cell and saved in a text file. The input code for these calculations are in appendix A. A program reads this text file and saves the cross sections in an input format for FCM2D. FCM2D calculates the angular flux for VCOEF2D, and VCOEF2D generates the VNEM coefficients and expansion functions for VNEM2D. See chapter 4 for details on these calculations.

5.1.2 Multi-assembly calculations

VNEM uses the VNEM coefficients and the expansion functions for the two different assemblies to calculate the effective multiplication factor and the power level of the multi-assembly system. We use reflective boundary condition at the outer boundaries of the system. The input data for these calculations can be found in appendix B.

HELIOS is also used to do a reference multi-assembly burnup calculation of the system in figure 5.1 with the same boundary condition at the outer boundaries of the system, and even a small system as this requires a very long computing time in HELIOS. The input for these calculations are in section A.3 in appendix A. These multi-assembly calculations are done to have a reference to compare with the results from VNEM2D, where we choose to compare the effective multiplication factor, k_{eff}, of the system and the relative difference between the maximum pin power in the assemblies. The results of these comparisons are presented in the next chapter.

Chapter 6

Results

6.1 The effective multiplication factor

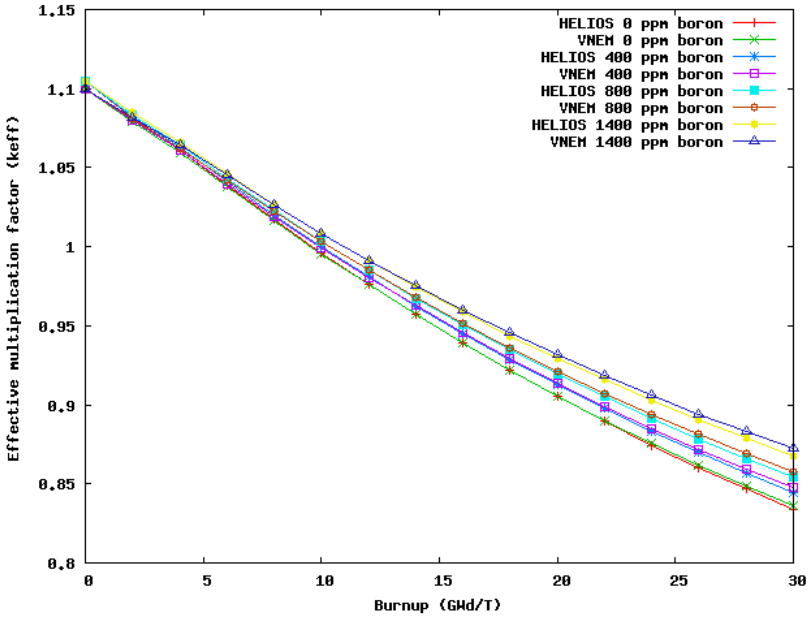

Figure 6.1: Effective multiplication factor (k_{eff}) for the system in figure 5.1, where the fuel are burnt from zero to 30 GWd/T. There is a small difference depending on the boron concentration the fuel has been burnt with, and the difference between HELIOS and VNEM is very small.

In figure 6.1 we see that the difference in k_{eff} is small between the HELIOS reference calculations and the VNEM calculations, but there is a difference between the HELIOS and VNEM calculations even for zero burnup. The precision of the VNEM method is well documented for zero burnup [1][23], and we are interested in how the precision changes for burnt fuel. We do therefore subtract the initial

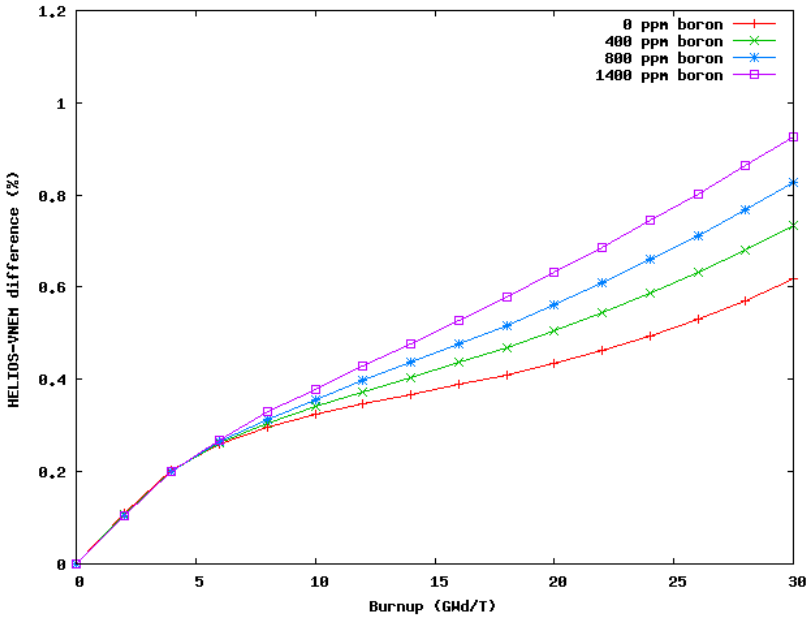

Figure 6.2: Here we see how big the difference in calculated effective multiplication factor is between HELIOS and VNEM. The difference increases fast in the start and it is dependent on the burnup history for higher burnup.

difference from the results to get a better picture of the change of the difference between the HELIOS and VNEM burnup calculations. Figure 6.2 shows how the HELIOS-VNEM difference accumulates from zero to 30 GWd/T. There is a difference in how the difference accumulates dependent on the burnup history of the fuel, where the different burnup histories are represented by fuel burnt with different boron concentrations.

We choose to look at how big the change of the HELIOS-VNEM difference is from one burnup step to the next one to get an even better view of the HELIOS-VNEM difference. This is the derivative of the graph in figure 6.2, and figure 6.3 shows a plot of this change. From this graph we see that the biggest change in the difference occurs from zero to six GWd/T, and the change in this interval is approximately the same for fuel that are burnt with different boron concentrations. The change of the difference is lower and more constant from ten to thirty GWd/T, and in this range there is a bigger difference between fuels with different burnup histories. When looking at the difference between zero ppm boron and 1400 ppm boron burnup history we see that there is nearly no difference from zero to four GWd/T. Figure 6.4 shows a plot of this difference, and we can see that it increases steadily from six to thirty GWd/T.

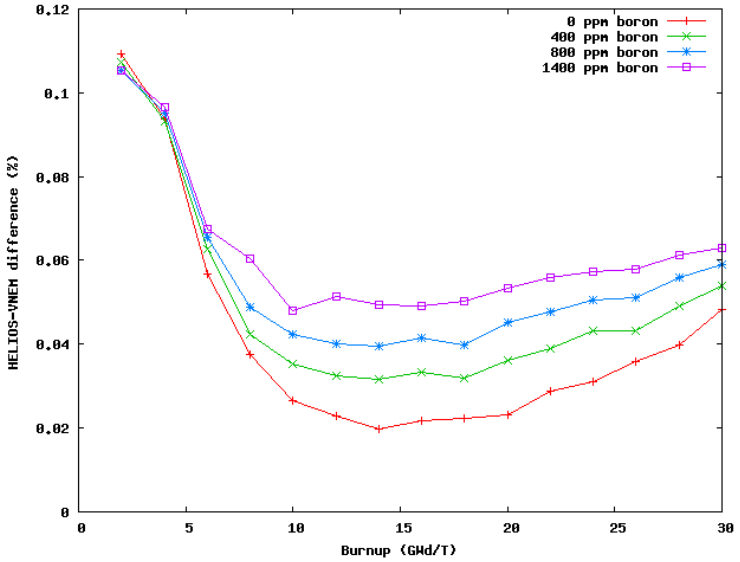

Figure 6.3: This graph is the derivative of the graph in figure 6.2. We see better how the change of difference is for individual burnup histories.

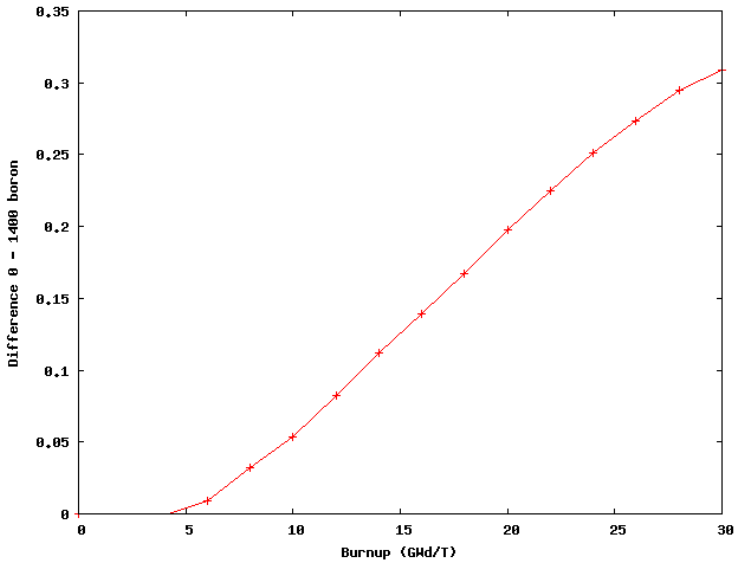

Figure 6.4: This is a plot of the difference between zero boron and 1400 ppm boron in figure 6.2. There is no difference from zero to four GWd/T, and a steady increase in the difference from six to thirty GWd/T.

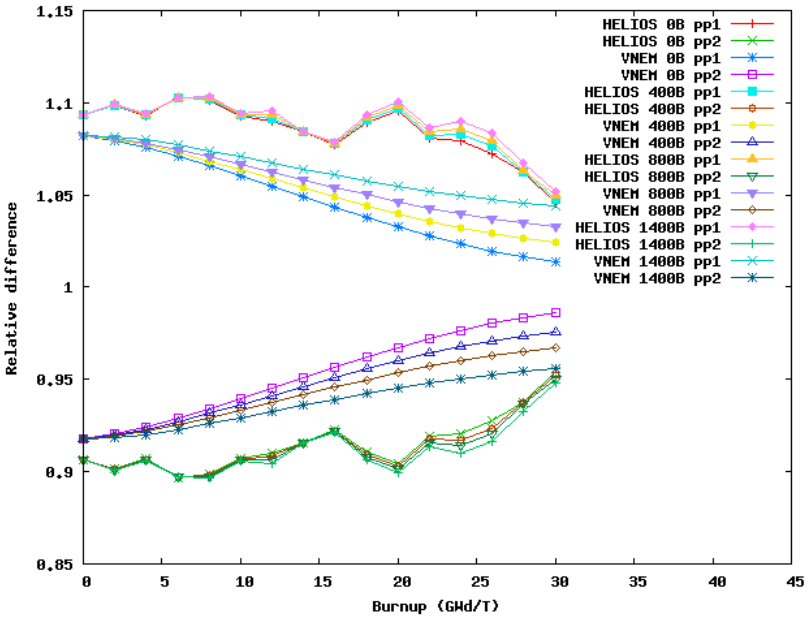

Figure 6.5: Relative difference between the maximum pin-powers in the two different assembly types. We see that VNEM is un-precise in calculating the pin-powers for burnup. Here pp1 and pp2 means relative max pin power in assembly type 1 and assembly type 5.

6.2 Pin powers

Before comparing the pin powers we normalize the power of the whole system in both the HELIOS and the VNEM calculations we normalize so that the average pin power of the system is one. We choose to look at the pin in each assembly with the highest power, and we compare the relative difference between these two pins. In figure 6.5 we see this relative difference obtained in the HELIOS and VNEM calculations.

We are most interested in how big the difference is between the HELIOS and the VNEM calculations, and figure 6.6 shows how the HELIOS-VNEM difference is at different burnup levels. We see that the difference is much bigger for the pin powers than for the effective multiplication factor. This is expected since the pin powers are local properties of the assemblies and they are therefore harder to calculate. When looking at the difference between zero and 1400 ppm boron burnup history (see figure 6.7) we see that the trend is that the difference increases with increasing burnup.

The inaccuracy in the calculations of the effective multiplication factor and the pin powers are significant, so the precision of the burnup calculations has to be improved. A way of improving the accuracy of the program is outlined in the next chapter.

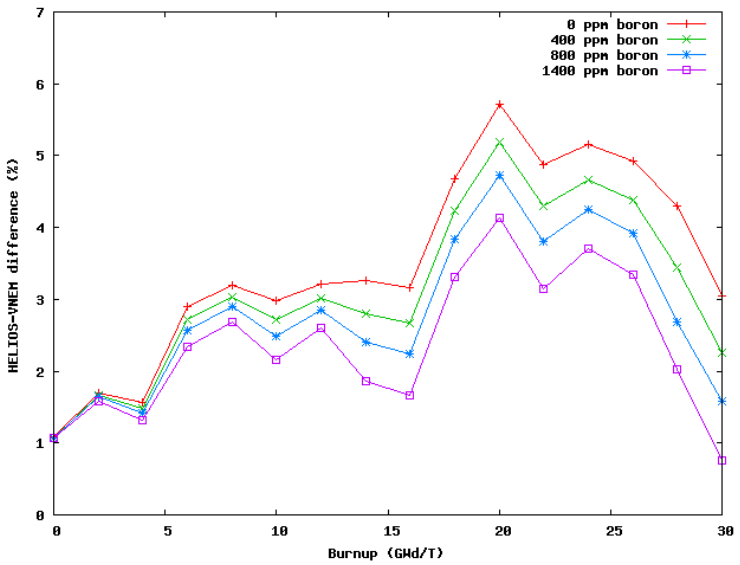

Figure 6.6: This is a plot of the difference between the VNEM and the HELIOS calculations of the pin powers. We see that the difference is large for higher burnup.

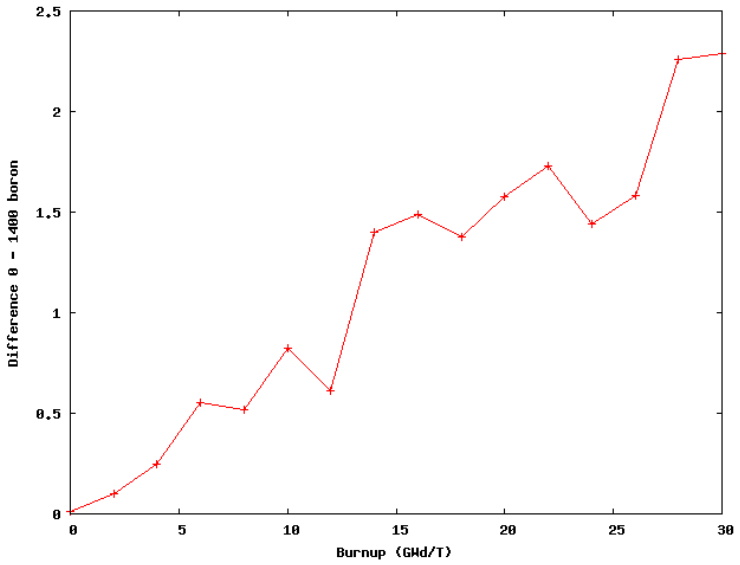

Figure 6.7: This plot shows the difference between zero boron and 1400 ppm boron in figure 6.6. The trend is that the difference increases for increasing burnup.

Chapter 7

Improvement of VNEM

VNEM is a bit inaccurate for burnup calculations and needs to be improved. The two stages method is a suggested method to improve the accuracy of VNEM for burnup calculations. In this method cross sections from single assembly burnup calculations are interpolated to resemble the actual burnup of the multi-assembly system.

7.1 Two stages method

Lattice phase (2D)

First an assembly cross sectional area is burnt by assuming 2D infinite-lattice model, where the whole life of an assembly is divided into a number of sufficiently short periods, and the cross sections are saved for each period.

Whole core phase (3D)

For the initial state the transport equation is solved for the whole core using the cross sections obtained in the 2D infinite-lattice model.

From this solution the power density can be found. By assuming that the power distribution is constant between the time steps, we can calculate the burnup of the core until the next time step. We expand the actual burnup into Legendre polynomials and use this to do a Taylor series expansion of the VNEM coefficients.

To find the derivative in the Taylor expansion we add a petubation to the burnup obtaioned in the lattice phase. We assume that cross sections of cells at this perturbed burnup can be approximated by interpolating those for un-tilted burnup. The interpolated cross sections are transferred to FCM2D and VCOEF yielding VNEM coefficients for the perturbed burnup. These VNEM coefficients can be used to calculate the derivative in the Taylor expansion.

Now we can solve the transport equation at the next time step, and this can be repeated until the end of the cycle.

7.2 Correction of VNEM coefficients

We are interested in finding a tilt corrected VNEM coefficient and we have to start by looking at the burnup. The group-wise power density at burnup step n' is

$$P_{n',g}(\mathbf{r}) = \Sigma_{f,n',g}(\mathbf{r})\phi_{n',g}(\mathbf{r}), \tag{7.1}$$

where $\Sigma_{f,n',g}$ is the macroscopic fission cross section and $\phi_{n',g}(\mathbf{r})$ is the scalar flux for energy group g at burnup step n'. The group-wise burnup $E_{n,g}(\mathbf{r})$ at burnup step n can be calculated from the group-wise power density.

$$E_{n,g}(\mathbf{r}) = \sum_{n'=1}^{n} \frac{P_{n',g}(\mathbf{r})\Delta t_{n'}}{w(\mathbf{r})} = \sum_{n'=1}^{n} \frac{\Sigma_{f,n',g}(\mathbf{r})\phi_{n',g}(\mathbf{r})\Delta t_{n'}}{w(\mathbf{r})}, \tag{7.2}$$

where $w(\mathbf{r})$ is the initial smeared density of the heavy elements and $\Delta t_{n'}$ is the time step between burnup step $n'-1$ and n'.

