Eigenvalue calculations in accelarator-driven subcritical systems

Bachelor thesis
Joran de Jong

August 3, 2012

*Supervisors:
Dr. ir. Danny Lathouwers
Dr. Jozsef Kophazi
Delft University of Technology
Faculty of Applied Sciences
Department of Radiation, Radionuclide's & Reactors
Section of Physics of Nuclear Reactors

Abstract

The main goal of this thesis was to investigate how it is possible that using eigenvalue calculations to weight the cross sections for accelerator driven sub-critical systems yields accurate results. In order to investigate this, simulations were performed on simplified models of a PWR and the VENUS-F reactor. It was found that the energy spectrum in the fuel zone, and especially in the high energy range, was almost exactly alike under eigenvalue- and source driven- calculations, causing the cross sections to be weighted correctly.

Contents

1 Introduction

Due to the increasing world population and energy usage, there is a large demand for new technologies of energy production. Already a fair share of the world's energy usage is produced by nuclear power plants, but the long lifetime radioactive waste inherent to this form of energy production is a major disadvantage, as well as complains from the general community about safety. Therefore, nucleair engineers aim to make nuclear power plants safer, and reduce the radioactive waste of these plants. One of these projects is GUINEVERE.

The GUINEVERE project aims to validate a method for the reactivity measurement of an Accelerator Driven sub-critical System (ADS) with a fast neutron spectrum. Therefore, an existing reactor at the SCK.CEN site, called VENUS, has been modified towards an installation with a fast spectrum: VENUS-F [5]. A general method for performing calculations on this set-up is to use eigenvalue calculations to weight the cross sections of the system, yielding very accurate results. This study is on how it is possible that this method, using an inadequate weighting scheme, has such a good performance.

The outline of the thesis is as follows. First, simulations were performed on a model of a thermal water cooled reactor, to study the behavior of eigenvalue calculations. Here the effects of variations in the specifications of the eigenvalue calculation were researched. This was done by investigating the results of calculations performed using cross sections that were weighted using the results of the previous calculations. Also, this system was made sub-critical to investigate the behavior of source calculations, and the effects of different weighting schemes.

After this, a model of the VENUS-F reactor, an accelerator driven sub-critical reactor, was set up. Here, the effects of different weighting calculations on the quality of the final results were examined. The neutron flux was normalized in several ways, and energy spectra for different parts of the reactor were created.

This project was done as a bachelor thesis project for the applied physics bachelor of the Delft University of Technology.

2 Theory

2.1 Nuclear reactions

A nuclear reaction is an event where an atomic nucleus interacts with another nucleus or a subatomic particle. In reactor physics, the most important reactions are the neutron-nucleus interactions. A neutron can interact with a nucleus quite easily, since they do not carry any charge, and hence are not repulsed by the negative charge of the electrons, or the positive charge of the neucleus itself. There are two main types of neutron-nucleus interactions, absorption and scattering.

Absorption is the collective name for all reactions where the neutron is absorbed by the nucleus. Some types of nuclei (like uranium-235) tend to split up in two smaller atoms when the nucleus absorbes a neutron. This reaction is then referred as fission. Actually fission will not only produce two smaller nuclei, but also some new neutrons, that can participate in future nuetron-nucleus interactions. Another type of absorption is called neutron capture. The nucleus will incapsulate the neutron, forming a heavier nucleus. These nuclei will sometimes be unstable, in wich case it will decay to a more stable atom.

Scattering is a more delicate form of neutron-nucleus interaction. The neutron will remain an unbound particle, in contrast to absorption, where the neutron will seize to exist as an individual particle. The direction of the neutron will be altered due to this interaction. Scattering can happen in two ways, elastic and inelastic. In elastic scattering, the kinetic energy is conserved, where in inelastic scattering it is not, wich leads to exitaion of the nucleus.

2.2 cross sections

For any neutron travelling in the proximity of an atomic nucleus there is a probabillity function for each type of interaction to take place. These probabillity functions are depending on how close to the nucleus the path of the neutron actually comes. This probabillity function can be replaced by a area surrounding the nucleus, called a microscopic cross section. So the probability-distribution for an event in the space surrounding the particle is replaced with a small surface, and the event is supposed to happen always for neutrons traveling through this surface, and never for neutrons anywhere outside the surface. For a two-diminsional layer of nuclei, the number of reactions, or reaction rate, is now described by formula 5.

$$R_i = \sigma_i I N_A \tag{1}$$

Here, R_i is the reaction rate for interaction type i, σ_i the microscopic cross section for this interaction, I the intensity of the neutron incident beam, and N_A the area density of nuclei. Since the cross sections reflect event probabillity's, the total cross section is just the sum of the cross section for each type of interaction (Eq. 2).