The burnup tilt is defined by

$$\Delta E_{g,n}(\mathbf{r}) = E_{g,n}(\mathbf{r}) - E^{\infty}_{g,n}(\mathbf{r}). \tag{7.3}$$

The tilt is expanded as

$$\Delta E_{g,n}(\mathbf{r}) = \sum_{k,l}^{K,L} \frac{\alpha_{k,l,n} P_k(g) P_l(\mathbf{r})}{w(\mathbf{r})}, \tag{7.4}$$

where $P_l(\mathbf{r})$ is the two dimensional Legendre polynomials defined within an assembly (see section 4.4.1) and $P_k(g)$ is the Legendre polynomials defined for the energy groups.

$$P_0(g) = 1, \tag{7.5}$$

$$P_1(g) = 2\frac{[g - 1 - (N_g - 1)/2]}{N_g - 1} \tag{7.6}$$

and

$$P_2(g) = 6\left[\frac{[g - (N_g + 1)/2]}{N_g - 1}\right]^2 - \frac{1}{2} + a(N_g), \tag{7.7}$$

where $a(N_g)$ is a correction factor to make $P_2(g)$ orthogonal to $P_0(g)$ since the number of energy groups, N_g, is so low that we cannot expect orthogonality. The correction factor can be calculated by

$$\sum_{g} P_0(g)P_2(g) = 0. \tag{7.8}$$

And $\alpha_{k,l,n}$ can be calculated by

$$\begin{aligned}
&\sum_g \int_{assembly} \Delta E_{g,n}(\mathbf{r}) P_k(g) P_l(\mathbf{r}) w(\mathbf{r}) \, \mathrm{d}\mathbf{r} \\
&= \sum_g \int_{assembly} P_k(g) P_l(\mathbf{r}) \sum_{k',l'} \alpha_{k',l',n} P_{k'}(g) P_{l'}(\mathbf{r}) \, \mathrm{d}\mathbf{r} \\
&= \sum_{k',l'} \alpha_{k',l',n} \sum_g P_k(g) P_{k'}(g) \int_{assembly} P_l(\mathbf{r}) P_{l'}(\mathbf{r}) \, \mathrm{d}\mathbf{r} \\
&= \alpha_{k,l,n} P_k^2(g) \int_{assembly} P_l^2(\mathbf{r}) \, \mathrm{d}\mathbf{r}. \qquad (7.9)
\end{aligned}$$

For $k = l = 0$ we get

$$\begin{aligned}
\alpha_{0,0,n} &= \Delta E_{g,n}(\mathbf{r}) w(\mathbf{r}) \, \mathrm{d}\mathbf{r} \\
&= \sum_g \int_{assembly} [E_{g,n}(\mathbf{r}) - E_{g,n}^{\infty}(\mathbf{r})] w(\mathbf{r}) \, \mathrm{d}\mathbf{r} \\
&= \sum_g \int_{assembly} E_{g,n}(\mathbf{r}) w(\mathbf{r}) \, \mathrm{d}\mathbf{r} - \sum_g \int_{assembly} E_{g,n}^{\infty}(\mathbf{r})] w(\mathbf{r}) \, \mathrm{d}\mathbf{r} \\
&= \sum_g E_{g,n}^{ass}(\mathbf{r}) w^{ass}(\mathbf{r}) - \sum_g E_{g,n}^{\infty,ass}(\mathbf{r}) w^{ass}(\mathbf{r}) \\
&= E_n^{ass}(\mathbf{r}) w^{ass}(\mathbf{r}) - E_n^{\infty,ass}(\mathbf{r}) w^{ass}(\mathbf{r}) = 0, \qquad (7.10)
\end{aligned}$$

because we require the the assembly average burnup of HELIOS and VNEM to be the same. We have also defined

$$E_n^{ass}(\mathbf{r}) = \sum_g E_{g,n}^{ass}(\mathbf{r}), \qquad (7.11)$$

$$E_{g,n}^{ass}(\mathbf{r}) = \int_{assembly} E_{g,n}(\mathbf{r}) w(\mathbf{r}) \, \mathrm{d}\mathbf{r} \qquad (7.12)$$

and

$$w^{ass}(\mathbf{r}) = \int_{assembly} w(\mathbf{r}) \, \mathrm{d}\mathbf{r}. \qquad (7.13)$$

The VNEM coefficients are functions of the burnup and we do therefore have

$$\begin{aligned}
C_n(E_{g,n}(\mathbf{r})) &= C_n(E_{g,n}^{\infty}(\mathbf{r}) + \Delta E_{g,n}(\mathbf{r})) \\
&= C_n(E_{g,n}^{\infty}(\mathbf{r}) + \sum_{k,l}^{K,L} \frac{\alpha_{k,l,n} P_k(g) P_l(\mathbf{r})}{w(\mathbf{r})})
\end{aligned} \qquad (7.14)$$

By doing a Taylor series expansion at $\alpha_{k,l,n} = 0$ we have

$$C_n(E_{g,n}(\mathbf{r})) = C_n(E^{\infty}_{g,n}(\mathbf{r}) + \sum_{k,l}^{K,L} \alpha_{k,l,n} \left[\frac{\partial C_n}{\partial \alpha_{k,l,n}} \right], \tag{7.15}$$

where $C_n(E^{\infty}_{g,n}(\mathbf{r})$ is the VNEM coefficient found by the infinite lattice approximation and $C_n(E_{g,n}(\mathbf{r}))$ is the correct VNEM coefficient. We see that we need to find $\partial C_n/\partial \alpha_{k,l,n}$ to be able to calculate $C_n(E_{g,n}(\mathbf{r}))$.

We start out by looking at the infinite lattice, group-wise burnup at time step n for fuel cell fc, $E^{\infty}_{g,n,fc}$, defined by

$$E^{\infty}_{g,n,fc} = \frac{\int_{fc} w(r) E^{\infty}_{g,n}(r) dr}{\int_{fc} w(r) dr}. \tag{7.16}$$

Now we add a perturbation element of order(k,l) and define $E_{g,n,fc,k,l}$ to be

$$E_{g,n,fc,k,l} = E^{\infty}_{g,n,fc} + \frac{a_{k,l,n} P_k(g) P_{l,fc}}{w_{fc}}, \tag{7.17}$$

where

$$w_{fc} = \frac{\int_{fc} w(r) dr}{\int_{fc} dr}, \tag{7.18}$$

and

$$P_{l,fc} = \frac{\int_{fc} P_l dr}{\int_{fc} dr}. \tag{7.19}$$

The cross sections corresponding to the pertubed burnup $E_{g,n,fc,k,l}$ can be calculated by an interpolation:

$$\Sigma_{g,n,fc,k,l} = W_1 \Sigma^{\infty}_{g,n1,fc} + W_2 \Sigma^{\infty}_{g,n2,fc}, \tag{7.20}$$

where $\Sigma^{\infty}_{g,n,fc}$ is the cross section obtained from the HELIOS infinite lattice calculations at burnup step n. The indicies $n1$ and $n2$ are the time steps chosen so that

$$E^{\infty}_{g,n1,fc} \leq E_{g,n,fc,k,l} \leq E^{\infty}_{g,n2,fc}, \tag{7.21}$$

so that $E^{\infty}_{g,n1,fc}$ and $E^{\infty}_{g,n2,fc}$ are the nearest burnup from the infinite lattice calculations. The weights are

$$W_1 = \frac{E^{\infty}_{g,n2,fc} - E_{g,n,fc,k,l}}{E^{\infty}_{g,n2,fc} - E^{\infty}_{g,n1,fc}} \tag{7.22}$$

and

$$W_2 = \frac{E_{g,n,fc,k,l} - E^{\infty}_{g,n1,fc}}{E^{\infty}_{g,n2,fc} - E^{\infty}_{g,n1,fc}}. \tag{7.23}$$

Using the perturbed cross sections in FCM2D and VCOEF2D calculations we can find the perturbed VNEM coefficients. Let the coefficient $C_m(E_{g,n,fc,k,l})$ be a coefficient at burnup step m corresponding to burnup $E_{g,n,fc,k,l}$. From equation (7.17) we then have

$$C_n(E_{g,n,fc,k,l}) = C_n(E^{\infty}_{g,n,fc} + \frac{a_{k,l,n}P_k(g)P_{l,fc}}{w_{fc}}). \tag{7.24}$$

By doing a Taylor expansion to the first order we get

$$C_n(E_{g,n,fc,k,l}) = C_n(E^{\infty}_{g,n,fc}) + a_{k,l,n}\left[\frac{\partial C_n}{\partial \alpha_{k,l,n}}\right], \tag{7.25}$$

and hence

$$\frac{\partial C_n}{\partial \alpha_{k,l,n}} = \frac{1}{a_{k,l,n}}\left[C_n(E_{g,n,fc,k,l}) - C_n(E^{\infty}_{g,n,fc})\right]. \tag{7.26}$$

We can now calculate the tilt corrected coefficient by inserting this into equation (7.15);

$$C_n = C^{\infty}_n + \sum_{k,l}^{K,L} \alpha_{k,l,n} \cdot \frac{1}{a_{k,l,n}}\left[C_n(E_{g,n,fc,k,l}) - C_n(E^{\infty}_{g,n,fc})\right], \tag{7.27}$$

where $a_{k,l,n}$, K and L has to be determined empirically.

This model has to be implemented into the system in a way so that it does not require too much computing resources.

Chapter 8

Conclusion

The VNEM method used to do burnup calculations in a two-dimensional assembly system. This method gives very good results for a system in its initial state, but the method does not give as good results for higher burnup.

The difference of the effective multiplication factor between the VNEM method and the reference increases steadily with increasing burnup of the fuel, and VNEM is more than 0.5 % away from the reference for all burnup histories at 30 GWd/T. The relative difference of the maximum pin powers is nearly 6 % away from the reference in the worst case. The magnitude of the difference between VNEM and the reference is also clearly dependent on the burnup history.

A method to improve the precision for burnup is outlined and has to be implemented in the VNEM method codes. Results obtained using this method should be compared to the results obtained in this study.

This method should also be tested on the outer parts of a reactor core. The burnup is also expected to be tilted in this part of the core since the assemblies here are only partly surrounded by other assemblies.

Bibliography

[1] M. Tsuiki and W. H. Beere. Comparision of VNEM to Measured Data from Ringhals Unit 3. *AR-VNEM-RH-01-08(r1)*, 12 2008.

[2] Pressurized Water Reactor. http://www.nrc.gov/reading-rm/basic-ref/students/reactors.html.

[3] J. Chadwick. Possible Existence of a Neutron. *Nature*, 129(312), February 1932.

[4] L. Szilárd. Improvements in or relating to the transmutation of chemical elements. *British patent number: GB630726*, March 1936.

[5] O. Hahn and F. Strassmann. Über den Nachweis und das Verhalten der bei der Bestrahlung des Urans mittels Neutronen entstehenden Erdalkalimetalle. *Naturwissenschaften*, 27(1):11–15, December 1939.

[6] L. Meitner and O. R. Frisch. Disintegration of Uranium by Neutrons: a New Type of Nuclear Reaction. *Physica Scripta*, 143(3615):239–240, February 1939.

[7] O. R. Frisch. Physical Evidence for the Division of Heavy Nuclei under Neutron Bombardment. *Nature*, 143(3616):276–276, February 1943.

[8] O. Hahn and F. Strassman. Nachweis der Entstehung aktiver Bariumisotope aus Uran und Thorium durch Neutronenbestrahlung; Nachweis weiterer aktiver Bruchstücke bei der Uranspaltung. *Naturwissenschaften*, 27(89):3097–3100, October 1939.

[9] H. L. Anderson, E. Fermi, and Leo Szilárd. Neutron production and absorption in uranium. *The Physical Review*, 56:284–286, August 1939.

[10] F. Gauthier-Lafaye, P. Holliger, P.-L Blanc. Natural fission reactors in the Franceville Basin, Gabon: a review of the conditions and results of a "critical event" in a geologic system. *Geochimica et Cosmochimica Acta*, 60(25):4831–4852, 1996.

[11] P. K. Kuroda. On the Nuclear Physical Stability of the Uranium Minerals. *Journal of Chemical Physics*, 25(781-782):1295–1296, 1956.

[12] Binding energy curve. http://en.wikipedia.org/wiki/File:Binding_energy_curve_-_common_isotopes.svg.

[13] Fission event. http://en.wikipedia.org/wiki/File:Nuclear_fission.svg.

[14] Cross sections. http://atom.kaeri.re.kr/endfplot.shtml.

[15] Fuel rod. http://web.clas.ufl.edu/jur/199911/papers/paper_szollosy.html.

[16] Fuel assembly. http://www.mhi.co.jp/en/products/detail/core_and_fuel.html.

[17] Table of isotopes. http://nucleardata.nuclear.lu.se/nucleardata/toi/pdf/chart.pdf.

[18] Fission products. http://www.uwuses.uwaterloo.ca/~cchieh/cact/nuctek/fissionyield.html.

[19] G. I. Bell and S. Glasstone. *Nuclear Reactor Theory*, pages 44–47. Krieger, Huntington, New York, reprint edition, 1979.

[20] F. D. Giust, R. J. J. Stamm'ler and A. A. Ferri. *HELIOS 1.9 User Guide and Manual*. Studsvik Scandpower, 2005.

[21] M.J. Halsall. CACTUS, A Characteristics Solution to the Neutron Transport Equations in Complicated Geometries. *AEEW-R*, 1291, 1980.

[22] Gauss-Seidel method. http://mathworld.wolfram.com/Gauss-SeidelMethod.html.

[23] M. Tsuiki and W. H. Beere. A varitional nodal transport method for pressurized water reactor core calculations. *American Nuclear Society Topical Meeting in Mathematics and Computation*, 9 2005.

Appendix A

HELIOS input data

The HELIOS input data for the single assembly calculations is written by William H. Beere, and small modifications has been done to all these files by Lars K. Jakobsson.

A.1 AURORA single assembly input

A.1.1 Assembly type 1

Files needed:

- w211-00.inp - main input file
- general.SET - general input data
- mat.SET - material properties
- pinstr.SET - pin structure
- w00.SET - assembly structure
- cases_crd.SET - burnup cases to be calculated
- Ringhals.SETS - file containing data from set files
- assemblies.SETS - file containing data from assembly set files

w211-00.inp

```
+HEL

'w211-00' = CASE( $library47 / 'Hw211-00.hrf' /
                      'Single UO2 PWR assembly',
                      'w211-00-00' )

 &general      = SET( '..\SETS\Ringhals.SETS' / ) ! general input data !
```

```
 &mat          = SET( '..\SETS\Ringhals.SETS' / ) ! materials !
 &pinstr       = SET( '..\SETS\Ringhals.SETS' / ) ! pin-structure !

 &w00          = SET( '..\assemblies\assemblies.SETS' / ) ! assembly structure !

 $kint   = PAR(8)

!boundary condition!
 CNXw00  = BDRY( (1,4,4)$kint(0) )

!pin material overlay!
 pinovlm = OVLM( 'w211'  / ($'w00_UO2') - * - fuel )

!general material overlay!
 generalovlm = OVLM( 'GTSS-304'    / ($'w00_TT') - * - tube /
                     'FuelZrClad'  / ($'w00_UO2') - * - clad/
                     'CRSS-304'    / ($'w00_CR') - * - tube/
                     'ZrClad'      / ($'w00_CR') - * - clad ,
                                     ($'w00_TT') - * - clad /
                     'spacerMat'   / ($'w00_UO2') - * - spacer )

'waterDens_ZP' = OVLD( $'waterDensity_ZP' / * - 0 - modr ,
                                            * - * - modr )

 ovltZP  = OVLT( "557.06" / *-** )

 generalovld = OVLD( 1.0        / *-** )

 temperatureZP = OVST( ovltZP  )

 density     = OVSD( generalovld , 'waterDens_ZP' )

 &'cases'        = SET( '..\SETS\Ringhals.SETS' / ) !burnup and boron cases!

 Cells = AREA( *-<**> )           ! average of each cell !
 Ass   = AREA( <*-**> )           ! average of assembly  !

 ! What to save !

 AssData = MACRO( ng1, Ass / bu, tr, ab, fi, nf, kf, ch, p0, p1 )

 CellData5 = MACRO(ng5, Cells / tr, ab, fi, nf, kf, ch, p0, p1 )

'w211-00'  = RUN(OUT:1
            /MT:            ! opt1 !
               , 2          ! opt2, use zero-buckling !
               ,            ! opt2 !
            /OM:            ! output lines per page !
               ,            ! dimensioning safety factor !
               , 30 000 000 ! MMRY!
               ,            !CMMRY!
               ,200 000 000 !SMMRY!
               )
```