$$\sigma_t = \sum_i \sigma_i = \sigma_{fiss} + \sigma_{capt} + \sigma_{scatt} + ... \tag{2}$$

Most of the time, the microscopic cross section is unpractical for calculations. The behaviour of a volume of nuclei is interesting, and not the behaviour of a single atom or a two-dimensional slab. For this case, the microscopic cross section is simply multiplied by the density of target nuclei, yielding the macroscopic cross section (Eq. 3).

$$\Sigma_i = N\sigma_i \tag{3}$$

Since N is expressed in [#/cm^3], and σ in [cm^2], the unit of Σ is cm^{-1}. This unit is clearly not consistent with the picture of a two-dimensional plane surrounding the nucleus. It still however, the basis for interaction probabillity's. As an example, for a beam of neutrons traveling through a certain material the remaining intensity is given as:

$$I(x) = I_0 e^{-\Sigma_t x} \tag{4}$$

The macroscopic cross section is depending on a lot of variables, like the type of interaction, the atomic species of the target nuclei, the density of target nuclei and the energy of the neutrons.

2.3 Neutron transport

Since the number of neutrons is extremely large, it is imposibble to compute all the single interactions and particle movements. Instead, a density function expresses the number of neutrons present. This density function, $N(\vec{r},t)$ is the expected number of neutrons per unit volume, at place $\vec{r}$, at time t.

In the same way as done before, the reaction rates can be calculated. In three dimensions, it is customary to talk about the reaction rate density F. The multiplication of the neutron density with the macroscopic cross section needs a multiplication with the neutron speed to get the reaction density.

$$F_i = v\Sigma_i N(\vec{r},t) \tag{5}$$

The multiplication of the neutron density with its speed produces a quantity that is quite intresting of its own. It is called the scalar angular flux ($\phi(\vec{r},t)$), and since the reaction rates are produced just by multiplicating this term with the macroscopic cross section, it is widely used in nuclear reactor analysis.

However, this neutron flux does not contain information on the direction of the neutron velocity. Instead of using the neutron velocity, it is customary to specify the energy E and the direction unit vector $\hat{\Omega}$. The energy is related to the absolute value of the speed in the classical way. By decomposing the angular neutron flux into its energy en direction, the angular neutron flux is obtained. It is denoted as $\phi(\vec{r},t,E,\hat{\Omega})$.

The interpretation of this quantity is that $\phi(\vec{r},t,E,\hat{\Omega})d^3rdEd\hat{\Omega}$ yields the number of neutrons in the volume $d^3\vec{r}$ at $\vec{r}$, having an energy between E and $E+dE$, and moving in directions between $\hat{\Omega}$ and $\hat{\Omega}+d\hat{\Omega}$, at time t. So the angular neutron flux is dependant on seven variables (three for space, one for energy, two for direction, and one for time). A different way to view this is as to say that the angular neutron flux lies in the six-dimensional phase space. The flux is now a value assigned to each point in this space, and this number expresses the neutron density at that place, time, energy and direction.

Neutron transport equation

If there are no nuclei present, the information on how this flux develops is self-included. If there is a certain amount of neutrons with a certain direction and energy at place $\vec{r}$ at time t_0, it is known how this flux will be at $t = t_o + \Delta t$. Yet in real situations, neutron-nuclei interactions tend to disturb this balance. For example, a scattered neutron will get an entire different direction, and perhaps even a different energy. This means that is gets a completely new location in the six-dimensional phase space. Knowing the cross sections for all these interactions, it is possible to acount for these effects.

For a system in order to be steady state, each produced neutron must be cancelled by a destructed neutron. Also, the angular flux must be constant in each point of the phase space. The productions and destructions are determined by the reaction rates for several interactions.

Now it is possible to express a steady state balance equation. Since the angular flux should be constant in every part of the phase space, this equation should hold for every value of $\vec{r},E$ and $\hat{\Omega}$. In this equation there are four terms, wich will be described shortly in the next paragraphs. All these terms are deduced for an arbitrary energy and direction, as the equation should hold for every value of these coordinates in the phase space. The two terms that contribute to neutron losses are called ”destruction terms”.

Collision Loss The first one describes the total amount of neutrons that collide with a nuclues. Any collison involves absorbtion or scattering. Since by definition a scattering collision changes $E,\hat{\Omega}$, every collision amounts to a loss of neutrons from that part of the phase space. The total collision reaction rate density is given by the product of the angular neutron flux and the total cross section. For an arbitrary volume V, this density is integrated over space. (Eq. 6).

$$\int_V F_t d^3r = \int_V \Sigma_t \phi(\vec{r},E,\hat{\Omega})d^3r \tag{6}$$

Leakage loss The second destruction term governs neutron loss due to leakage. For an arbitrary chosen volume V, the leakage into or from this volume can be combined into the net leakage. Here, it is usefull to introduce the angular current density $\vec{j}$, which is simply the angular flux density multiplied with its direction vector $\hat{\Omega}$. Now, the rate at which neutrons leak out of a piece of surface $d\vec{S}$ is the inner product of the angular current density and this piece of surface (Eq. 7)

$$\vec{j}(\vec{r},E,\hat{\Omega})\cdot d\vec{S} = \hat{\Omega}\phi(\vec{r},E,\hat{\Omega})\cdot d\vec{(}S) \tag{7}$$

Using Gauss's theorem the integral over the area of the volume can be replaced with a integral over space (eq: 8). This term can now be used to acount for the neutron loss due to leakage.

$$\oint_S \hat{\Omega}\phi(\vec{r},E,\hat{\Omega})\cdot d\vec{(}S) = \int_V \hat{\Omega}\nabla\phi(\vec{r},E,\hat{\Omega})d^3r \tag{8}$$

In-scatter gain There are also two terms called "production terms". The first term describes the total amount of neutrons that are scattered from other energies and directions. The cross section that governs the scattering is called the scattering cross section, $\Sigma_s(\hat{\Omega}'\rightarrow\hat{\Omega},E'\rightarrow E)$. It describes the chance of neutrons with a certain energy and direction $\hat{\Omega}',E'$, to scatter into $\hat{\Omega},E$,the energy and direction under consideration. The total in-scatter term is determined from neutrons scattering into $\hat{\Omega},E$ from all other energies and directions. Hence this in-scatter reaction rate density is the integral over all $\hat{\Omega}',E'$. As usuall, this density is integrated over an arbitrary volume V, to obtain the in-scatter reaction rate for this volume (eq: 9).

$$\text{in-scatter reaction rate} = \int_V d^3r \int_{4\pi} d\hat{\Omega}' \int_0^\infty dE'\Sigma_s(\hat{\Omega}'\rightarrow\hat{\Omega},E'\rightarrow E)\phi(\vec{r},E,\hat{\Omega} \tag{9}$$

Source gain The second term is called the source term. The source term is governed by two additions: the neutrons produced due to fissions, and sometimes an additional external source of neutrons, that are not produced by the nuclei under calculation. The rate at which neutrons with energy E' have collisons that induces fission is governed by the fission cross section: $\Sigma_f(E')$. The average number of new neutrons created during a fission induced by a neutron at energy E' is denoted by $\nu(E')$. Now, the total number of neutrons created at in a volume V is given by equation 10

$$\text{rate of neutron creation} = \int_V d^3r \int_{4\pi} d\hat{\Omega}' \int_0^\infty dE'\nu(E')\Sigma_f(E')\phi(\vec{r},E,\hat{\Omega}) \tag{10}$$

In order to specify the directions of this newly created neutrons, it is assumed that the neutrons are created isotropically. Also, the energy distribution will be given by the fission spectrum $\chi(E)$. The rate at which neutrons with energy E and direction $\hat{\omega}$ are created at place $\vec{r}$ will now be denoted as $s_f(\vec{r},E,\hat{\Omega})$ (eq:11)

$$\int_V s_f(\vec{r},E,\hat{\Omega})d^3r = \int_V d^3r\frac{\chi E}{4\pi}\int_{4\pi} d\hat{\Omega}' \int_0^\infty dE'\nu(E')\Sigma_f(E')\phi(\vec{r},E,\hat{\Omega}) \tag{11}$$

Sometimes, an external source is applied over the system. This term is not depending on $\phi(\vec{r},E,\hat{\Omega}$ and is denoted with $s_e(\vec{r},E,\hat{\Omega})$

Now, the total rate of change of number of neutrons with energy E and direction $\hat{\Omega}$ in volume V can be expressed, and since the system is steady state, this has to be zero for every $E, \hat{\Omega}$ (eq: 12).

$$\int_V d^3r \Bigg(\int_{4\pi} d\hat{\Omega}' \int_0^\infty dE' \Sigma_s \left(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}\right) \phi \left(\vec{r}, E', \hat{\Omega}'\right) - \hat{\Omega} \cdot \nabla \phi \left(\vec{r}, E, \hat{\Omega}\right) + s_f \left(\vec{r}, E, \hat{\Omega}\right) + s_e \left(\vec{r}, E, \hat{\Omega}\right) - \Sigma_t \left(\vec{r}, E\right) \phi \left(\vec{r}, E, \hat{\Omega}\right) \Bigg) \tag{12}$$

Since the volume V was arbitrary chosen, the fact that the integral over V is zero can only hold if its integrand is zero. Taking into account that the equation must hold for every energy and direction, the desired steady state balance equation is obtained (eq: 13). This equation is extremly important in reactor physics, as it determines the angular flux resulting from the set of cross sections and external sources. It is called the Boltzmann neutron transport equation (NTE).