A.1.2 General input data

general.SET

```
+SET

 &ADD      = SET ( 'Ringhals.SETS' / &general )

 $library47  = PAR( '../library/hy047n18g19a.dat' )
 $library190 = PAR( '../library/hy190n48g19a.dat' )

 $sin225   = PAR( "0.3827" )
 $sin675   = PAR( "0.9239" )

 ng1       = GROUP(N/0)                ! just the one energy group, for bu !


 ng5       = GROUP(N/1.0E6             ! 5 energy groups !
                  ,9.119E+3
                  ,3.9279
                  ,2.7052E-1
                  ,0         )

!  Geometry !

 $AssPitch = PAR( "21.50"  )
 $pitch    = PAR( "$AssPitch / 17 ")   ! cell pitch distance                  !
                                       ! should be 1.260 cm                  !
                                       ! modelled as 1.2647 cm to account for  !
                                       ! gap arround assemblies               !
 $CoreHeight = PAR( "365.76" )         ! active core height                   !

! boron glass rod !

 $BGtubeii = PAR( "0.2140" )           ! boron glass inner tube inner         !
 $BGtubeio = PAR( "0.2305" )           ! boron glass inner tube outer         !
 $BGi      = PAR( "0.2413" )           ! boron glass inner                    !
 $BGo      = PAR( "0.4267" )           ! boron glass outer                    !
 $BGtubeoi = PAR( "0.4369" )           ! boron glass outer tube inner         !
 $BGtubeoo = PAR( "0.4839" )           ! boron glass outer tube outer         !

! boron glass smears !

 $BGtubeiSmear =
  PAR( " ( $BGtubeio**2 - $BGtubeii**2 ) / ( $BGi**2 ) " )
                                       ! boron glass inner tube is smeared out !
                                       ! from centre to boron glass inner      !
 $BGtubeoSmear =
  PAR( " ( $BGtubeoo**2 - $BGtubeoi**2 ) / ( $BGtubeoo**2 - $BGo**2 ) " )
                                       ! boron glass outer tube is smeared out !
                                       ! to boron glass outer                  !

! boron glass CCS !
```

```
 BGpin = CCS( $BGi ,
              $BGo ,
              $BGtubeoo /
              4 /                       ! 4 radial divisions                    !
              tubei, tubei, tubei, tubei,
              glass,glass,glass,glass,
              tubeo, tubeo, tubeo, tubeo )

! fuel rods !

 $fuelo    = PAR( "0.4096" )           ! fuel outer radius                     !
 $cladi    = PAR( "0.4178" )           ! clad inner radius                     !
 $clado    = PAR( "0.4750" )           ! clad outer radius                     !
 $spacero  = PAR( "0.4769" )           ! ring for spacer material              !

! fuel rod smears !

 $FuelTubeSmear =
  PAR( " ( $clado**2 - $cladi**2 ) / ( $clado**2 - $fuelo**2 ) " )
                                       ! fuel cladding is smeared out to fuel  !
                                       ! outer                                 !

! fuel CCS !

 FuelPin = CCS( " $fuelo * 0.8**0.5 ",
                 $fuelo ,               ! outer fuel layer                      !
                 $clado ,               ! cladding                              !
                 $spacero /             ! spacer                                !
                 4 /                    ! 4 radial divisions                    !
                 fuel, fuel, fuel, fuel,
                 fuel, fuel, fuel, fuel,
                 clad, clad, clad, clad,
                 spacer,spacer,spacer,spacer)


! guide tube !

 $Gtubei   = PAR( "0.255" )            ! Guide tube inner radius               !
 $Gtubeo   = PAR( "0.380" )            ! Guide tube outer radius               !
 $ABBtubei = PAR( "0.5715" )          ! Guide tube Westinghouse ABB (E50) inner!
 $ABBtubeo = PAR( "0.6121" )          ! Guide tube Westinghouse ABB (E50) outer!

 $GTsmear  = PAR( " ( $Gtubeo**2 - $Gtubei**2 ) / ( $Gtubeo**2 ) " )
                                      ! guide tube smeared out into air region !
! control rods !

 $CRo      = PAR( "0.4331" )           ! outer radius control rod              !
 $CRtubei  = PAR( "0.4369" )           ! inner radius control rod tube         !
 $CRtubeo  = PAR( "0.4839" )           ! outer radius control rod tube         !

! control rod tube smear !

 $CRtubeSmear =
  PAR( " ( $CRtubeo**2 - $CRtubei**2 ) / ( $CRtubeo**2 - $CRo**2 ) " )
```

```
                                    ! control rod tube is smeared out to    !
                                    ! outer diameter of control rod         !
! CRD, guide and thimble CCS !

! control rod CCS !

 CRpin = CCS( $CRo ,
              $CRtubeo,
              $ABBtubei,
              $ABBtubeo /
              4 /                   ! 4 radial divisions !
              crd, crd, crd, crd,
              tube, tube, tube, tube,
              modr,modr,modr,modr,
              clad,clad,clad,clad )


 ThimblePin = CCS(  $Gtubeo ,
                    $ABBtubei,
                    $ABBtubeo /
                    4 /             ! 4 radial divisions !
                    tube,tube,tube,tube,
                    modr,modr,modr,modr,
                    clad,clad,clad,clad )

 GuidePin  = CCS( "$ABBtubei * 0.5 ",
                  $ABBtubei,
                  $ABBtubeo /
                  4(2,3) /          ! 4 radial divisions from ring 2 !
                  modr,
                  modr, modr, modr, modr,
                  clad, clad, clad, clad )
```

A.1.3 Material properties

mat.SET

```
+SET

 &ADD      = SET ( 'Ringhals.SETS' / &mat )

! Materials !
! Taken from problem description from Urban Sandberg !
! Benchmark calculations of power !
! distribution within fuel assemblies !

! Control rods not used !
 CRD = MAT(NB/ 10.17 / ! 80 % Ag !
                   47107, "80.0 * 0.5183 ";   ! Ag107 !
                   47109, "80.0 * 0.4817 ";   ! Ag109 !
                   ! 15 % In !
                   49113, "15.0 * 0.043  ";   ! In113 !
                   49115, "15.0 * 0.957  ";   ! In115 !
```

```
                    !  5 % Cd !
                    ! Cd106 and Cd108 not available !
                    48110, " 5.0 * 0.1251 ";   ! Cd110 !
                    48111, " 5.0 * 0.1281 ";   ! Cd111 !
                    48112, " 5.0 * 0.2413 ";   ! Cd112 !
                    48113, " 5.0 * 0.1222 ";   ! Cd113 !
                    48114, " 5.0 * 0.2872 ")   ! Cd114 !

! He material, for gaps !

! Avoid use of He, better to smear !

! Steel SS-304 !

! Density obtained from Goodfellow catalogue, rho = 7.93 !
! Note: desity is taken from CASMO input, rho = 8.00 !
'SS-304' = MAT( 8.00/ 26001, 100 )  ! generic SS-304 in Helios !

'CRSS-304' = MAT(NB/ "8.00 * $CRtubeSmear " / 26001, 100 )
                            ! SS in crontrol rod smeared !

'BGiSS-304' = MAT(NB/ "8.00 * $BGtubeiSmear " / 26001, 100 )
                            ! SS inner tube in boron glass rod smeared !

'BGoSS-304' = MAT(NB/ "8.00 * $BGtubeoSmear " / 26001, 100 )
                            ! SS outer tube in boron glass rod smeared !

'GTSS-304' = MAT(NB/  "8.00 * $GTsmear  " /
                    26001, 100    ;       ! SS in guide tube smeared !
                    92235, 1.0E-6 )

! Boron Glass / Pyrex !
! as defined in problem description !
! NOTE: density is taken from CASMO input, !
 BorGlass = MAT(NB/ 2.23/ 5010,  0.712 ;  ! B-10 !
                       5011,  3.17  ;  ! B-11 !
                      14000, 37.5   ;  ! Natural Si !
                      13027,  1.16  ;  ! Al !
                      11023,  3.78  ;  ! Na !
                       8016, 53.7   )  ! O !

! Zirconium 4 !
! just one type of Zirconium used !
! taken from NEACRP problem !

ZrClad    = MAT(NB/ 0 / 40000 , "4.30E-2 " )


FuelZrClad    = MAT(NB/ 0 / 40000 , "4.30E-2 * $FuelTubeSmear  " )

                             ! fuel cladding smeared !

! fuel  !
```

```
'w211'       = MAT( 10.30 / 92235,  2.108;
                            92238, 97.892;
                             8001,  0 )

'w260'       = MAT( 10.21 / 92235,  2.597;
                            92238, 97.403;
                             8001,  0 )

'w310'       = MAT( 10.23 / 92235,  3.101;
                            92238, 96.899;
                             8001,  0 )

! moderator !

!From an output of SIMULATE M:\CUACOS_REV\MATC\r2-c23-burn.sum    !
!                                                                !
!>SUMMARY                                                        !
!>     1     0     0.00000    18.07108     0                     !
!>           0     0.00000                                      !
!>     0.00000   100.00000  1420.08948   161.30402   546.79999   !
!>  2277.10010     0.00000     0.00000                           !
!>     1.00000  2277.10010   161.30402                          !
!               ^^^^^^^^^^                                       !
!               pressure (psia)                                  !
!                                                                !
!Therefore, the nominal pressure for Ringhals2 is                !
!                                                                !
!Nominal pressure in MPa = 2277.10010 *6.895E-3 = 15.7(MPa).     !
!                                                                !
! From Makoto                                                 !

! values taken from steam library in tempo !
! T(K)   T(C)     rho        !
! 293    20       1.0050     !
! 403    130      0.9425     !
! 559    286      0.7541     !
! 595    322      0.6752     !
!                            !
! for zero power state       !
! 557.06 283.91   0.7576     !

$'waterDensity_ZP'   = PAR( "0.7576" )


! spacers need to be added to water !
! spacer density is 8.180 g/cm3 !
! spacer mass per unit length of fuel rod is 12.30 g/cm !
! so they occupy an area of 12.30 / 8.180 cm2   !
!                         = 1.504 cm2           !
! area of water is approximately:               !
! assembly pitch ^2 - fuel rod area * 17^2      !
! = 21,5^2  - PI * 0.4750^2 * 289               !
! = 462.25  - 204.84                            !
! = 257.41                                      !
```

```
! so the spacers occupy 1.504 / 257.41 of the water !
!                     = 0.584 % volume          !
! alternatively !
! add as extra ring around fuel !
! area per fuel pin = 1.504 / 264               !
!                   = 0.005697                  !
! external radius of cladding 0.4750            !
! additional radius to incease volume is        !
!       = sqrt(( 0.005697 + PI * 0.4750^2 )/PI) !
!       = 0.4769                                !

spacerMat = MAT( NB / 8.180 /
                 28001, "77.33" ;  ! inconel 625 !
                 26001, "22.67" )  ! SS-304      !

! Note Inconel 625 is used instead of Inconel 718 !
! as only the former was available in the library !

'water_0B'   = MAT(NB/ 1.0 /  1001, 11.19 ;
                              8016, 88.81 )

'water_400B' = MAT(NB/ 1.0 /  1001, 11.19 ;
                              8016, 88.81 ;
                              5000, 400.0E-4 )

'water_800B' = MAT(NB/ 1.0 /  1001, 11.19 ;
                              8016, 88.81 ;
                              5000, 800.0E-4 )

'water_1400B' = MAT(NB/ 1.0 /  1001, 11.19 ;
                               8016, 88.81 ;
                               5000, 1400.0E-4 )
```

A.1.4 Pin structure

pinstr.SET

```
+SET

 &ADD      = SET ( 'Ringhals.SETS' / &pinstr )

! structures !

 $PeriferyNodes = PAR(
             (" $pitch / 2", " $pitch / 2"),
             (" $pitch / 2", "-$pitch / 2"),
             ("-$pitch / 2", "-$pitch / 2"),
             ("-$pitch / 2", " $pitch / 2")  )

 $CRPinNodes   = PAR(
             (" $CRtubeo * $sin225", " $CRtubeo * $sin675"),
             (" $CRtubeo * $sin675", " $CRtubeo * $sin225"),
             (" $CRtubeo * $sin675", "-$CRtubeo * $sin225"),
             (" $CRtubeo * $sin225", "-$CRtubeo * $sin675"),
```

```
            ("-$CRtubeo * $sin225", "-$CRtubeo * $sin675"),
            ("-$CRtubeo * $sin675", "-$CRtubeo * $sin225"),
            ("-$CRtubeo * $sin675", " $CRtubeo * $sin225"),
            ("-$CRtubeo * $sin225", " $CRtubeo * $sin675")
                )

$BGPinNodes   = PAR(
            (" $BGtubeoo * $sin225", " $BGtubeoo * $sin675"),
            (" $BGtubeoo * $sin675", " $BGtubeoo * $sin225"),
            (" $BGtubeoo * $sin675", "-$BGtubeoo * $sin225"),
            (" $BGtubeoo * $sin225", "-$BGtubeoo * $sin675"),
            ("-$BGtubeoo * $sin225", "-$BGtubeoo * $sin675"),
            ("-$BGtubeoo * $sin675", "-$BGtubeoo * $sin225"),
            ("-$BGtubeoo * $sin675", " $BGtubeoo * $sin225"),
            ("-$BGtubeoo * $sin225", " $BGtubeoo * $sin675")
                )

$FuelPinNodes   = PAR(
            (" $spacero * $sin225", " $spacero * $sin675"),
            (" $spacero * $sin675", " $spacero * $sin225"),
            (" $spacero * $sin675", "-$spacero * $sin225"),
            (" $spacero * $sin225", "-$spacero * $sin675"),
            ("-$spacero * $sin225", "-$spacero * $sin675"),
            ("-$spacero * $sin675", "-$spacero * $sin225"),
            ("-$spacero * $sin675", " $spacero * $sin225"),
            ("-$spacero * $sin225", " $spacero * $sin675")
                )

$GTPinNodes   = PAR(
            (" $ABBtubeo * $sin225", " $ABBtubeo * $sin675"),
            (" $ABBtubeo * $sin675", " $ABBtubeo * $sin225"),
            (" $ABBtubeo * $sin675", "-$ABBtubeo * $sin225"),
            (" $ABBtubeo * $sin225", "-$ABBtubeo * $sin675"),
            ("-$ABBtubeo * $sin225", "-$ABBtubeo * $sin675"),
            ("-$ABBtubeo * $sin675", "-$ABBtubeo * $sin225"),
            ("-$ABBtubeo * $sin675", " $ABBtubeo * $sin225"),
            ("-$ABBtubeo * $sin225", " $ABBtubeo * $sin675")
                )

 $OuterNodes  = PAR(
            (" $pitch / 6", " $pitch / 2"),
            (" $pitch / 2", " $pitch / 6"),
            (" $pitch / 2", "-$pitch / 6"),
            (" $pitch / 6", "-$pitch / 2"),
            ("-$pitch / 6", "-$pitch / 2"),
            ("-$pitch / 2", "-$pitch / 6"),
            ("-$pitch / 2", " $pitch / 6"),
            ("-$pitch / 6", " $pitch / 2")
                )

 $InnerNodes = PAR(
            (" $pitch / 6", " $pitch / 6"),
            (" $pitch / 6", "-$pitch / 6"),
            ("-$pitch / 6", "-$pitch / 6"),
```

```
             ("-$pitch / 6", " $pitch / 6")
                 )

 $InnerNodes2 = PAR(
             (        0    , " $pitch / 3"),
             (" $pitch / 3", " $pitch / 3"),
             (" $pitch / 3",       0      ),
             (" $pitch / 3", "-$pitch / 3"),
             (        0    , "-$pitch / 3"),
             ("-$pitch / 3", "-$pitch / 3"),
             ("-$pitch / 3",       0      ),
             ("-$pitch / 3", " $pitch / 3"),
             (      0      ,       0      )
                  )

 $str1regions = PAR( )

 $str2regions = PAR(
             13,1,14,6,5,   modr;
             14,15,7,6,     modr;
             15,2,16,8,7,   modr;
             16,17,9,8,     modr;
             17,3,18,10,9,  modr;
             18,19,11,10,   modr;
             19,4,20,12,11, modr
                  )

! for control rods !

CRStr  = STR( $PeriferyNodes,
              $GTPinNodes,
              $OuterNodes /
              4, modr /
              CRpin( 0, 0) /
              $str2regions )

! for boron glass rods !

BGStr  = STR( $PeriferyNodes,
              $BGPinNodes,
              $OuterNodes /
              4, modr /
              BGpin( 0, 0) /
              $str2regions )



! For fuel pins !

FuelStr = STR( $PeriferyNodes,
              $FuelPinNodes,
              $OuterNodes /
              4, modr /
              FuelPin( 0, 0) /
```

```
              $str2regions )


! for guide tubes !

GuideStr  = STR( $PeriferyNodes,
              $GTPinNodes,
              $OuterNodes /
              4, modr /
              GuidePin( 0, 0) /
              $str2regions )

! for thimble tubes !

ThimbleStr  = STR( $PeriferyNodes,
              $GTPinNodes,
              $OuterNodes /
              4, modr /
              ThimblePin( 0, 0) /
              $str2regions )

! for reflector assembly !

REFstr = STR( $PeriferyNodes/
              4, modr//
              $str1regions )
```