$$\int_{4\pi} d\hat{\Omega}' \int_0^\infty dE' \Sigma_s \left(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}\right) \phi \left(\vec{r}, E', \hat{\Omega}'\right) + s_f \left(\vec{r}, E, \hat{\Omega}\right) + s_e \left(\vec{r}, E, \hat{\Omega}\right) = \hat{\Omega} \cdot \nabla \phi \left(\vec{r}, E, \hat{\Omega}\right) + \Sigma_t \left(\vec{r}, E\right) \phi \left(\vec{r}, E, \hat{\Omega}\right) \tag{13}$$

2.4 Discretization

Solving this equation analytically is very limited, so usually the only way to approximate it is numerically. In order to accomplish this, each parameter of $(\vec{r}, E, \hat{\Omega})$ has to be discretized. This means that the continuous parametres are replaced with a mesh of values at certain points. Using these numerical techniques, the transport equation can now be used to compute the neutron flux from the cross sections and geometry provided. The next paragrahs will be on how each of this parametres is discretized.

Discrete ordinates approach In order to discretize the direction variable $\hat{\Omega}$, a few options are available. In this research, the discrete ordinates approach is used. This method is a widely used method in the industry of nuclear power production. In this method, the continuous angular direction is replaced by a discrete set of N directions. All functions depending on $\hat{\Omega}$ are split up in N functions, with the directional variable replaced by its corresponding value. For example, $\phi(\vec{r}, E, \hat{\Omega})$ becomes $\phi_n(\vec{r}, \hat{\Omega})$. This way, the continuous neutron transport equation is split into N coupled equations, each for one discrete direction. These equations are called the S_N equations (eq. 14), and N is called the angular quadrature.

$$\hat{\Omega}_n \cdot \nabla \phi_n \left(\vec{r}, E\right) + \Sigma_t \left(\vec{r}, E\right) \phi_n \left(\vec{r}, E\right) = \sum_{n'=1}^{N} w_{n'} \int_0^\infty dE' \Sigma_s \left(\vec{r}, E' \to E, \hat{\Omega}_{n'} \to \hat{\Omega}_n\right) \phi_{n'} \left(\vec{r}, E'\right) + S_n \left(\vec{r}, E\right) \tag{14}$$

As visible the intergral over $d\hat{\omega}$ becomes a summation over n, multiplied with the quarature weights w_n.

Discretization of the energy Also, the energy needs to be discretized. The energy range of neutrons in a nuclear reactor is about 10^{-5} to 10^{7} eV. This energy range is discretized in some small regions, called energy groups, which together span the entire energy range. The variable E becomes a group number denoted with g, with group boundaries denotated with g_l and g_u. Functions depending on E will be integrated over dE from g_l to g_u. For example, the energy discretization of the angular neutron flux is shown in equation 15

$$\phi_n^g(\vec{r}) = \int_{g_l}^{g_u} \phi_n(\vec{r}) dE \tag{15}$$

This way, the neutron transport equation can be discretized even further, reducing the integral over dE to a summ over g. The function depending on energy can be replaced with there respective group functions. This results in equation **??**.

Since lots of processes, such as fission are very dependent on small energy differences, a correct group structure is very important. The most widely used basic group structure is one with 238 groups. This structure does not contain to many groups to enable a computer simulation, yet distinguishes enough differences in the energy to keep a simulation reliable. Simulations with less energy groups although, are much faster, and however just joining some energy groups into bigger energy groups doesn't give accurate results, some methods do exist to make an accurate few energy group calculation. These processes are now as collapsing of the energy-group structure. One of these methods is of major importance to this project, and will be explained later.

Spatial discretization Finally, the three-dimensional space is divided in small pieces, or spatial zones. Since the codes used in this investigation are codes for solving one-dimensional problems, the three-dimensional problem must be replaced with a one-dimensional representation. So the geometry must be approached with a one-dimensional form, like infinite slabs of material stacked one on another. For this research slightly more sophisticated models are used, such as infinitely long cylinders and spheres.

Although these forms are three-dimensional, all properties are only dependent on a one-dimensional parameter, making the problem suitable for the simulation programs used. The $\vec{r}$ variable can be written in spherical coördinates: (r, θ, φ) and traditionally the cosines of the angles are used in calculations, so $cos(\varphi) \rightarrow \mu$ and $cos(\theta) \rightarrow \eta$. Using these coördinates, r, μ and η are all required for cylinders, while using only r and μ is sufficient in slabs and spherical geometries. These one dimensional geometries allow considerable simplifications to be made to the NTE, especially in the leakage term, (see table **??**). All functions depending on $\vec{r}$ are determined solely by the number r_i, and some geometrical properties that correspond to r_i (table **??**) Now, the NTE reduces to a coupled set of one dimensional equations, for