A.1.5 Assembly structure

w00.SET

```
+SET

 &ADD = SET( 'assemblies.SETS' / &'w00' )

 'CNXw00' = CNX(
! line 1!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 2!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 3!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr,
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
! line 4!
 FuelStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, FuelStr,
! line 5!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
```

```
! line 6!
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr,
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr,
! line 7!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 8!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 9!
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, ThimbleStr,
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr,
! line 10!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 11!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 12!
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr,
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr,
! line 13!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 14!
 FuelStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, FuelStr,
! line 15!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, CRStr, FuelStr, FuelStr, CRStr,
 FuelStr, FuelStr, CRStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
! line 16!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 17!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr
!line  1!
/ (  1,1)(  1,2)$kint(  2,4)(  2,3)
/ (  2,1)(  2,2)$kint(  3,4)(  3,3)
/ (  3,1)(  3,2)$kint(  4,4)(  4,3)
/ (  4,1)(  4,2)$kint(  5,4)(  5,3)
/ (  5,1)(  5,2)$kint(  6,4)(  6,3)
/ (  6,1)(  6,2)$kint(  7,4)(  7,3)
/ (  7,1)(  7,2)$kint(  8,4)(  8,3)
/ (  8,1)(  8,2)$kint(  9,4)(  9,3)
/ (  9,1)(  9,2)$kint( 10,4)( 10,3)
/ ( 10,1)( 10,2)$kint( 11,4)( 11,3)
```

```
/ ( 11,1)( 11,2)$kint( 12,4)( 12,3)
/ ( 12,1)( 12,2)$kint( 13,4)( 13,3)
/ ( 13,1)( 13,2)$kint( 14,4)( 14,3)
/ ( 14,1)( 14,2)$kint( 15,4)( 15,3)
/ ( 15,1)( 15,2)$kint( 16,4)( 16,3)
/ ( 16,1)( 16,2)$kint( 17,4)( 17,3)
!line  2!
/ (  1,2)(  1,3)$kint( 18,1)( 18,4)
/ ( 18,1)( 18,2)$kint( 19,4)( 19,3)
/ ( 19,1)( 19,2)$kint( 20,4)( 20,3)
/ ( 20,1)( 20,2)$kint( 21,4)( 21,3)
/ ( 21,1)( 21,2)$kint( 22,4)( 22,3)
/ ( 22,1)( 22,2)$kint( 23,4)( 23,3)
/ ( 23,1)( 23,2)$kint( 24,4)( 24,3)
/ ( 24,1)( 24,2)$kint( 25,4)( 25,3)
/ ( 25,1)( 25,2)$kint( 26,4)( 26,3)
/ ( 26,1)( 26,2)$kint( 27,4)( 27,3)
/ ( 27,1)( 27,2)$kint( 28,4)( 28,3)
/ ( 28,1)( 28,2)$kint( 29,4)( 29,3)
/ ( 29,1)( 29,2)$kint( 30,4)( 30,3)
/ ( 30,1)( 30,2)$kint( 31,4)( 31,3)
/ ( 31,1)( 31,2)$kint( 32,4)( 32,3)
/ ( 32,1)( 32,2)$kint( 33,4)( 33,3)
/ ( 33,1)( 33,2)$kint( 34,4)( 34,3)
!line  3!
/ ( 18,2)( 18,3)$kint( 35,1)( 35,4)
/ ( 35,1)( 35,2)$kint( 36,4)( 36,3)
/ ( 36,1)( 36,2)$kint( 37,4)( 37,3)
/ ( 37,1)( 37,2)$kint( 38,4)( 38,3)
/ ( 38,1)( 38,2)$kint( 39,4)( 39,3)
/ ( 39,1)( 39,2)$kint( 40,4)( 40,3)
/ ( 40,1)( 40,2)$kint( 41,4)( 41,3)
/ ( 41,1)( 41,2)$kint( 42,4)( 42,3)
/ ( 42,1)( 42,2)$kint( 43,4)( 43,3)
/ ( 43,1)( 43,2)$kint( 44,4)( 44,3)
/ ( 44,1)( 44,2)$kint( 45,4)( 45,3)
/ ( 45,1)( 45,2)$kint( 46,4)( 46,3)
/ ( 46,1)( 46,2)$kint( 47,4)( 47,3)
/ ( 47,1)( 47,2)$kint( 48,4)( 48,3)
/ ( 48,1)( 48,2)$kint( 49,4)( 49,3)
/ ( 49,1)( 49,2)$kint( 50,4)( 50,3)
/ ( 50,1)( 50,2)$kint( 51,4)( 51,3)
!line  4!
/ ( 35,2)( 35,3)$kint( 52,1)( 52,4)
/ ( 52,1)( 52,2)$kint( 53,4)( 53,3)
/ ( 53,1)( 53,2)$kint( 54,4)( 54,3)
/ ( 54,1)( 54,2)$kint( 55,4)( 55,3)
/ ( 55,1)( 55,2)$kint( 56,4)( 56,3)
/ ( 56,1)( 56,2)$kint( 57,4)( 57,3)
/ ( 57,1)( 57,2)$kint( 58,4)( 58,3)
/ ( 58,1)( 58,2)$kint( 59,4)( 59,3)
/ ( 59,1)( 59,2)$kint( 60,4)( 60,3)
/ ( 60,1)( 60,2)$kint( 61,4)( 61,3)
/ ( 61,1)( 61,2)$kint( 62,4)( 62,3)
```

```
/ ( 62,1)( 62,2)$kint( 63,4)( 63,3)
/ ( 63,1)( 63,2)$kint( 64,4)( 64,3)
/ ( 64,1)( 64,2)$kint( 65,4)( 65,3)
/ ( 65,1)( 65,2)$kint( 66,4)( 66,3)
/ ( 66,1)( 66,2)$kint( 67,4)( 67,3)
/ ( 67,1)( 67,2)$kint( 68,4)( 68,3)
!line  5!
/ ( 52,2)( 52,3)$kint( 69,1)( 69,4)
/ ( 69,1)( 69,2)$kint( 70,4)( 70,3)
/ ( 70,1)( 70,2)$kint( 71,4)( 71,3)
/ ( 71,1)( 71,2)$kint( 72,4)( 72,3)
/ ( 72,1)( 72,2)$kint( 73,4)( 73,3)
/ ( 73,1)( 73,2)$kint( 74,4)( 74,3)
/ ( 74,1)( 74,2)$kint( 75,4)( 75,3)
/ ( 75,1)( 75,2)$kint( 76,4)( 76,3)
/ ( 76,1)( 76,2)$kint( 77,4)( 77,3)
/ ( 77,1)( 77,2)$kint( 78,4)( 78,3)
/ ( 78,1)( 78,2)$kint( 79,4)( 79,3)
/ ( 79,1)( 79,2)$kint( 80,4)( 80,3)
/ ( 80,1)( 80,2)$kint( 81,4)( 81,3)
/ ( 81,1)( 81,2)$kint( 82,4)( 82,3)
/ ( 82,1)( 82,2)$kint( 83,4)( 83,3)
/ ( 83,1)( 83,2)$kint( 84,4)( 84,3)
/ ( 84,1)( 84,2)$kint( 85,4)( 85,3)
!line  6!
/ ( 69,2)( 69,3)$kint( 86,1)( 86,4)
/ ( 86,1)( 86,2)$kint( 87,4)( 87,3)
/ ( 87,1)( 87,2)$kint( 88,4)( 88,3)
/ ( 88,1)( 88,2)$kint( 89,4)( 89,3)
/ ( 89,1)( 89,2)$kint( 90,4)( 90,3)
/ ( 90,1)( 90,2)$kint( 91,4)( 91,3)
/ ( 91,1)( 91,2)$kint( 92,4)( 92,3)
/ ( 92,1)( 92,2)$kint( 93,4)( 93,3)
/ ( 93,1)( 93,2)$kint( 94,4)( 94,3)
/ ( 94,1)( 94,2)$kint( 95,4)( 95,3)
/ ( 95,1)( 95,2)$kint( 96,4)( 96,3)
/ ( 96,1)( 96,2)$kint( 97,4)( 97,3)
/ ( 97,1)( 97,2)$kint( 98,4)( 98,3)
/ ( 98,1)( 98,2)$kint( 99,4)( 99,3)
/ ( 99,1)( 99,2)$kint(100,4)(100,3)
/ (100,1)(100,2)$kint(101,4)(101,3)
/ (101,1)(101,2)$kint(102,4)(102,3)
!line  7!
/ ( 86,2)( 86,3)$kint(103,1)(103,4)
/ (103,1)(103,2)$kint(104,4)(104,3)
/ (104,1)(104,2)$kint(105,4)(105,3)
/ (105,1)(105,2)$kint(106,4)(106,3)
/ (106,1)(106,2)$kint(107,4)(107,3)
/ (107,1)(107,2)$kint(108,4)(108,3)
/ (108,1)(108,2)$kint(109,4)(109,3)
/ (109,1)(109,2)$kint(110,4)(110,3)
/ (110,1)(110,2)$kint(111,4)(111,3)
/ (111,1)(111,2)$kint(112,4)(112,3)
/ (112,1)(112,2)$kint(113,4)(113,3)
```

```
/ (113,1)(113,2)$kint(114,4)(114,3)
/ (114,1)(114,2)$kint(115,4)(115,3)
/ (115,1)(115,2)$kint(116,4)(116,3)
/ (116,1)(116,2)$kint(117,4)(117,3)
/ (117,1)(117,2)$kint(118,4)(118,3)
/ (118,1)(118,2)$kint(119,4)(119,3)
!line  8!
/ (103,2)(103,3)$kint(120,1)(120,4)
/ (120,1)(120,2)$kint(121,4)(121,3)
/ (121,1)(121,2)$kint(122,4)(122,3)
/ (122,1)(122,2)$kint(123,4)(123,3)
/ (123,1)(123,2)$kint(124,4)(124,3)
/ (124,1)(124,2)$kint(125,4)(125,3)
/ (125,1)(125,2)$kint(126,4)(126,3)
/ (126,1)(126,2)$kint(127,4)(127,3)
/ (127,1)(127,2)$kint(128,4)(128,3)
/ (128,1)(128,2)$kint(129,4)(129,3)
/ (129,1)(129,2)$kint(130,4)(130,3)
/ (130,1)(130,2)$kint(131,4)(131,3)
/ (131,1)(131,2)$kint(132,4)(132,3)
/ (132,1)(132,2)$kint(133,4)(133,3)
/ (133,1)(133,2)$kint(134,4)(134,3)
/ (134,1)(134,2)$kint(135,4)(135,3)
/ (135,1)(135,2)$kint(136,4)(136,3)
!line  9!
/ (120,2)(120,3)$kint(137,1)(137,4)
/ (137,1)(137,2)$kint(138,4)(138,3)
/ (138,1)(138,2)$kint(139,4)(139,3)
/ (139,1)(139,2)$kint(140,4)(140,3)
/ (140,1)(140,2)$kint(141,4)(141,3)
/ (141,1)(141,2)$kint(142,4)(142,3)
/ (142,1)(142,2)$kint(143,4)(143,3)
/ (143,1)(143,2)$kint(144,4)(144,3)
/ (144,1)(144,2)$kint(145,4)(145,3)
/ (145,1)(145,2)$kint(146,4)(146,3)
/ (146,1)(146,2)$kint(147,4)(147,3)
/ (147,1)(147,2)$kint(148,4)(148,3)
/ (148,1)(148,2)$kint(149,4)(149,3)
/ (149,1)(149,2)$kint(150,4)(150,3)
/ (150,1)(150,2)$kint(151,4)(151,3)
/ (151,1)(151,2)$kint(152,4)(152,3)
/ (152,1)(152,2)$kint(153,4)(153,3)
!line 10!
/ (137,2)(137,3)$kint(154,1)(154,4)
/ (154,1)(154,2)$kint(155,4)(155,3)
/ (155,1)(155,2)$kint(156,4)(156,3)
/ (156,1)(156,2)$kint(157,4)(157,3)
/ (157,1)(157,2)$kint(158,4)(158,3)
/ (158,1)(158,2)$kint(159,4)(159,3)
/ (159,1)(159,2)$kint(160,4)(160,3)
/ (160,1)(160,2)$kint(161,4)(161,3)
/ (161,1)(161,2)$kint(162,4)(162,3)
/ (162,1)(162,2)$kint(163,4)(163,3)
/ (163,1)(163,2)$kint(164,4)(164,3)
```

```
/ (164,1)(164,2)$kint(165,4)(165,3)
/ (165,1)(165,2)$kint(166,4)(166,3)
/ (166,1)(166,2)$kint(167,4)(167,3)
/ (167,1)(167,2)$kint(168,4)(168,3)
/ (168,1)(168,2)$kint(169,4)(169,3)
/ (169,1)(169,2)$kint(170,4)(170,3)
!line 11!
/ (154,2)(154,3)$kint(171,1)(171,4)
/ (171,1)(171,2)$kint(172,4)(172,3)
/ (172,1)(172,2)$kint(173,4)(173,3)
/ (173,1)(173,2)$kint(174,4)(174,3)
/ (174,1)(174,2)$kint(175,4)(175,3)
/ (175,1)(175,2)$kint(176,4)(176,3)
/ (176,1)(176,2)$kint(177,4)(177,3)
/ (177,1)(177,2)$kint(178,4)(178,3)
/ (178,1)(178,2)$kint(179,4)(179,3)
/ (179,1)(179,2)$kint(180,4)(180,3)
/ (180,1)(180,2)$kint(181,4)(181,3)
/ (181,1)(181,2)$kint(182,4)(182,3)
/ (182,1)(182,2)$kint(183,4)(183,3)
/ (183,1)(183,2)$kint(184,4)(184,3)
/ (184,1)(184,2)$kint(185,4)(185,3)
/ (185,1)(185,2)$kint(186,4)(186,3)
/ (186,1)(186,2)$kint(187,4)(187,3)
!line 12!
/ (171,2)(171,3)$kint(188,1)(188,4)
/ (188,1)(188,2)$kint(189,4)(189,3)
/ (189,1)(189,2)$kint(190,4)(190,3)
/ (190,1)(190,2)$kint(191,4)(191,3)
/ (191,1)(191,2)$kint(192,4)(192,3)
/ (192,1)(192,2)$kint(193,4)(193,3)
/ (193,1)(193,2)$kint(194,4)(194,3)
/ (194,1)(194,2)$kint(195,4)(195,3)
/ (195,1)(195,2)$kint(196,4)(196,3)
/ (196,1)(196,2)$kint(197,4)(197,3)
/ (197,1)(197,2)$kint(198,4)(198,3)
/ (198,1)(198,2)$kint(199,4)(199,3)
/ (199,1)(199,2)$kint(200,4)(200,3)
/ (200,1)(200,2)$kint(201,4)(201,3)
/ (201,1)(201,2)$kint(202,4)(202,3)
/ (202,1)(202,2)$kint(203,4)(203,3)
/ (203,1)(203,2)$kint(204,4)(204,3)
!line 13!
/ (188,2)(188,3)$kint(205,1)(205,4)
/ (205,1)(205,2)$kint(206,4)(206,3)
/ (206,1)(206,2)$kint(207,4)(207,3)
/ (207,1)(207,2)$kint(208,4)(208,3)
/ (208,1)(208,2)$kint(209,4)(209,3)
/ (209,1)(209,2)$kint(210,4)(210,3)
/ (210,1)(210,2)$kint(211,4)(211,3)
/ (211,1)(211,2)$kint(212,4)(212,3)
/ (212,1)(212,2)$kint(213,4)(213,3)
/ (213,1)(213,2)$kint(214,4)(214,3)
/ (214,1)(214,2)$kint(215,4)(215,3)
```

```
/ (215,1)(215,2)$kint(216,4)(216,3)
/ (216,1)(216,2)$kint(217,4)(217,3)
/ (217,1)(217,2)$kint(218,4)(218,3)
/ (218,1)(218,2)$kint(219,4)(219,3)
/ (219,1)(219,2)$kint(220,4)(220,3)
/ (220,1)(220,2)$kint(221,4)(221,3)
!line 14!
/ (205,2)(205,3)$kint(222,1)(222,4)
/ (222,1)(222,2)$kint(223,4)(223,3)
/ (223,1)(223,2)$kint(224,4)(224,3)
/ (224,1)(224,2)$kint(225,4)(225,3)
/ (225,1)(225,2)$kint(226,4)(226,3)
/ (226,1)(226,2)$kint(227,4)(227,3)
/ (227,1)(227,2)$kint(228,4)(228,3)
/ (228,1)(228,2)$kint(229,4)(229,3)
/ (229,1)(229,2)$kint(230,4)(230,3)
/ (230,1)(230,2)$kint(231,4)(231,3)
/ (231,1)(231,2)$kint(232,4)(232,3)
/ (232,1)(232,2)$kint(233,4)(233,3)
/ (233,1)(233,2)$kint(234,4)(234,3)
/ (234,1)(234,2)$kint(235,4)(235,3)
/ (235,1)(235,2)$kint(236,4)(236,3)
/ (236,1)(236,2)$kint(237,4)(237,3)
/ (237,1)(237,2)$kint(238,4)(238,3)
!line 15!
/ (222,2)(222,3)$kint(239,1)(239,4)
/ (239,1)(239,2)$kint(240,4)(240,3)
/ (240,1)(240,2)$kint(241,4)(241,3)
/ (241,1)(241,2)$kint(242,4)(242,3)
/ (242,1)(242,2)$kint(243,4)(243,3)
/ (243,1)(243,2)$kint(244,4)(244,3)
/ (244,1)(244,2)$kint(245,4)(245,3)
/ (245,1)(245,2)$kint(246,4)(246,3)
/ (246,1)(246,2)$kint(247,4)(247,3)
/ (247,1)(247,2)$kint(248,4)(248,3)
/ (248,1)(248,2)$kint(249,4)(249,3)
/ (249,1)(249,2)$kint(250,4)(250,3)
/ (250,1)(250,2)$kint(251,4)(251,3)
/ (251,1)(251,2)$kint(252,4)(252,3)
/ (252,1)(252,2)$kint(253,4)(253,3)
/ (253,1)(253,2)$kint(254,4)(254,3)
/ (254,1)(254,2)$kint(255,4)(255,3)
!line 16!
/ (239,2)(239,3)$kint(256,1)(256,4)
/ (256,1)(256,2)$kint(257,4)(257,3)
/ (257,1)(257,2)$kint(258,4)(258,3)
/ (258,1)(258,2)$kint(259,4)(259,3)
/ (259,1)(259,2)$kint(260,4)(260,3)
/ (260,1)(260,2)$kint(261,4)(261,3)
/ (261,1)(261,2)$kint(262,4)(262,3)
/ (262,1)(262,2)$kint(263,4)(263,3)
/ (263,1)(263,2)$kint(264,4)(264,3)
/ (264,1)(264,2)$kint(265,4)(265,3)
/ (265,1)(265,2)$kint(266,4)(266,3)
```

```
/ (266,1)(266,2)$kint(267,4)(267,3)
/ (267,1)(267,2)$kint(268,4)(268,3)
/ (268,1)(268,2)$kint(269,4)(269,3)
/ (269,1)(269,2)$kint(270,4)(270,3)
/ (270,1)(270,2)$kint(271,4)(271,3)
/ (271,1)(271,2)$kint(272,4)(272,3)
!line 17!
/ (256,2)(256,3)$kint(273,1)(273,4)
/ (273,1)(273,2)$kint(274,4)(274,3)
/ (274,1)(274,2)$kint(275,4)(275,3)
/ (275,1)(275,2)$kint(276,4)(276,3)
/ (276,1)(276,2)$kint(277,4)(277,3)
/ (277,1)(277,2)$kint(278,4)(278,3)
/ (278,1)(278,2)$kint(279,4)(279,3)
/ (279,1)(279,2)$kint(280,4)(280,3)
/ (280,1)(280,2)$kint(281,4)(281,3)
/ (281,1)(281,2)$kint(282,4)(282,3)
/ (282,1)(282,2)$kint(283,4)(283,3)
/ (283,1)(283,2)$kint(284,4)(284,3)
/ (284,1)(284,2)$kint(285,4)(285,3)
/ (285,1)(285,2)$kint(286,4)(286,3)
/ (286,1)(286,2)$kint(287,4)(287,3)
/ (287,1)(287,2)$kint(288,4)(288,3)
/ (288,1)(288,2)$kint(289,4)(289,3)
)
! CR !
$'w00_CR' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX! 40 !UX!!UX!,43 !UX!!UX!,46 !UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!,55 !UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!,65 !UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!,88 !UX!!UX!,91 !UX!!UX!,94 !UX!!UX!,97 !UX!!UX!,100 !UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!,139 !UX!!UX!,142 !UX!!UX!!TT!!UX!!UX!,148 !UX!!UX!,151 !UX!!UX!

!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!,190 !UX!!UX!,193 !UX!!UX!,196 !UX!!UX!,199 !UX!!UX!,202 !UX!!UX!

!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!,225 !UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!,235 !UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!,244 !UX!!UX!,247 !UX!!UX!,250 !UX!!UX!!UX!!UX!!UX!

!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )
! BG !
$'w00_BG' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!UX!
```

```
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!TT!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )
! UO2 !
$'w00_UO2' = PAR(
  1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17
,18 ,19 ,20 ,21 ,22 ,23 ,24 ,25 ,26 ,27 ,28 ,29 ,30 ,31 ,32 ,33 ,34
,35 ,36 ,37 ,38 ,39 !CR!,41 ,42 !CR!,44 ,45 !CR!,47 ,48 ,49 ,50 ,51
,52 ,53 ,54 !CR!,56 ,57 ,58 ,59 ,60 ,61 ,62 ,63 ,64 !CR!,66 ,67 ,68
,69 ,70 ,71 ,72 ,73 ,74 ,75 ,76 ,77 ,78 ,79 ,80 ,81 ,82 ,83 ,84 ,85
,86 ,87 !CR!,89 ,90 !CR!,92 ,93 !CR!,95 ,96 !CR!,98 ,99 !CR!,101 ,102
,103 ,104 ,105 ,106 ,107 ,108 ,109 ,110 ,111 ,112 ,113 ,114 ,115 ,116 ,117
     ,118 ,119
,120 ,121 ,122 ,123 ,124 ,125 ,126 ,127 ,128 ,129 ,130 ,131 ,132 ,133 ,134
     ,135 ,136
,137 ,138 !CR!,140 ,141 !CR!,143 ,144 !TT!,146 ,147 !CR!,149 ,150 !CR!,152
     ,153
,154 ,155 ,156 ,157 ,158 ,159 ,160 ,161 ,162 ,163 ,164 ,165 ,166 ,167 ,168
     ,169 ,170
,171 ,172 ,173 ,174 ,175 ,176 ,177 ,178 ,179 ,180 ,181 ,182 ,183 ,184 ,185
     ,186 ,187
,188 ,189 !CR!,191 ,192 !CR!,194 ,195 !CR!,197 ,198 !CR!,200 ,201 !CR!,203
     ,204
,205 ,206 ,207 ,208 ,209 ,210 ,211 ,212 ,213 ,214 ,215 ,216 ,217 ,218 ,219
     ,220 ,221
,222 ,223 ,224 !CR!,226 ,227 ,228 ,229 ,230 ,231 ,232 ,233 ,234 !CR!,236
     ,237 ,238
,239 ,240 ,241 ,242 ,243 !CR!,245 ,246 !CR!,248 ,249 !CR!,251 ,252 ,253
     ,254 ,255
,256 ,257 ,258 ,259 ,260 ,261 ,262 ,263 ,264 ,265 ,266 ,267 ,268 ,269 ,270
     ,271 ,272
,273 ,274 ,275 ,276 ,277 ,278 ,279 ,280 ,281 ,282 ,283 ,284 ,285 ,286 ,287
     ,288 ,289
 )
! TT !
$'w00_TT' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
```

```
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX! 145 !UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )
! GT !
$'w00_GT' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!TT!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!CR!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )

 'CNXw20bps' = CNX(
! line 1!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 2!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 3!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr, BGStr,