Geometry	Area	Volume	$\hat{\Omega} \cdot \nabla \phi$
Slab	1	$r_{i+1} - r_i$	$\mu \frac{\partial \phi}{\partial r}$
Cylinder	$2\pi r_i$	$\pi(r_{i+1}^2 - r_i^2)$	$\frac{\mu}{r} \frac{\partial (r\phi)}{\partial r} - \frac{1}{r} \frac{\partial (\eta \phi)}{\partial \varphi}$
Sphere	$4\pi r_i^2$	$4/3\pi(r_{i+1}^3 - r_i^3)$	$\frac{\mu}{r^2} \frac{\partial (r^2 \phi)}{r} + \frac{1}{r} \frac{\partial [(1-\mu^2)\phi]}{\mu}$

Table 1: One-dimensional geometrical properties and leakage terms

example the slab equation (Eq. 16). These equations can easily be discretized in r by taking the function values at certain points r_i, and replace the partial derivative by differnce of the values at two adjacent r_i's devided by Δr.

$$\mu \frac{\partial \phi_n^g(r,\mu)}{\partial r} + \Sigma_t^g(r,\mu)\, \phi_n^g(r,\mu) = \sum_{n'=1}^{N} w_{n'} \sum_{g=1}^{G} \Sigma_s\left(r, \mu, g' \rightarrow g, \hat{\Omega}_{n'} \rightarrow \hat{\Omega}_n\right) \phi_{n'}^{g'}(r,\mu) + S_n^g(r,\mu,E) \tag{16}$$

2.5 Eigenvalue calculations

For a steady state solution, the production and destruction terms in the neutron transport equation must be in balance exactly. Basicly, this means that from the neutrons that are created during one fission, exactly one is able to induce another fission. Then the process is constant over time, or the reactor is critical.

Although real reactors are usually operated in a critical state, in reactor analysis this doesn't always have to be so. Calculations can be made on systems that are super- or sub-critical from themselves. This means that respectively the production term is bigger than the destruction term, causing the system to escalate over time, or smaller, causing the reaction to fade away. For these scenarios, the terms in the transport equation do not cancel each other. To keep the equation steady state, either the destruction or the production term has to be changed by a artificial factor. This factor can be found by solving the eigenvalue equation.

The continuous eigenvalue equation can be expressed, yet it is quite hard to evaluate. In this research, all parametres are discretized. This allows the use of the discrete eigenvalue equation, which can be evaluated more easely. The discrete eigenvalue equation is more comprehensive if stated in vector notation.

Vector notation In vector notation, the angular neutron flux will be treated as a vector, $\phi_n^g(r_i) \rightarrow \vec{\phi}$, and the operators as matrices. This usually results in massively huge vectors, since the angular neutron flux for each energy, direction and place will have its own value in the vector. Hence the vector $\vec{\phi}$ will be of the size: number of discrete directions $\times$ number of energy groups $\times$ number of spatial intervals. The operator matrices are square, so the size is the square of this already huge number. Note that the specific arrangment of the units in this vectors can be chosen quite arbitrarly, as long as the ordening of the operator matrices is chosen consistent with it. For example, when calculating the reaction rates for a certain event i, the vector notation is as in eq. 17.

$$\begin{pmatrix} r.r._1 \\ r.r._2 \\ \vdots \\ r.r._M \end{pmatrix} = \begin{bmatrix} \Sigma_i(1\rightarrow 1) & \Sigma_i(1\rightarrow 2) & \cdots & \Sigma_i(1\rightarrow M) \\ \Sigma_i(2\rightarrow 1) & \Sigma_i(2\rightarrow 2) & \cdots & \Sigma_i(2\rightarrow M) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_i(M\rightarrow 1) & \Sigma_i(M\rightarrow 1) & \cdots & \Sigma_i(M\rightarrow M) \end{bmatrix} \cdot \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_M \end{pmatrix} \tag{17}$$

Eigenvalue equation Using vector notation, the total destruction matrix (denoted $\hat{D}$) is simply the summ of the total cross section matrix (including the scatter matrix) and the leakage matrix. The fission reaction rates depends on the angular flux, so it can also be written as an operator matrix, as well as the in scatter reaction rates. The addition of these two matrices will produce the total destruction matrix, denoted with $\hat{P}$.

The external source term is not depending on the flux, and hence can also not be expressed as a operator working on the flux. For this reason, the eigenvalue equation can't take this term into account. This means that eigenvalue equations can't be used while there is a external source.