 FuelStr, FuelStr, BGStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
! line 4!
 FuelStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr, FuelStr,
! line 5!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 6!
 FuelStr, FuelStr, BGStr, FuelStr, FuelStr, GuideStr, FuelStr, FuelStr, BGStr,
 FuelStr, FuelStr, GuideStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr,
```

```
! line 7!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 8!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 9!
 FuelStr, FuelStr, BGStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr, ThimbleStr,
 FuelStr, FuelStr, BGStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr,
! line 10!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 11!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 12!
 FuelStr, FuelStr, BGStr, FuelStr, FuelStr, GuideStr, FuelStr, FuelStr, BGStr,
 FuelStr, FuelStr, GuideStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr,
! line 13!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 14!
 FuelStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr, FuelStr,
! line 15!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, BGStr, FuelStr, FuelStr, BGStr,
 FuelStr, FuelStr, BGStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
! line 16!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr,
! line 17!
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr, FuelStr,
 FuelStr
!line  1!
/ (  1,1)(  1,2)$kint(  2,4)(  2,3)
/ (  2,1)(  2,2)$kint(  3,4)(  3,3)
/ (  3,1)(  3,2)$kint(  4,4)(  4,3)
/ (  4,1)(  4,2)$kint(  5,4)(  5,3)
/ (  5,1)(  5,2)$kint(  6,4)(  6,3)
/ (  6,1)(  6,2)$kint(  7,4)(  7,3)
/ (  7,1)(  7,2)$kint(  8,4)(  8,3)
/ (  8,1)(  8,2)$kint(  9,4)(  9,3)
/ (  9,1)(  9,2)$kint( 10,4)( 10,3)
/ ( 10,1)( 10,2)$kint( 11,4)( 11,3)
/ ( 11,1)( 11,2)$kint( 12,4)( 12,3)
/ ( 12,1)( 12,2)$kint( 13,4)( 13,3)
/ ( 13,1)( 13,2)$kint( 14,4)( 14,3)
```

```
/ ( 14,1)( 14,2)$kint( 15,4)( 15,3)
/ ( 15,1)( 15,2)$kint( 16,4)( 16,3)
/ ( 16,1)( 16,2)$kint( 17,4)( 17,3)
!line  2!
/ (  1,2)(  1,3)$kint( 18,1)( 18,4)
/ ( 18,1)( 18,2)$kint( 19,4)( 19,3)
/ ( 19,1)( 19,2)$kint( 20,4)( 20,3)
/ ( 20,1)( 20,2)$kint( 21,4)( 21,3)
/ ( 21,1)( 21,2)$kint( 22,4)( 22,3)
/ ( 22,1)( 22,2)$kint( 23,4)( 23,3)
/ ( 23,1)( 23,2)$kint( 24,4)( 24,3)
/ ( 24,1)( 24,2)$kint( 25,4)( 25,3)
/ ( 25,1)( 25,2)$kint( 26,4)( 26,3)
/ ( 26,1)( 26,2)$kint( 27,4)( 27,3)
/ ( 27,1)( 27,2)$kint( 28,4)( 28,3)
/ ( 28,1)( 28,2)$kint( 29,4)( 29,3)
/ ( 29,1)( 29,2)$kint( 30,4)( 30,3)
/ ( 30,1)( 30,2)$kint( 31,4)( 31,3)
/ ( 31,1)( 31,2)$kint( 32,4)( 32,3)
/ ( 32,1)( 32,2)$kint( 33,4)( 33,3)
/ ( 33,1)( 33,2)$kint( 34,4)( 34,3)
!line  3!
/ ( 18,2)( 18,3)$kint( 35,1)( 35,4)
/ ( 35,1)( 35,2)$kint( 36,4)( 36,3)
/ ( 36,1)( 36,2)$kint( 37,4)( 37,3)
/ ( 37,1)( 37,2)$kint( 38,4)( 38,3)
/ ( 38,1)( 38,2)$kint( 39,4)( 39,3)
/ ( 39,1)( 39,2)$kint( 40,4)( 40,3)
/ ( 40,1)( 40,2)$kint( 41,4)( 41,3)
/ ( 41,1)( 41,2)$kint( 42,4)( 42,3)
/ ( 42,1)( 42,2)$kint( 43,4)( 43,3)
/ ( 43,1)( 43,2)$kint( 44,4)( 44,3)
/ ( 44,1)( 44,2)$kint( 45,4)( 45,3)
/ ( 45,1)( 45,2)$kint( 46,4)( 46,3)
/ ( 46,1)( 46,2)$kint( 47,4)( 47,3)
/ ( 47,1)( 47,2)$kint( 48,4)( 48,3)
/ ( 48,1)( 48,2)$kint( 49,4)( 49,3)
/ ( 49,1)( 49,2)$kint( 50,4)( 50,3)
/ ( 50,1)( 50,2)$kint( 51,4)( 51,3)
!line  4!
/ ( 35,2)( 35,3)$kint( 52,1)( 52,4)
/ ( 52,1)( 52,2)$kint( 53,4)( 53,3)
/ ( 53,1)( 53,2)$kint( 54,4)( 54,3)
/ ( 54,1)( 54,2)$kint( 55,4)( 55,3)
/ ( 55,1)( 55,2)$kint( 56,4)( 56,3)
/ ( 56,1)( 56,2)$kint( 57,4)( 57,3)
/ ( 57,1)( 57,2)$kint( 58,4)( 58,3)
/ ( 58,1)( 58,2)$kint( 59,4)( 59,3)
/ ( 59,1)( 59,2)$kint( 60,4)( 60,3)
/ ( 60,1)( 60,2)$kint( 61,4)( 61,3)
/ ( 61,1)( 61,2)$kint( 62,4)( 62,3)
/ ( 62,1)( 62,2)$kint( 63,4)( 63,3)
/ ( 63,1)( 63,2)$kint( 64,4)( 64,3)
/ ( 64,1)( 64,2)$kint( 65,4)( 65,3)
```

```
/ ( 65,1)( 65,2)$kint( 66,4)( 66,3)
/ ( 66,1)( 66,2)$kint( 67,4)( 67,3)
/ ( 67,1)( 67,2)$kint( 68,4)( 68,3)
!line  5!
/ ( 52,2)( 52,3)$kint( 69,1)( 69,4)
/ ( 69,1)( 69,2)$kint( 70,4)( 70,3)
/ ( 70,1)( 70,2)$kint( 71,4)( 71,3)
/ ( 71,1)( 71,2)$kint( 72,4)( 72,3)
/ ( 72,1)( 72,2)$kint( 73,4)( 73,3)
/ ( 73,1)( 73,2)$kint( 74,4)( 74,3)
/ ( 74,1)( 74,2)$kint( 75,4)( 75,3)
/ ( 75,1)( 75,2)$kint( 76,4)( 76,3)
/ ( 76,1)( 76,2)$kint( 77,4)( 77,3)
/ ( 77,1)( 77,2)$kint( 78,4)( 78,3)
/ ( 78,1)( 78,2)$kint( 79,4)( 79,3)
/ ( 79,1)( 79,2)$kint( 80,4)( 80,3)
/ ( 80,1)( 80,2)$kint( 81,4)( 81,3)
/ ( 81,1)( 81,2)$kint( 82,4)( 82,3)
/ ( 82,1)( 82,2)$kint( 83,4)( 83,3)
/ ( 83,1)( 83,2)$kint( 84,4)( 84,3)
/ ( 84,1)( 84,2)$kint( 85,4)( 85,3)
!line  6!
/ ( 69,2)( 69,3)$kint( 86,1)( 86,4)
/ ( 86,1)( 86,2)$kint( 87,4)( 87,3)
/ ( 87,1)( 87,2)$kint( 88,4)( 88,3)
/ ( 88,1)( 88,2)$kint( 89,4)( 89,3)
/ ( 89,1)( 89,2)$kint( 90,4)( 90,3)
/ ( 90,1)( 90,2)$kint( 91,4)( 91,3)
/ ( 91,1)( 91,2)$kint( 92,4)( 92,3)
/ ( 92,1)( 92,2)$kint( 93,4)( 93,3)
/ ( 93,1)( 93,2)$kint( 94,4)( 94,3)
/ ( 94,1)( 94,2)$kint( 95,4)( 95,3)
/ ( 95,1)( 95,2)$kint( 96,4)( 96,3)
/ ( 96,1)( 96,2)$kint( 97,4)( 97,3)
/ ( 97,1)( 97,2)$kint( 98,4)( 98,3)
/ ( 98,1)( 98,2)$kint( 99,4)( 99,3)
/ ( 99,1)( 99,2)$kint(100,4)(100,3)
/ (100,1)(100,2)$kint(101,4)(101,3)
/ (101,1)(101,2)$kint(102,4)(102,3)
!line  7!
/ ( 86,2)( 86,3)$kint(103,1)(103,4)
/ (103,1)(103,2)$kint(104,4)(104,3)
/ (104,1)(104,2)$kint(105,4)(105,3)
/ (105,1)(105,2)$kint(106,4)(106,3)
/ (106,1)(106,2)$kint(107,4)(107,3)
/ (107,1)(107,2)$kint(108,4)(108,3)
/ (108,1)(108,2)$kint(109,4)(109,3)
/ (109,1)(109,2)$kint(110,4)(110,3)
/ (110,1)(110,2)$kint(111,4)(111,3)
/ (111,1)(111,2)$kint(112,4)(112,3)
/ (112,1)(112,2)$kint(113,4)(113,3)
/ (113,1)(113,2)$kint(114,4)(114,3)
/ (114,1)(114,2)$kint(115,4)(115,3)
/ (115,1)(115,2)$kint(116,4)(116,3)
```

```
/ (116,1)(116,2)$kint(117,4)(117,3)
/ (117,1)(117,2)$kint(118,4)(118,3)
/ (118,1)(118,2)$kint(119,4)(119,3)
!line  8!
/ (103,2)(103,3)$kint(120,1)(120,4)
/ (120,1)(120,2)$kint(121,4)(121,3)
/ (121,1)(121,2)$kint(122,4)(122,3)
/ (122,1)(122,2)$kint(123,4)(123,3)
/ (123,1)(123,2)$kint(124,4)(124,3)
/ (124,1)(124,2)$kint(125,4)(125,3)
/ (125,1)(125,2)$kint(126,4)(126,3)
/ (126,1)(126,2)$kint(127,4)(127,3)
/ (127,1)(127,2)$kint(128,4)(128,3)
/ (128,1)(128,2)$kint(129,4)(129,3)
/ (129,1)(129,2)$kint(130,4)(130,3)
/ (130,1)(130,2)$kint(131,4)(131,3)
/ (131,1)(131,2)$kint(132,4)(132,3)
/ (132,1)(132,2)$kint(133,4)(133,3)
/ (133,1)(133,2)$kint(134,4)(134,3)
/ (134,1)(134,2)$kint(135,4)(135,3)
/ (135,1)(135,2)$kint(136,4)(136,3)
!line  9!
/ (120,2)(120,3)$kint(137,1)(137,4)
/ (137,1)(137,2)$kint(138,4)(138,3)
/ (138,1)(138,2)$kint(139,4)(139,3)
/ (139,1)(139,2)$kint(140,4)(140,3)
/ (140,1)(140,2)$kint(141,4)(141,3)
/ (141,1)(141,2)$kint(142,4)(142,3)
/ (142,1)(142,2)$kint(143,4)(143,3)
/ (143,1)(143,2)$kint(144,4)(144,3)
/ (144,1)(144,2)$kint(145,4)(145,3)
/ (145,1)(145,2)$kint(146,4)(146,3)
/ (146,1)(146,2)$kint(147,4)(147,3)
/ (147,1)(147,2)$kint(148,4)(148,3)
/ (148,1)(148,2)$kint(149,4)(149,3)
/ (149,1)(149,2)$kint(150,4)(150,3)
/ (150,1)(150,2)$kint(151,4)(151,3)
/ (151,1)(151,2)$kint(152,4)(152,3)
/ (152,1)(152,2)$kint(153,4)(153,3)
!line 10!
/ (137,2)(137,3)$kint(154,1)(154,4)
/ (154,1)(154,2)$kint(155,4)(155,3)
/ (155,1)(155,2)$kint(156,4)(156,3)
/ (156,1)(156,2)$kint(157,4)(157,3)
/ (157,1)(157,2)$kint(158,4)(158,3)
/ (158,1)(158,2)$kint(159,4)(159,3)
/ (159,1)(159,2)$kint(160,4)(160,3)
/ (160,1)(160,2)$kint(161,4)(161,3)
/ (161,1)(161,2)$kint(162,4)(162,3)
/ (162,1)(162,2)$kint(163,4)(163,3)
/ (163,1)(163,2)$kint(164,4)(164,3)
/ (164,1)(164,2)$kint(165,4)(165,3)
/ (165,1)(165,2)$kint(166,4)(166,3)
/ (166,1)(166,2)$kint(167,4)(167,3)
```

```
/ (167,1)(167,2)$kint(168,4)(168,3)
/ (168,1)(168,2)$kint(169,4)(169,3)
/ (169,1)(169,2)$kint(170,4)(170,3)
!line 11!
/ (154,2)(154,3)$kint(171,1)(171,4)
/ (171,1)(171,2)$kint(172,4)(172,3)
/ (172,1)(172,2)$kint(173,4)(173,3)
/ (173,1)(173,2)$kint(174,4)(174,3)
/ (174,1)(174,2)$kint(175,4)(175,3)
/ (175,1)(175,2)$kint(176,4)(176,3)
/ (176,1)(176,2)$kint(177,4)(177,3)
/ (177,1)(177,2)$kint(178,4)(178,3)
/ (178,1)(178,2)$kint(179,4)(179,3)
/ (179,1)(179,2)$kint(180,4)(180,3)
/ (180,1)(180,2)$kint(181,4)(181,3)
/ (181,1)(181,2)$kint(182,4)(182,3)
/ (182,1)(182,2)$kint(183,4)(183,3)
/ (183,1)(183,2)$kint(184,4)(184,3)
/ (184,1)(184,2)$kint(185,4)(185,3)
/ (185,1)(185,2)$kint(186,4)(186,3)
/ (186,1)(186,2)$kint(187,4)(187,3)
!line 12!
/ (171,2)(171,3)$kint(188,1)(188,4)
/ (188,1)(188,2)$kint(189,4)(189,3)
/ (189,1)(189,2)$kint(190,4)(190,3)
/ (190,1)(190,2)$kint(191,4)(191,3)
/ (191,1)(191,2)$kint(192,4)(192,3)
/ (192,1)(192,2)$kint(193,4)(193,3)
/ (193,1)(193,2)$kint(194,4)(194,3)
/ (194,1)(194,2)$kint(195,4)(195,3)
/ (195,1)(195,2)$kint(196,4)(196,3)
/ (196,1)(196,2)$kint(197,4)(197,3)
/ (197,1)(197,2)$kint(198,4)(198,3)
/ (198,1)(198,2)$kint(199,4)(199,3)
/ (199,1)(199,2)$kint(200,4)(200,3)
/ (200,1)(200,2)$kint(201,4)(201,3)
/ (201,1)(201,2)$kint(202,4)(202,3)
/ (202,1)(202,2)$kint(203,4)(203,3)
/ (203,1)(203,2)$kint(204,4)(204,3)
!line 13!
/ (188,2)(188,3)$kint(205,1)(205,4)
/ (205,1)(205,2)$kint(206,4)(206,3)
/ (206,1)(206,2)$kint(207,4)(207,3)
/ (207,1)(207,2)$kint(208,4)(208,3)
/ (208,1)(208,2)$kint(209,4)(209,3)
/ (209,1)(209,2)$kint(210,4)(210,3)
/ (210,1)(210,2)$kint(211,4)(211,3)
/ (211,1)(211,2)$kint(212,4)(212,3)
/ (212,1)(212,2)$kint(213,4)(213,3)
/ (213,1)(213,2)$kint(214,4)(214,3)
/ (214,1)(214,2)$kint(215,4)(215,3)
/ (215,1)(215,2)$kint(216,4)(216,3)
/ (216,1)(216,2)$kint(217,4)(217,3)
/ (217,1)(217,2)$kint(218,4)(218,3)
```

```
/ (218,1)(218,2)$kint(219,4)(219,3)
/ (219,1)(219,2)$kint(220,4)(220,3)
/ (220,1)(220,2)$kint(221,4)(221,3)
!line 14!
/ (205,2)(205,3)$kint(222,1)(222,4)
/ (222,1)(222,2)$kint(223,4)(223,3)
/ (223,1)(223,2)$kint(224,4)(224,3)
/ (224,1)(224,2)$kint(225,4)(225,3)
/ (225,1)(225,2)$kint(226,4)(226,3)
/ (226,1)(226,2)$kint(227,4)(227,3)
/ (227,1)(227,2)$kint(228,4)(228,3)
/ (228,1)(228,2)$kint(229,4)(229,3)
/ (229,1)(229,2)$kint(230,4)(230,3)
/ (230,1)(230,2)$kint(231,4)(231,3)
/ (231,1)(231,2)$kint(232,4)(232,3)
/ (232,1)(232,2)$kint(233,4)(233,3)
/ (233,1)(233,2)$kint(234,4)(234,3)
/ (234,1)(234,2)$kint(235,4)(235,3)
/ (235,1)(235,2)$kint(236,4)(236,3)
/ (236,1)(236,2)$kint(237,4)(237,3)
/ (237,1)(237,2)$kint(238,4)(238,3)
!line 15!
/ (222,2)(222,3)$kint(239,1)(239,4)
/ (239,1)(239,2)$kint(240,4)(240,3)
/ (240,1)(240,2)$kint(241,4)(241,3)
/ (241,1)(241,2)$kint(242,4)(242,3)
/ (242,1)(242,2)$kint(243,4)(243,3)
/ (243,1)(243,2)$kint(244,4)(244,3)
/ (244,1)(244,2)$kint(245,4)(245,3)
/ (245,1)(245,2)$kint(246,4)(246,3)
/ (246,1)(246,2)$kint(247,4)(247,3)
/ (247,1)(247,2)$kint(248,4)(248,3)
/ (248,1)(248,2)$kint(249,4)(249,3)
/ (249,1)(249,2)$kint(250,4)(250,3)
/ (250,1)(250,2)$kint(251,4)(251,3)
/ (251,1)(251,2)$kint(252,4)(252,3)
/ (252,1)(252,2)$kint(253,4)(253,3)
/ (253,1)(253,2)$kint(254,4)(254,3)
/ (254,1)(254,2)$kint(255,4)(255,3)
!line 16!
/ (239,2)(239,3)$kint(256,1)(256,4)
/ (256,1)(256,2)$kint(257,4)(257,3)
/ (257,1)(257,2)$kint(258,4)(258,3)
/ (258,1)(258,2)$kint(259,4)(259,3)
/ (259,1)(259,2)$kint(260,4)(260,3)
/ (260,1)(260,2)$kint(261,4)(261,3)
/ (261,1)(261,2)$kint(262,4)(262,3)
/ (262,1)(262,2)$kint(263,4)(263,3)
/ (263,1)(263,2)$kint(264,4)(264,3)
/ (264,1)(264,2)$kint(265,4)(265,3)
/ (265,1)(265,2)$kint(266,4)(266,3)
/ (266,1)(266,2)$kint(267,4)(267,3)
/ (267,1)(267,2)$kint(268,4)(268,3)
/ (268,1)(268,2)$kint(269,4)(269,3)
```

```
/ (269,1)(269,2)$kint(270,4)(270,3)
/ (270,1)(270,2)$kint(271,4)(271,3)
/ (271,1)(271,2)$kint(272,4)(272,3)
!line 17!
/ (256,2)(256,3)$kint(273,1)(273,4)
/ (273,1)(273,2)$kint(274,4)(274,3)
/ (274,1)(274,2)$kint(275,4)(275,3)
/ (275,1)(275,2)$kint(276,4)(276,3)
/ (276,1)(276,2)$kint(277,4)(277,3)
/ (277,1)(277,2)$kint(278,4)(278,3)
/ (278,1)(278,2)$kint(279,4)(279,3)
/ (279,1)(279,2)$kint(280,4)(280,3)
/ (280,1)(280,2)$kint(281,4)(281,3)
/ (281,1)(281,2)$kint(282,4)(282,3)
/ (282,1)(282,2)$kint(283,4)(283,3)
/ (283,1)(283,2)$kint(284,4)(284,3)
/ (284,1)(284,2)$kint(285,4)(285,3)
/ (285,1)(285,2)$kint(286,4)(286,3)
/ (286,1)(286,2)$kint(287,4)(287,3)
/ (287,1)(287,2)$kint(288,4)(288,3)
/ (288,1)(288,2)$kint(289,4)(289,3)
)
! CR !
$'w20bps_CR' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!TT!!UX!!UX!!BG!!UX!!UX!!BG!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )
! BG !
$'w20bps_BG' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX! 40 !UX!!UX!,43 !UX!!UX!,46 !UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!,55 !UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!,65 !UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!,88 !UX!!UX!!GT!!UX!!UX!,94 !UX!!UX!!GT!!UX!!UX!,100 !UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!,139 !UX!!UX!,142 !UX!!UX!!TT!!UX!!UX!,148 !UX!!UX!,151 !UX!!UX!
```

```
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!,190 !UX!!UX!!GT!!UX!!UX!,196 !UX!!UX!!GT!!UX!!UX!,202 !UX!!UX!

!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!,225 !UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!,235 !UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!,244 !UX!!UX!,247 !UX!!UX!,250 !UX!!UX!!UX!!UX!!UX!

!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )
! UO2 !
$'w20bps_UO2' = PAR(
  1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17
,18 ,19 ,20 ,21 ,22 ,23 ,24 ,25 ,26 ,27 ,28 ,29 ,30 ,31 ,32 ,33 ,34
,35 ,36 ,37 ,38 ,39 !BG!,41 ,42 !BG!,44 ,45 !BG!,47 ,48 ,49 ,50 ,51
,52 ,53 ,54 !BG!,56 ,57 ,58 ,59 ,60 ,61 ,62 ,63 ,64 !BG!,66 ,67 ,68
,69 ,70 ,71 ,72 ,73 ,74 ,75 ,76 ,77 ,78 ,79 ,80 ,81 ,82 ,83 ,84 ,85
,86 ,87 !BG!,89 ,90 !GT!,92 ,93 !BG!,95 ,96 !GT!,98 ,99 !BG!,101 ,102
,103 ,104 ,105 ,106 ,107 ,108 ,109 ,110 ,111 ,112 ,113 ,114 ,115 ,116 ,117
     ,118 ,119
,120 ,121 ,122 ,123 ,124 ,125 ,126 ,127 ,128 ,129 ,130 ,131 ,132 ,133 ,134
     ,135 ,136
,137 ,138 !BG!,140 ,141 !BG!,143 ,144 !TT!,146 ,147 !BG!,149 ,150 !BG!,152
     ,153
,154 ,155 ,156 ,157 ,158 ,159 ,160 ,161 ,162 ,163 ,164 ,165 ,166 ,167 ,168
     ,169 ,170
,171 ,172 ,173 ,174 ,175 ,176 ,177 ,178 ,179 ,180 ,181 ,182 ,183 ,184 ,185
     ,186 ,187
,188 ,189 !BG!,191 ,192 !GT!,194 ,195 !BG!,197 ,198 !GT!,200 ,201 !BG!,203
     ,204
,205 ,206 ,207 ,208 ,209 ,210 ,211 ,212 ,213 ,214 ,215 ,216 ,217 ,218 ,219
     ,220 ,221
,222 ,223 ,224 !BG!,226 ,227 ,228 ,229 ,230 ,231 ,232 ,233 ,234 !BG!,236
     ,237 ,238
,239 ,240 ,241 ,242 ,243 !BG!,245 ,246 !BG!,248 ,249 !BG!,251 ,252 ,253
     ,254 ,255
,256 ,257 ,258 ,259 ,260 ,261 ,262 ,263 ,264 ,265 ,266 ,267 ,268 ,269 ,270
     ,271 ,272
,273 ,274 ,275 ,276 ,277 ,278 ,279 ,280 ,281 ,282 ,283 ,284 ,285 ,286 ,287
     ,288 ,289
 )
! TT !
$'w20bps_TT' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX! 145 !UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
```

```
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!!UX!!GT!!UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )
! GT !
$'w20bps_GT' = PAR(
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX! 91 !UX!!UX!!BG!!UX!!UX!,97 !UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!TT!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!BG!!UX!!UX!,193 !UX!!UX!!BG!!UX!!UX!,199 !UX!!UX!!BG!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!BG!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!!UX!
 )
```

A.1.6 Burnup cases for assembly type 1

cases_CRD.SET

```
+SET

!  input required to run different cases !
 &ADD      = SET ( 'Ringhals.SETS' / &'cases_CRD' )

! water material overlays !

'OVLMwater_0B'   = OVLM( 'water_0B'    / * - 0 - modr ,
                                         * - * - modr ,
                                         * -CRpin -crd)
'OVLMwater_400B' = OVLM( 'water_400B'  / * - 0 - modr ,
                                         * - * - modr ,
                                         * -CRpin -crd)
'OVLMwater_800B' = OVLM( 'water_800B'  / * - 0 - modr ,
                                         * - * - modr ,
                                         * -CRpin -crd)
'OVLMwater_1400B'= OVLM( 'water_1400B' / * - 0 - modr ,
                                         * - * - modr ,
                                         * -CRpin -crd)

! cotrol rod material overlay !
```

```
'OVLMcrd'      = OVLM( 'CRD'         / * -CRpin -crd)

! combined material overlays !

!0 Boron exposure case !


'material_0B'  = OVSM( pinovlm,
                       generalovlm,
                      'OVLMwater_0B')

'stateFP_0B'  = STAT('material_0B' , density, temperatureZP, "38.24" )

'Burnup_0B' = PATH(/('stateFP_0B'), 1000/5
                                  , 10000/9
                                  , 50000/20)

!400 Boron exposure case !

'material_400B'  = OVSM( 'material_0B' / 'OVLMwater_400B')

'stateFP_400B'  = STAT('material_400B' , density, temperatureZP, "38.24" )

'Burnup_400B' = PATH(/('stateFP_400B'), 1000/5
                                      , 10000/9
                                      , 50000/20)

!800 Boron exposure case !

'material_800B'  = OVSM( 'material_0B' / 'OVLMwater_800B')

'stateFP_800B'  = STAT('material_800B' , density, temperatureZP, "38.24" )

'Burnup_800B' = PATH(/('stateFP_800B'), 1000/5
                                      , 10000/9
                                      , 50000/20)

!1400 Boron exposure case !
'material_1400B'  = OVSM( 'material_0B' / 'OVLMwater_1400B')

'stateFP_1400B'  = STAT('material_1400B' , density, temperatureZP, "38.24" )

'Burnup_1400B' = PATH(/('stateFP_1400B'), 1000/5
                                        , 10000/9
                                        , 50000/20)

! 0 Boron !

'stateZP_0B'    = STAT('material_0B'   , density, temperatureZP, "0" )

'BOL_0B'    = TREE( 'Burnup_0B' / ('stateZP_400B')    / 0, 50000/25 )

! 400 Boron !
```

```
'stateZP_400B'  = STAT('material_400B' , density, temperatureZP, "0" )

'BOL_400B'  = TREE( 'Burnup_400B' / ('stateZP_400B')  / 0, 50000/25)

! 800 Boron !

'stateZP_800B'  = STAT('material_800B' , density, temperatureZP, "0" )

'BOL_800B'  = TREE( 'Burnup_800B' / ('stateZP_400B')  / 0, 50000/25)

! 1400 Boron !

'stateZP_1400B' = STAT('material_1400B', density, temperatureZP, "0" )

'BOL_1400B' = TREE( 'Burnup_1400B' / ('stateZP_400B') / 0, 50000/25)
```

A.1.7 Assembly type 5

Files needed:

- w260-20bps.inp - main input file
- general.SET - general input data
- mat.SET - material properties
- pinstr.SET - pin structure
- w00.SET - assembly structure
- cases.SET - burnup cases to be calculated
- Ringhals.SETS - file containing data from set files
- assemblies.SETS - file containing assembly data from set files

w260-20bps.SET

```
+HEL

'w260-20bps' = CASE( $library47 / 'Hw260-20bps.hrf' /
                     'Single UO2 PWR assembly',
                     'w260-20bps' )

 &general      = SET( '..\SETS\Ringhals.SETS' / )
 &mat          = SET( '..\SETS\Ringhals.SETS' / )
 &pinstr       = SET( '..\SETS\Ringhals.SETS' / )

 &w00          = SET( '..\assemblies\assemblies.SETS' / )

 $kint   = PAR(8)
 $kpin   = PAR(8)
```

```
!boundary condition!
 CNXw20bps  = BDRY( (1,4,4)$kint(0) )

!pin material overlay!
 pinovlm = OVLM( 'w260'  / ($'w20bps_UO2') - * - fuel )

!general material overlay!
 generalovlm = OVLM( 'GTSS-304'    / ($'w20bps_TT') - * - tube /
                     'FuelZrClad'  / ($'w20bps_UO2') - * - clad/
                     'ZrClad'      / ($'w20bps_GT') - * - clad ,
                                     ($'w20bps_TT') - * - clad /
                     'BGiSS-304'   / ($'w20bps_BG') - * - tubei /
                      BorGlass     / ($'w20bps_BG') - * - glass /
                     'BGoSS-304'   / ($'w20bps_BG') - * - tubeo /
                     'spacerMat'   / ($'w20bps_UO2') - * - spacer )

'waterDens_ZP' = OVLD( $'waterDensity_ZP' / * - 0 - modr ,
                                            * - * - modr )

 ovltZP  = OVLT( "557.06" / *-** )

 generalovld = OVLD( 1.0        / *-** )

 temperatureZP = OVST( ovltZP  )

 density     = OVSD( generalovld , 'waterDens_ZP' )

&'cases'        = SET( '..\SETS\Ringhals.SETS' / )

 Cells = AREA( *-<**> )           ! average of each cell !
 Ass   = AREA( <*-**> )

 AssData = MACRO( ng1, Ass / bu, tr, ab, fi, nf, kf, ch, p0, p1 )

 CellData1 = MACRO(ng1, Cells / bu, tr, ab, fi, nf, kf, ch, p0, p1 )
 CellData2 = MACRO(ng2, Cells / tr, ab, fi, nf, kf, ch, p0, p1 )
 CellDatam1 = MICRO(ng1, Cells /92235 / )

 CellData3 = MACRO(ng3, Cells / tr, ab, fi, nf, kf, ch, p0, p1 )

 CellData5 = MACRO(ng5, Cells / tr, ab, fi, nf, kf, ch, p0, p1 )

'w260-20bps'  = RUN(OUT:1
            /MT:            ! opt1 !
               , 2          ! opt2, use zero-buckling !
               ,            ! opt2 !
            /OM:            ! output lines per page !
               ,            ! dimensioning safety factor !
               , 30 000 000 ! MMRY!
               ,            !CMMRY!
               ,200 000 000 !SMMRY!
               )
```