Using this definitions, the transport equation for non-critical systmes reduces to a much more simple form (eq:18).

$$\hat{P}\vec{\phi} = k \cdot \hat{D}\vec{\phi} \tag{18}$$

Here, k is the factor to acount for the difference in produced versus destructed neutrons. To find k, both sides of this equation are multiplied with $\hat{D}^{-1}$, resulting in the equation 19, called the eigenvalue equation.

$$\hat{D}^{-1}\hat{P}\vec{\phi} = k \cdot \vec{\phi} \tag{19}$$

This means that k is an eigenvalue of the $\hat{D}^{-1}\hat{P}$ matrix. The matrix usually has lots of eigenvalue's, but from a reactor analysis point of view only the largest eigenvalue is important. The largest eigenvalue is called the effective eigenvalue, and is denoted k_{eff}. The eigenvector associated with k_{eff} is considered to be the angular neutron flux.

This eigenvalue, k_{eff}, is an important value, as it states whether the system is sub-critical ($k_{eff} < 1$), critical ($k_{eff} = 1$) or super-critical ($k_{eff} > 1$). So this is one of the ways to make calculations on not-critical system, by virtually altering the ratio of the production and destruction, to keep the system steady state.

2.6 Source Calculations

The eigenvalue method as described is not the only way to keep the system steady state. Another way is to keep the reactor subcritical, and apply an external source to the system. This means that next to the fission and scatter sources, an extra term must be supplied. This term accounts for the difference between the production and destruction term. Of course, such calculations can only be done in sub-critical systems, since adding neutrons to a system that is super-critical by itself would increase the number of neutrons infinitely, yielding no steady state solutions. In this case, the spatial, directional and energy distribution of the source has to be specified. Then computations can use this specified source, and the cross sections for the various materials, to calculate the angular flux.

These methods are not just mathematical interesting, since they are applied in real. Existing experimental reactors use an external source to drive reactions in sub-critical assemblies. Here it is absolutely necessary to use the source-driven method to make simulations about the system, as eigenvalue calculations produce an entire different neutron flux. Though, sometimes eigenvalue calculations are used in some part of the simulation process, with good results. This cannot be explained with the theory used in this study, since it is an entire inconsistent calculation, that doesn't reflect the actual physical properties of the system. Why this methods can be used is a main question in this paper, and will be investigated.

2.7 Cross section weighting

One of the key processes in this project is cross section weighting. This process is basically combining energy groups together. So from a large amount of small energy groups (called fine energy groups), a smaller amount of broad groups is created. This is useful, since less energy groups substantially decrease the computational time. From a complete set of cross sections for all fine energy groups, a new set containing cross sections for the broad groups is created. But just taking the average of the cross sections for the fine groups that together form the broad group is not giving reliable results. Imagine combining two energy groups in one broad group. One of the energy groups has a quite large cross section for a certain process, say fission, where the cross section of the other is rather small for this process. Simply taking the cross section for the broad group as the average of that of the fine groups yields a medium-large cross section for this process. But what if, in the geometry under investigation, almost the entire flux is the first fine group. Those neutrons will dominate the cross section size in the broad group, so it should be quite large.

To account for this phenomenon, cross sections are zone-weighted, as in equation 20. Here, Σ_G is the average cross section in broad group G, for any interaction type. The summations are taken over the fine energy groups that together form the broad group G. Therefore, this weighting proces keeps the reaction rates in each zone equal for the process of collapsing into less energy groups [3].

$$\vec{\Sigma}_G(\vec{r}) \sum_G \phi(E,\vec{r}) = \sum_G \Sigma(E,\vec{r})\phi(E,\vec{r})) \tag{20}$$

In order to perform this weighting, the program must know wat the shape of $\psi(E,\vec{r})$, the angular flux in each energy groups and spatial zone is. This neutron flux is sometimes called the weighting flux. Because the relative intensity of the flux is not equal in all parts of the reactor, the broad group cross sections will be different for each zone. So this weighting calculation provides seperate set's of cross sections for each zone specified. A special form of weighting is called homogenizing. Here, the energy structure stays the same, but the cross sections from several materials that are present will be mixed to produce a single material cross section set. This happens similar to weighting the cross sections for collapsing the energy group structure.

2.8 Codes

For all the calculations descirbed before, specialized codes exist to perform them. These codes are specially written to calculate the results of the discrete neutron transport equation from a set of cross sections. One of the code packages that is frequently used, is SCALE. SCALE is short for Standarized Computer Analyses for Licensing Evaluation. It was developed by the Oak Ridge National Laboratory, on behalf of the U.S. Nuclear Regulatory Commission. Like the name says, the aim was to develop a complete simulation package to perform standardized nucleair safety analysis and design testing. SCALE is now used widly across the world, due to its high reliability. It contains a large number of separate modules, that often interact with each other in effort to solve the neutron transport equation. All these modules have their own specialties in the field of nuclear reactor analysis [2]. Two of these programs were mainly used during this investigation, CSASi and XSDRN. These programs will be discussed here shortly.