A.1.8 Burnup cases for assembly type 5

cases.SET

```
+SET

!  input required to run different cases !
 &ADD      = SET ( 'Ringhals.SETS' / &cases )


! water material overlays !

'OVLMwater_0B'   = OVLM( 'water_0B'    /  * - 0 - modr ,
                                          * - * - modr )
'OVLMwater_400B' = OVLM( 'water_400B'  /  * - 0 - modr ,
                                          * - * - modr )
'OVLMwater_800B' = OVLM( 'water_800B'  /  * - 0 - modr ,
                                          * - * - modr )
'OVLMwater_1400B'= OVLM( 'water_1400B' /  * - 0 - modr ,
                                          * - * - modr )


! combined material overlays !


!0 Boron exposure case !

'material_0B'  = OVSM( pinovlm,
                        generalovlm,
                       'OVLMwater_0B')

'stateFP_0B'  = STAT('material_0B' , density, temperatureZP, "38.24" )

'Burnup_0B' = PATH(/('stateFP_0B'), 1000/5
                                  , 10000/9
                                  , 30000/10)


!400 Boron exposure case !

'material_400B'  = OVSM( 'material_0B' / 'OVLMwater_400B')

'stateFP_400B'  = STAT('material_400B' , density, temperatureZP, "38.24" )

'Burnup_400B' = PATH(/('stateFP_400B'), 1000/5
                                      , 10000/9
                                      , 30000/10)


!800 Boron exposure case !

'material_800B'  = OVSM(  'material_0B' / 'OVLMwater_800B')

'stateFP_800B'  = STAT('material_800B' , density, temperatureZP, "38.24" )
```

```
'Burnup_800B' = PATH(/('stateFP_800B'), 1000/5
                                      , 10000/9
                                      , 30000/10)


!1400 Boron exposure case !
'material_1400B'  = OVSM( 'material_0B' / 'OVLMwater_1400B')

'stateFP_1400B'  = STAT('material_1400B' , density, temperatureZP, "38.24" )

'Burnup_1400B' = PATH(/('stateFP_1400B'), 1000/5
                                        , 10000/9
                                        , 30000/10)


! 0 Boron !

'stateZP_0B'    = STAT('material_0B'   , density, temperatureZP, "0" )

'BOL_0B'    = TREE( 'Burnup_0B' / ('stateZP_400B')    / 0, 30000/15 )

! 400 Boron !

'stateZP_400B'  = STAT('material_400B' , density, temperatureZP, "0" )

'BOL_400B'  = TREE( 'Burnup_400B' / ('stateZP_400B')  / 0, 30000/15 )

! 800 Boron !

'stateZP_800B'  = STAT('material_800B' , density, temperatureZP, "0" )

'BOL_800B'  = TREE( 'Burnup_800B' / ('stateZP_400B')  / 0, 30000/15 )

! 1400 Boron !

'stateZP_1400B' = STAT('material_1400B', density, temperatureZP, "0" )

'BOL_1400B' = TREE( 'Burnup_1400B' / ('stateZP_400B') / 0, 30000/15 )
```

A.2 ZENITH input data

Files needed:

- Hw211-00.hrf - file containing results from assembly type 1
- Hw260-20bps.hrf - file containing results from assembly type 5
- Zw211-00_fcm2d_C1.inp - main input for ZENITH for assembly type 1
- Zw260-20bps_fcm2d_C1.inp - main input for ZENITH for assembly type 5
- fcm2dXS.SET - calculation of cross sections
- zenith.SETS - file containing ZENITH set files

The contents in "Zw211-00_fcm2d_C1.inp" and "Zw260-20bps_fcm2d_C1.inp" are identical except that "w211-00" is replaced with "w260-20bps" in the latter.

Zw211-00_fcm2d_C1.inp

```
BEGIN('Differential of cross sections for Ringhals assembly w211-00')

 'w211-00' = IMPORT( HELIOS; 'w211-00' / 'Hw211-00.hrf' )

! Calculational burnup-points to consider !

 $calPath = PAR( 0@2, 2000, 4000, 6000, 8000, 10000,
                 12000, 14000, 16000, 18000, 20000,
                 22000, 24000, 26000, 28000, 30000)
! Xenon equlibrium  !

 $calBranch = PAR( 0@2, 2000@2, 4000@2, 6000@2, 8000@2, 10000@2,
                   12000@2, 14000@2, 16000@2, 18000@2, 20000@2,
                   22000@2, 24000@2, 26000@2, 28000@2, 30000@2)
! Zero Xenon, due to zero power !

 $outfilehrf = PAR( 'Zw211-00_fcm2d_C1.hrf' )
 $outfiletxt = PAR( 'Zw211-00_fcm2d_C1.txt' )

! calculate cross sections !

 $caseTR = PAR( 'w211-00' )
 $caseCM = PAR( 'w211-00' )
 $assG1  = PAR( AssData )
 $cellG1 = PAR( CellData1 )
 $cellG2 = PAR( CellData5 )

 node = ARRAY( R:1.0 )
 NREGX = ARRAY( R:17.0 )
 NREGY = ARRAY( R:17.0 )
 RWIDX = ARRAY( R:17 * 1.26 )
 RWIDY = ARRAY( R:17 * 1.26 )
```

```
 $IDdata = PAR( node;boron;NREGX;NREGY;RWIDX;RWIDY )

! for 0ppm boron case !

 $calTR  = PAR( 'Burnup_0B':('stateFP_0B') $calPath )
 $calCM  = PAR( 'BOL_0B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 0ppm  Boron 4% power')
 boron = ARRAY( R:0.0 )
 $path      = PAR( Boron0 )

 &fcm2dXS = SET( '..\zenith\zenith.SETS' / )

! for 400ppm boron case !

 $calTR  = PAR( 'Burnup_400B':('stateFP_400B') $calPath )
 $calCM  = PAR( 'BOL_400B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 400ppm  Boron 4% power')
 boron = ARRAY( R:400.0 )
 $path      = PAR( Boron400 )

 &fcm2dXS = SET( '..\zenith\zenith.SETS' / )

! for 800ppm boron case !

 $calTR  = PAR( 'Burnup_800B':('stateFP_800B') $calPath )
 $calCM  = PAR( 'BOL_800B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 800ppm  Boron 4% power')
 boron = ARRAY( R:800.0 )
 $path      = PAR( Boron800 )

 &fcm2dXS = SET( '..\zenith\zenith.SETS' / )

! for 1400ppm boron case !

 $calTR  = PAR( 'Burnup_1400B':('stateFP_1400B') $calPath )
 $calCM  = PAR( 'BOL_1400B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 1400ppm  Boron 4% power')
 boron = ARRAY( R:1400.0 )
 $path      = PAR( Boron1400 )

 &fcm2dXS = SET( '..\zenith\zenith.SETS' / )

 $outfiletxt = CON( to: TA / $outfilehrf  /  )

END()
```

A.2.1 Calculation of cross sections

fcm2dXS.SET

```
+SET

 &ADD = SET( 'zenith.SETS' / &'fcm2dXS' )
```

```
! BEGIN DESCRIPTION !
! Requires          !
! $XStitle = title for output                                !
! $caseTR = name of transmutation case, which needs to be open !
! $caseCM = name of comparison case, which needs to be open !
! $calTR  = calculation points to be considered              !
! $calCM  = calculation points to be considered              !
! $assG1  = one group assembly data                          !
! $outfilehrf = name of output hermes file to write to       !
! $outfiletxt = name of output TABGEN ascci file             !
! $IDdata     = identifying data                             !
! $catalogue  = Name of catalogue to write data to           !
! $path       = Name of path to write data to                !
!                                                            !
! data is written catalogue $case in path                    !
! END DESCRIPTION   !

 ab   = SEL( ab / MACRO / $caseCM ; $cellG2   /  $calCM )
 ch   = SEL( ch / MACRO / $caseCM ; $cellG2   /  $calCM )
 kf   = SEL( kf / MACRO / $caseCM ; $cellG2   /  $calCM )
 nf   = SEL( nf / MACRO / $caseCM ; $cellG2   /  $calCM )
 p0   = SEL( p0 / MACRO / $caseCM ; $cellG2   /  $calCM )
 p1   = SEL( p1 / MACRO / $caseCM ; $cellG2   /  $calCM )

pburn = SEL( pburn / MACRO / $caseTR ; $assG1    / $calTR )
keff = SEL( eigv/ MACRO / $caseCM ; $assG1   /  $calCM )

Ecase = FOR( pburn * 1.0E-3 )   ! convert to GWd/tonne !

! get self scatter !
! multiply scatter matrix by !
!    0  1  2                 !
!  G                         !
!  1    1  0                 !
!  2    0  1                 !
!                            !

 maskP = FOR( @NbO( p0 ) - @NbG( p0 ) )
 getSelfScatter = INDEX( maskP / WINDOW: -0.5, 0.5 / 0.0 )
 getDownScatter = INDEX( maskP / WINDOW:     ,-0.5 / 0.0 )
 getUpScatter = INDEX( maskP / WINDOW:  0.5,     / 0.0 )

! reconstructed to matrix form  !
! no need for reconstruction    !
 Pdown = FOR( ( p0 ^ getDownScatter ) ^^ getDownScatter )
 Pself = FOR( p0 ^ getSelfScatter )
 Pup   = FOR( ( p0 ^ getUpScatter ) ^^ getUpScatter )

! generate transport corrected scatter function !
 Pself0 = FOR( ( p0 ^ getSelfScatter ) ^^ getSelfScatter )
 Pself1 = FOR( ( p1 ^ getSelfScatter ) ^^ getSelfScatter )
 PselfTR = FOR( Pself0 - Pself1 )
```

```
! for fcm2d input - including self scatter  !
! scatter matrix includes transport corrected self scatter !
! uncorrected down scatter and up scatter !

 SIGS = FOR( PselfTR + Pdown + Pup )

!Must be divided into smaller arrays!
!Zenith can not convert arrays bigger than 50000 elements!
 si1 = INDEX(SIGS/RANGE: ; ; ; ;1 )
 si2 = INDEX(SIGS/RANGE: ; ; ; ;2 )
 si3 = INDEX(SIGS/RANGE: ; ; ; ;3 )
 si4 = INDEX(SIGS/RANGE: ; ; ; ;4 )
 si5 = INDEX(SIGS/RANGE: ; ; ; ;5 )
 SIGS1 = FOR(SIGS^si1)
 SIGS2 = FOR(SIGS^si2)
 SIGS3 = FOR(SIGS^si3)
 SIGS4 = FOR(SIGS^si4)
 SIGS5 = FOR(SIGS^si5)

 permuteGO = INDEX( p0 / PERMUTE: R I E O G )

!Total cross section!
 SIGT = FOR( ab + @SmO( SIGS ^permuteGO )  )

!Fission energy production!
 SIGF = FOR( kf     )
!Fission neutron production!
 USIG = FOR( nf     )
!Fission spectrum!
 AKAI = FOR( ch     )

! write output to calalouge.  !

 fcm2d = EXP( +: $outfilehrf / $path / $IDdata /
Ecase : keff,
                         SIGT ,
                         SIGS1,
 SIGS2,
 SIGS3,
 SIGS4,
                         SIGS5,
                         SIGF ,
                         USIG ,
                         AKAI )

! $outfiletxt = CON( to: TA / $outfilehrf / $path ; $catalogue ) !

! End of zenith set !
```

A.3 AURORA multi-assembly input

Files needed:

- c1.inp - main input data
- general.SET - general input data
- mat.SET - material properties
- pinstr.SET - pin structure
- w00.SET - assembly properties
- c1.SET - system properties
- cases_c1.SET - burnup cases to be calculated
- Ringhals.SETS - file containing set files
- assemblies.SETS - file containing assembly and system set files

A.3.1 Main input

c1.inp

```
+HEL

'c1' = CASE( $library47 / 'Hwc1.hrf' /
                      'c1 ringhals UO2_211 and UO2_260 PWR assembly',
                      'c1-00-00' )

 &general      = SET( '..\SETS\Ringhals.SETS' / )
 &mat          = SET( '..\SETS\Ringhals.SETS' / )
 &pinstr       = SET( '..\SETS\Ringhals.SETS' / )

 &w00          = SET( '..\assemblies\assemblies.SETS' / )
 &c1           = SET( '..\assemblies\assemblies.SETS' / )

 $kint   = PAR(8)
 $kext   = PAR(4)

!boundary condition!
 CNXc1  = BDRY( (1-1,4,4)$kext(0) )

!pin material overlay!
 pinovlm = OVLM( 'w211'  / CNXw00-($'w00_UO2') - * - fuel /
                 'w260'  / CNXw20bps-($'w20bps_UO2') - * - fuel )

!general material overlay!
 generalovlm = OVLM( 'GTSS-304'    / CNXw00-($'w00_TT') - * - tube,
                                     CNXw20bps-($'w20bps_TT') - * - tube/
                     'FuelZrClad'  / CNXw00-($'w00_UO2') - * - clad,
```

```
                                  CNXw20bps-($'w20bps_UO2') - * - clad/
                    'CRSS-304'    / CNXw00-($'w00_CR') - * - tube/
                    'ZrClad'      / CNXw00-($'w00_CR') - * - clad,
                                    CNXw00-($'w00_TT') - * - clad,
                                    CNXw20bps-($'w20bps_GT') - * - clad,
                                    CNXw20bps-($'w20bps_TT') - * - clad/
                    'BGiSS-304'   / CNXw20bps-($'w20bps_BG') - * - tubei/
                    'BorGlass'    / CNXw20bps-($'w20bps_BG') - * - glass/
                    'BGoSS-304'   / CNXw20bps-($'w20bps_BG') - * - tubeo/
                    'spacerMat'   / CNXw00-($'w00_UO2') - * - spacer,
                                    CNXw20bps-($'w20bps_UO2') - * - spacer)

'waterDens_ZP' = OVLD( $'waterDensity_ZP' / * - * - 0 - modr ,
                                            * - * - * - modr )

 ovltZP  = OVLT( "557.06" / *-*-** )

 generalovld = OVLD( 1.0        / *-*-** )

 temperatureZP = OVST( ovltZP  )

 density     = OVSD( generalovld , 'waterDens_ZP' )

 &'cases_c1'   = SET( '..\SETS\Ringhals.SETS' / )

  'cell_1_1' = AREA( 1-*-<**> )
  'cell_1_2' = AREA( 3-*-<**> )
  'ass_ave' = AREA( <*-*-**> )


  'cellData_1_1_1' = MACRO( ng1,
                    'cell_1_1' / bu,  tr, ab, fi, nf, kf, ch, p0, p1 )

  'cellData_1_2_1' = MACRO( ng1,
                    'cell_1_2' / bu,  tr, ab, fi, nf, kf, ch, p0, p1 )

  'AssData_ave' = MACRO( ng1,
                     'ass_ave' / bu, tr, ab, fi, nf, kf, ch, p0, p1 )

'c1'  = RUN(OUT:1
            /MT:            ! opt1 !
               ,            ! opt2, use zero-buckling !
               ,            ! opt2 !
            /OM:            ! output lines per page !
               ,            ! dimensioning safety factor !
               , 51 200 000 ! MMRY!
               ,            !CMMRY!
               ,300 000 000 !SMMRY!
               )
```

A.3.2 System structure

c1.SET

```
+SET

 &ADD = SET( 'assemblies.SETS' / &c1 )

 CNXc1 = CNX(
! line 1!
 CNXw00, CNXw20bps,
! line 2!
 CNXw20bps, CNXw00
!line  1!
/ (  1-17,1)(  1-289,2)$kext(  2-1,4)(  2-273,3)
!line  2!
/ (  1-289,2)(  1-273,3)$kext(  3-17,1)(  3-1,4)
/ (  3-17,1)(  3-289,2)$kext(  4-1,4)(  4-273,3)
)
```