CSASi

CSASi is a module specialized in homogenizing the unit cell of a reactor lattice [4]. Its capabillities includes weighting one fuel pin and its surroundings using a certain neutron flux, and create a single material cross section library that represents all the materials in that unit cell. To find the correct weighting function, a complete cross section library must be provided for all the materials specified. Such libraries are called master libraries. They include the fine group cross sections for lots of material, and are available in several forms. In this investigation, a 238 group version of the ENDF library was used. After the homogenizing process is completed, a new single material file will be created, specific for the geometry of the unit cell under investigation.

XSDRN

XSDRN is the SCALE module that is most used in this investigation. It is a transport equation solver for one-dimensional problems, so it can only be applied to spherical or cylindrical problem set-ups. It is program that can perform reactor analysis using two different types of calculations. It can perform transport calculations (based on the boltzmann neutron transport equation as described in 2.3) and diffusion calculation (based on the diffusion equation) [3]. Since the diffusion method yields far less generally applicable results, it was not used in this research, and will not be discussed further. XSDRN solves the NTE using discretization as described in section 2.4 for the parameters of space and energy. It further discretizes the equation by using the discrete ordinates approach (section 2.4) to account for the angular directions. XSDRN is able to perform both eigenvalue as well as source-driven calculations. A great deal of flexibillity is allowed in describing a problem for XSDRN. The number of spatial zones, energy groups, types of nuclides and quadrature order are all arbitrary and can be chosen to best describe the problem on hand. They are however, all limited by computer resources. This is why it is often useful to make calculations with a greater discretization in, lets say, the number of energy groups. In order to perform such calculations, a complete set of cross sections is needed in that amount of energy groups. XSDRN can actually get this set of cross sections from a set of cross sections in a larger number of energy groups. This is done by the process of cross section weighting, as described in section 2.7. The fact that XSDRN can perform both eigenvalue and source-driven calculations, and use the resulting flux to weight cross sections, makes it a very suitable programm for this research.

3 Outline of the thesis

3.1 Aim of the project

As depicted before, the main research topic is a certain weighting scheme, which involves using eigenvalue calculations to weight the cross sections for final source-driven calculations. This weighting scheme seems inconsistent, since the angular flux resulting from a eigenvalue calculation is very different than a flux resulting from a source-driven calculation. So this way the weighted cross sections are those for a eigenvalue calculations, and should not be used to perform final source-driven calculations. Yet this is exactly what is done in the VENUS-F project, and the acquired results are very accurate. The aim of this project is to determine how it is possible to use this inconsistent weighting scheme, and still acquire accurate results.

3.2 Calculation levels

For reactor analysis, usually a single calculation is not enough. Several ”layers” of calculations are used, where the cross section library produced by the old one is used in the new one. These layers are also called levels. For this project, always a three level structure was used.

In the first level, the fuel pins where homogenized. This means that on the smallest scale of the reactor core, the cross sections are weighted to create one single material. This material has the same macroscopic properties as the original fuel pin, and its surrounding unit cell.

In the second level, a calculation was performed, and the result of this calculation was used to weight the cross sections, forming a new cross section library with a decreased amount of energy groups. This calculation will sometimes be referred to as the weighting calculation.

In the third level, the cross sections from the second level calculation were used to perform the final calculation. This final calculation was usually done in a smaller amount of energy groups, depending on the specifications given in the weighting process after the second level calculation. The quality of final calculation could then be compared to a reference: the second level calculation.

3.3 Models

In order to investigate this, two models were used, witch will be further specified later in this section. The first model was that of a thermal system. The model chosen was that of a VVER-440 wich is a russian design power plant. This is a standard pressurized water reactor, and its behaviour is very well known. This was usefull to test the way results depends on the types of calculational methods, weigthing schemes and other specifications. The second model is a resemblence of the actual VENUS-F reactor, wich is the origin of the reseach topic. Is more experimental, and hence its behaviour is not as well known. Yet this model is far more interesting to evaluate the main question in this project, as it is a model of a reactor that is actually driven in both critical as well as accelarator-driven mode.

Thermal system

The model of the thermal system was quite straight forward. In order to simplify the geometry, all the cladding and assembly boundaries were ommited. The reactor core was modelled as containing only two things, fuel pins and water. The large amount of fuel pins was placed in a triangular lattice, forming a rough cylinder with a diameter of two metres. The water was modelled as filling up al the space surrounding the fuel pins. This way only one CSASi calculation to homogenize the unit cell (de space surrounding one fuel pin) had to be performed. The cross sections for the homogenized material could now be used uniformily throughout the core. For the whole-core calculations, XSDRN was used. Now, since XSDRN is able to weight the cross sections and use then for a next calculation, the following things could be examined.

First, the effects of collapsing the energy group structure using an eigenvalue calculation. Since both calculations are being performed in XSDRN, the effect of using less energy groups in a calculations can easily be examined by comparing the results of the third level calculations by those of the second level.