A.3.3 Burnup cases for the system

cases_c1.SET

```
+SET

 &ADD      = SET ( 'Ringhals.SETS' / &'cases_c1' )

! water material overlays !

'OVLMwater_0B'   = OVLM( 'water_0B'    / * - * - 0 - modr ,
                                         * - * - * - modr ,
                                         CNXw00 - * -CRpin -crd)
'OVLMwater_400B' = OVLM( 'water_400B'  / * - * - 0 - modr ,
                                         * - * - * - modr ,
                                         CNXw00 - * -CRpin -crd)
'OVLMwater_800B' = OVLM( 'water_800B'  / * - * - 0 - modr ,
                                         * - * - * - modr ,
                                         CNXw00 - * -CRpin -crd)
'OVLMwater_1400B'= OVLM( 'water_1400B' / * - * - 0 - modr ,
                                         * - * - * - modr ,
                                         CNXw00 - * -CRpin -crd)

! cotrol rod material overlay !

'OVLMcrd'      = OVLM( 'CRD'         / CNXw00 - * -CRpin -crd)

! combined material overlays !

!0 Boron exposure case !

'material_0B'  = OVSM( pinovlm,
                       generalovlm,
                      'OVLMwater_0B')
```

```
'stateFP_0B'  = STAT('material_0B' , density, temperatureZP, "38.24" )

'Burnup_0B' = PATH(/('stateFP_0B'), 1000/5
                                  , 10000/9
                                  , 30000/10)

!400 Boron exposure case !

'material_400B'  = OVSM( 'material_0B' / 'OVLMwater_400B')

'stateFP_400B'  = STAT('material_400B' , density, temperatureZP, "38.24" )

'Burnup_400B' = PATH(/('stateFP_400B'), 1000/5
                                      , 10000/9
                                      , 30000/10)

!800 Boron exposure case !

'material_800B'  = OVSM( 'material_0B' / 'OVLMwater_800B')

'stateFP_800B'  = STAT('material_800B' , density, temperatureZP, "38.24" )

'Burnup_800B' = PATH(/('stateFP_800B'), 1000/5
                                      , 10000/9
                                      , 30000/10)

!1400 Boron exposure case !
'material_1400B'  = OVSM( 'material_0B' / 'OVLMwater_1400B')

'stateFP_1400B'  = STAT('material_1400B' , density, temperatureZP, "38.24" )

'Burnup_1400B' = PATH(/('stateFP_1400B'), 1000/5
                                        , 10000/9
                                        , 30000/10)

! 0 Boron !

'stateZP_0B'    = STAT('material_0B'   , density, temperatureZP, "0" )

'BOL_0B'    = TREE( 'Burnup_0B' / ('stateZP_400B')    / 0, 30000/15 )

! 400 Boron !

'stateZP_400B'  = STAT('material_400B' , density, temperatureZP, "0" )

'BOL_400B'  = TREE( 'Burnup_400B' / ('stateZP_400B')  / 0, 30000/15)

! 800 Boron !

'stateZP_800B'  = STAT('material_800B' , density, temperatureZP, "0" )

'BOL_800B'  = TREE( 'Burnup_800B' / ('stateZP_400B')  / 0, 30000/15)
```

```
! 1400 Boron !

'stateZP_1400B' = STAT('material_1400B', density, temperatureZP, "0" )

'BOL_1400B' = TREE( 'Burnup_1400B' / ('stateZP_400B') / 0, 30000/15)
```

A.4 ZENITH multi-assembly input

Files needed:

- Hwc1.hrf - file containing results from multi-assembly calculation
- ZwC1.inp - input data for ZENITH for multi-assembly calculation
- pinpow.SET - file calculating pin powers
- zenith.SETS - file containing ZENITH set files

ZwC1.SET

```
BEGIN('Pin powers for Ringhals assemblies c1')

 'c1' = IMPORT( HELIOS; 'c1' / 'Hwc1.hrf' )

! calculational burnup-points to consider !

 $calPath = PAR( 0@2, 2000, 4000, 6000, 8000, 10000,
                 12000, 14000, 16000, 18000, 20000,
                 22000, 24000, 26000, 28000, 30000)
! Xenon equlibrium  !

 $calBranch = PAR( 0@2, 2000@2, 4000@2, 6000@2, 8000@2, 10000@2,
                   12000@2, 14000@2, 16000@2, 18000@2, 20000@2,
                   22000@2, 24000@2, 26000@2, 28000@2, 30000@2)
! Zero Xenon, due to zero power !

 $outfilehrf = PAR( 'ZwC1.hrf' )
 $outfiletxt = PAR( 'ZwC1.txt' )

! Calculate pin powers !

 $caseTR = PAR( 'c1' )
 $caseCM = PAR( 'c1' )
 $assG1  = PAR( 'AssData_ave' )
 $cellG11 = PAR( 'cellData_1_1_1' )
 $cellG12 = PAR( 'cellData_1_2_1' )

 node = ARRAY( R:1.0 )
 NREGX = ARRAY( R:17.0 )
 NREGY = ARRAY( R:17.0 )
 RWIDX = ARRAY( R:17 * 1.26 )
 RWIDY = ARRAY( R:17 * 1.26 )
```

```
 $IDdata = PAR( node;boron;NREGX;NREGY;RWIDX;RWIDY )

! for 0ppm boron case !

 $calTR  = PAR( 'Burnup_0B':('stateFP_0B') $calPath )
 $calCM  = PAR( 'BOL_0B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 0ppm  Boron 4% power')
 boron = ARRAY( R:0.0 )
 $path      = PAR( Boron0 )

 &pinpow = SET( '..\zenith\zenith.SETS' / )

! for 400ppm boron case !

 $calTR  = PAR( 'Burnup_400B':('stateFP_400B') $calPath )
 $calCM  = PAR( 'BOL_400B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 400ppm  Boron 4% power')
 boron = ARRAY( R:400.0 )
 $path      = PAR( Boron400 )

 &pinpow = SET( '..\zenith\zenith.SETS' / )


! for 800ppm boron case !

 $calTR  = PAR( 'Burnup_800B':('stateFP_800B') $calPath )
 $calCM  = PAR( 'BOL_800B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 800ppm  Boron 4% power')
 boron = ARRAY( R:800.0 )
 $path      = PAR( Boron800 )

 &pinpow = SET( '..\zenith\zenith.SETS' / )

! for 1400ppm boron case !

 $calTR  = PAR( 'Burnup_1400B':('stateFP_1400B') $calPath )
 $calCM  = PAR( 'BOL_1400B':('stateZP_400B') $calBranch )
 $XStitle = PAR( 'XS output G5 for w211-00 1400ppm  Boron 4% power')
 boron = ARRAY( R:1400.0 )
 $path      = PAR( Boron1400 )

 &pinpow = SET( '..\zenith\zenith.SETS' / )

 $outfiletxt = CON( to: TA / $outfilehrf  /  )

END()
```

A.4.1 Calculation of pinpowers

pinpow.SET

```
+SET
```

```
 &ADD = SET( 'zenith.SETS' / &pinpow )

! BEGIN DESCRIPTION !
! requires          !
! $caseTR = name of transmutation case, which needs to be open !
! $caseCM = name of comparison case, which needs to be open !
! $calTR  = calculation points to be considered              !
! $calCM  = calculation points to be considered              !
! $assG1  = one group assembly data                         !
! $cellG11 = five group cell data                            !
! $cellG12 = five group cell data                            !
! $outfilehrf = name of output hermes file to write to     !
! $outfiletxt = name of output TABGEN ascci file           !
! $IDdata     = identifying data                           !
! $path       = Name of path to write data to              !
!                                                          !
! data is written catalogue $case in path                  !
! END DESCRIPTION   !

 keff  = SEL( eigv/ MACRO / $caseCM ; $assG1   /  $calCM )
 vo1   = SEL( vo / MACRO / $caseCM ; $cellG11   /  $calCM )
 vo2   = SEL( vo / MACRO / $caseCM ; $cellG12   /  $calCM )
 kf1   = SEL( kf / MACRO / $caseCM ; $cellG11   /  $calCM )
 kf2   = SEL( kf / MACRO / $caseCM ; $cellG12   /  $calCM )
 fx1   = SEL( fx / MACRO / $caseCM ; $cellG11   /  $calCM )
 fx2   = SEL( fx / MACRO / $caseCM ; $cellG12   /  $calCM )

 pburn = SEL( pburn / MACRO / $caseTR ; $assG1    / $calTR )

 pinpow1 = FOR(kf1 * fx1 * vo1)

 pinpow2 = FOR(kf2 * fx2 * vo2)

 Ecase = FOR( pburn * 1.0E-3 )

! Write output to calalouge !

 fcm2d = EXP( +: $outfilehrf / $path / $IDdata /
Ecase : keff, pinpow1, pinpow2 )

! End of zenith set !
```

Appendix B

VNEM input data

B.1 Selecting cross sections from HELIOS

hermes.exe uses a input file to select the cross sections. Only the first part in one of the input file is included since the rest is obvious. In the following case the input file is from the 2.11% uranium-235 calculations:

hermes-w211-00_C1.inp

```
case1
filename ..\..\HELIOS\w211-00\Zw211-00_fcm2d_C1.txt
catalogue \Boron0\fcm2d\
go /
point
filename w211-00_B0_E0_C1.XS
burnup 0.0
go
point
filename w211-00_B0_E2_C1.XS
burnup 2.0
go
point
filename w211-00_B0_E4_C1.XS
burnup 4.0
go
point
filename w211-00_B0_E6_C1.XS
burnup 6.0
go
point
filename w211-00_B0_E8_C1.XS
burnup 8.0
go
point
filename w211-00_B0_E10_C1.XS
burnup 10.0
go
```

```
case1
filename ..\..\HELIOS\w211-00\Zw211-00_fcm2d_C1.txt
catalogue \Boron400\fcm2d\
go /
point
filename w211-00_B400_E0_C1.XS
burnup 0.0
go
point
filename w211-00_B400_E2_C1.XS
burnup 2.0
go
```

The rest of the input file and the format of the input file for the 2.60 % uranium-235 case is similar. The command "stop" is at the last line of the input file.

B.2 FCM2D input

FCM2D uses these cross section files one by one. The input file for each set of cross sections looks like this:

```
N1 RINGHALS IFT1 B0 E0 5G
$INC commondata.inc
  IAT(1,1) = 1 ! assembly type index
$INC Z:\Bill\VNEM\Ringhals3\hermes\w211-00\w211-00_B0_E0_C1.XS ! this file is from HELIOS
  NMSHX(1,1) = 1 * 17 ! number of mesh spaces in region nrx
  NMSHY(1,1) = 1 * 17 ! number of mesh spaces in region nry
  NEDIT(51) = 0 ! 0: no effect
  $$
afpaa1.bin ! scratch file name
ifpaa1.bin ! scratch file name
IFT1 ! VCOEF2D data file name
```

The IAT and IFT-number is increased by one for each run.

We have one input file containing common input data:

commondata.inc

```
 NGMAX     =   5      ! number of energy groups
 IFAMX     =  10      ! number of angular meshes
 KPOLMX    =  10      ! number of polar meshes
 NYMX      =  50      ! limit to the NY parameter
 ARPR      =   0.25   ! accuracy of the rational approximation
 SPBS      =   0.009  ! limit to the track spacing
 NSMAX     = 100      ! limit to the number of source iterations
 INNMAX    =   1      ! number of inner iterations
 EPS5      =   1.0E-6 ! convergence limit for the source iterations
!
 IMAX = 1      ! number of assemblies in the x-direction of the system
 JMAX = 1      ! number of assemblies in the y-direction of the system
```

```
 IROT(1,1) = 0 ! rotation index, 0: no rotation
 IBOUN = 1 1 1 1 ! boundary condition, 1: reflective
 NEDIT = 0 *100  ! 0: no effect
 NEDIT(11) = 1   ! for VCOEF
 MPEDIT(1,1) = 1 ! save mesh/region power density into "EXCEL.txt"
 MFEDIT(1,1) = 1 ! save mesh/region scalar flux into "EXCEL.txt"
!
 EPSIN     =  1.0E-6    ! convergence criterion for inner iterations
 INTMAX    = 20         ! limit to the number of intermediate iterations
 EPSINT    =  1.0E-6    ! convergence limit to intermediate iterations
 DOMRT  =     0.75 * 50 ! upper limit to the iup-th updated dominance ratio
 NCCHEB =     5         ! minimum number of power method iterations
                        ! before starting Chebyshev accleration
 EPSCHB =     1.0E-1    ! upper limit to the convergence error of the power
                        ! method to starting Chebyshev accleration
 NUDR   =     3         ! minimum number of Chebyshev iterations with a
                        ! fixed dominance ratio
!
 IOMEGA_M = 10      ! limit to the number of omega iterations
 OMEGA_CC =  1.0E-4 ! convergence limit to the omega iterations
 OMEGA_MU =  0.6    ! omega-multiplier
 OMEGA_ST =  0.1    ! upper limit to the convergence error of the intermediate
                    ! iterations to starting omega accleration
 IOMEGA_S =  5      ! minimum number of intermediate iterations before
                    ! starting omega accleration
```

B.3 VCOEF2D input

The input file for each of the VCOEF2D runs looks like this:

```
N RINGHALS w211-00_B0_E0_C1
$INC common.inc
$INC use.inc
!
 ift = 1 !
 $$
```

The ift number is increased by one for each run here too.

Common data for the VCOEF2D runs:

common.inc

```
 lmax = 5 ! order of Legendre polynomials
 ndegmx(1) = 6 6 6 6 6 ! number of Source Expansion Functions (SEF)
 ndegx(1, 1) = 0 1 0 1 2 0 ! degree of Legendre polynomials for SEF
 ndegy(1, 1) = 0 0 1 1 0 2 !
 ndegx(1, 2) = 0 1 0 1 2 0 !
 ndegy(1, 2) = 0 0 1 1 0 2 !
 ndegx(1, 3) = 0 1 0 1 2 0 !
 ndegy(1, 3) = 0 0 1 1 0 2 !
 ndegx(1, 4) = 0 1 0 1 2 0 !
```

```
 ndegy(1, 4) = 0 0 1 1 0 2 !
 ndegx(1, 5) = 0 1 0 1 2 0 !
 ndegy(1, 5) = 0 0 1 1 0 2 !
 mdegmx(1) = 3 3 3 3 3 ! number of Boundary Value Expansion Functions (BVEF)
 mdeg(1, 1) = 0 1 2 ! degree of Legendre polynomials for BVEF
 mdeg(1, 2) = 0 1 2 !
 mdeg(1, 3) = 0 1 2 !
 mdeg(1, 4) = 0 1 2 !
 mdeg(1, 5) = 0 1 2 !
 iusec = 0 * 168 ! * LMCLIM(=6) *MDLIM(=4) *NGLIM(=7) << NB! PARAM.INC
 iuses = 0 *  84 ! * LMSLIM(=3) *MDLIM(=4) *NGLIM(=7) << NB! PARAM.INC
 immx = 17 ! upper limit to index im of mapreg
 jmmx = 17 ! upper limit to index jm of mapreg
mapreg( 1,17) = 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 !
mapreg( 1,16) = 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 !
mapreg( 1,15) = 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 !
mapreg( 1,14) = 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 !
mapreg( 1,13) = 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 !
mapreg( 1,12) = 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 !
mapreg( 1,11) = 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 !
mapreg( 1,10) = 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 !
mapreg( 1, 9) = 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 !
mapreg( 1, 8) = 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 !
mapreg( 1, 7) = 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 !
mapreg( 1, 6) =  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 !
mapreg( 1, 5) =  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85 !
mapreg( 1, 4) =  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 !
mapreg( 1, 3) =  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 !
mapreg( 1, 2) =  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 !
mapreg( 1, 1) =   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17 !
 vcthre = 0.0 ! threshold of setting negleglible VNEM coefficient = 0.0
 iopt( 1) = 1 ! use mapreg
 iopt( 2) = 1 ! MOMENTS.txt
 iopt( 3) = 1 ! concal
 iopt( 4) = 1 ! aabbcal
 iopt( 5) = 0 ! no  radial symmetry
!iopt( 5) = 1 ! PWR radial symmetry
!iopt( 6) = 0 ! not used
!iopt( 7) = 1 ! PNc.txt
!iopt( 8) = 1 ! PCppc.txt
!iopt( 9) = 1 ! PSppc.txt
!iopt(10) = 1 ! PNcv.txt, PNsv.txt
!iopt(11) = 1 ! PCppcv.txt, PCppsv.txt
!iopt(12) = 1 ! PSppcv.txt, PSppsv.txt
!iopt(13) = 1 !   use vcthre
 iopt(13) = 0 ! nouse vcthre
 iopt(14) = 0 ! no SEF correction    <<<< 02
!iopt(14) = 1 ! SEF correction       <<<< 01, 04
!iopt(14) =-1 ! SEF correction
!iopt(16) = 1 ! SEF orthonormalize
!iopt(17) = 1 ! BVEF orthonormalize
 iopt(18) = 0 ! 0: no effect
 iopt(19) = 0 ! 0: no effect
 iopt(20) = 0 ! 0: no effect
```

And more common data:

use.inc

```
!        md ng   00 20 22 40 42 44
 iusec(1, 1, 1) =-1  -1  -1  -1  -1  -1 ! -1: inclusion also in
 iusec(1, 2, 1) =-1  -1  -1  -1  -1  -1 !     current coefficients
 iusec(1, 3, 1) =-1  -1  -1  -1  -1  -1 !
!
 iusec(1, 1, 2) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 2, 2) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 3, 2) =-1  -1  -1  -1  -1  -1 !
!
 iusec(1, 1, 3) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 2, 3) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 3, 3) =-1  -1  -1  -1  -1  -1 !
!
 iusec(1, 1, 4) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 2, 4) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 3, 4) =-1  -1  -1  -1  -1  -1 !
!
 iusec(1, 1, 5) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 2, 5) =-1  -1  -1  -1  -1  -1 !
 iusec(1, 3, 5) =-1  -1  -1  -1  -1  -1 !
```

B.4 VNEM2D input

The IFT number is one unit higher in the 2.60% uranium calculations to avoid that they have the same number as in the 2.11% uranium calculations.

```
 1 RINGHALS c1-00_B0_E0_C1
 IMAX = 2 ! number of assemblies in the x-direction of the system
 JMAX = 2 ! number of assemblies in the y-direction of the system
 IFT(1,2) = 1 2! first row of the system with assembly IFT numbers
 IFT(1,1) = 2 1! second row of the system with assembly IFT numbers
 IBOUN = 1 1 1 1 ! boundary condition, 1: reflective
 NSMAX = 50 ! limit to the number of source iterations
 EPSS  = 1.0E-6 ! convergence limit for the source iterations
 NIMAX = 1 1 1 1 1 ! number of inner iterations for each energy group
 EPSIN = 1.0E-5 ! convergence limit for the inner iterations
 IOPT(21) = 101 ! pin cell power edit option, saved into VEXCEL.txt
 IOPT(22) = 102 !
 IOPT(23) = 103 !
 IOPT(24) = 0   !
 INTMAX   = 80 ! limit to the number of in-group iterations
 EPSINT   = 1.0E-6 ! convergence criterion
 LMAXUSE = 5 ! max order of Legendre polynomials
 $$
..\VCOEF2D_w211\IFT1V.bin
..\VCOEF2D_w260-20bps\IFT2V.bin
$$
```