Also, the compensating effect of increasing the amount of weighting zones. This was done by weighting the cross sections in different numbers of zones, while performing the same energy collapsing. This way, these effects could also be investigated by comparing the third with the second level results.

And finally, the effects of different weighting schemes. Here some of the eigenvalue calculations were replaced with source-driven calculations, showing what effects these had on the quality of the final calculation.

VENUS-F geometry

The geometry of the model for the VENUS-F reactor was a little more complicated. The specific properties of the fuel zone (the place in the reactor where the fissionable material is) were quite important for this reseach. Therefore, the cross sections for the homogenied material in these zone were obtained from an earlier study. The actual shape of the reactor core is approximately cylindrical, with a neutron source in the middle. This neutron source is surounded by the fissionable material, which at its place is again surrounded by a lead reflector.

However the actual reactor core is shaped like a cylinder, XSDRN can only handle infinitely long cylinders. So the spherical model was used. The inner sphere was used to resemble the neutron source, and was filled with lead. A spherical shell containing the fissionable material was laid around this, and the final shell was also made of lead, to represent the reflector. More specifics of the model will be provided later. Using this model, it was investigated how accurate results derived from the inconsistent weighting scheme were. After this, the quality of the cross sections were examined, as well as the spectrum of the angular flux in several parts of the reactor.

4 Thermal System

The number of energy groups and spatial zones used affects the quality of simulations. In order to investigate these effects on a eigenvalue calculation, a model of a thermal system was created. This model resembles the geometry of a VVER-440, a Russian designed power plant. VVER is a pressurized water reactor (PWR), and is a reactor very commonly used in powerplants [7]. The reactor core was modeled as a cylinder with a diameter of two meters, consisting of fuel pins placed into a triangular lattice. Although much details were abandoned, it is a good approximation of a real thermal system. The level 1 homogenization of the fuel pins was done in CSASi, using the specifications stated below. For illustration, see figure 1.

- Master library: v7-238
- Lattice: triangular
- Fuel diameter: 0.75 cm
- Fuel: UO_2 3,6% enriched
- Cladding diameter: 0.9 cm
- Cladding: ZR
- Moderator: H_2O
- Total pitch diameter: 1.22 cm

The gap between the fuel and cladding was omitted. The produced material cross sections of this homogenized fuel pins where loaded into XSDRN. In this program, an infinite cylinder was filled with this material. This allowed a one-dimensional calculation in a geometry that represents a VVER. The cylinder's radius was divided in 100 intervals. Each interval was 1 cm long, spanning the center to the outer boundary. The second level 238-group calculation was performed in this geometry. This calculation reduced the number of energy groups and splits up the cylinder into a different number of cylinder shells, or zones. It produced a cross section library with less energy groups and as much materials as zones in the calculation. This cross section library was loaded into XSDRN again, and was used to run a third level eigenvalue calculation.

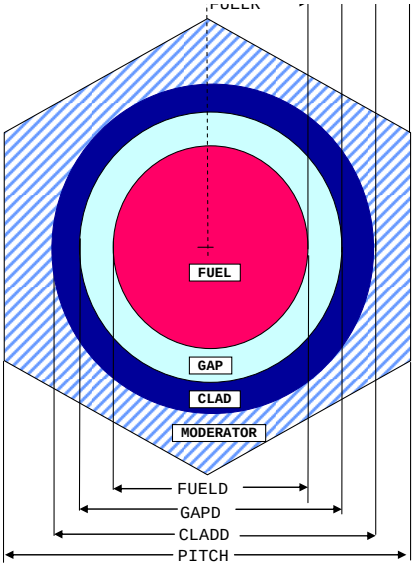

Figure 1: Model of the fuelpin

4.1 Effects of number of energy groups

Reducing the amount of energy groups has a negative effect on the quality of the calculation. Hence the results of third level calculations will be worse than the second level calculations. For this geometry, the second level k_{eff}-value is 1.37110. For several numbers of energy groups, the k_{eff}-value for the third level calculation can be seen in table 2. The k_{eff}-values approach the reference value as the number of energy groups increases, so the system behaves as expected. However, systems with distinctive neutron fluxes can have similar eigen values, so comparing only the k_{eff}-values can lead to false results. The neutron flux shapes must also been taken into account. Figure 2 shows that the neutron flux shapes also perform the expected behavior. From this results, it can be concluded that the quality of the eigenvalue calculation depends on the number of energy groups used as expected. So that it performs worse when a smaller amount of energy groups is used.

Nr. of E-groups	k_{eff}- value
1	1.38016
2	1.37719
4	1.37593
8	1.37311
16	1.37196
32	1.37169
60	1.37148
120	1.37125
238	1.37110

Table 2: The third level k_{eff} -values for different numbers of energy groups.

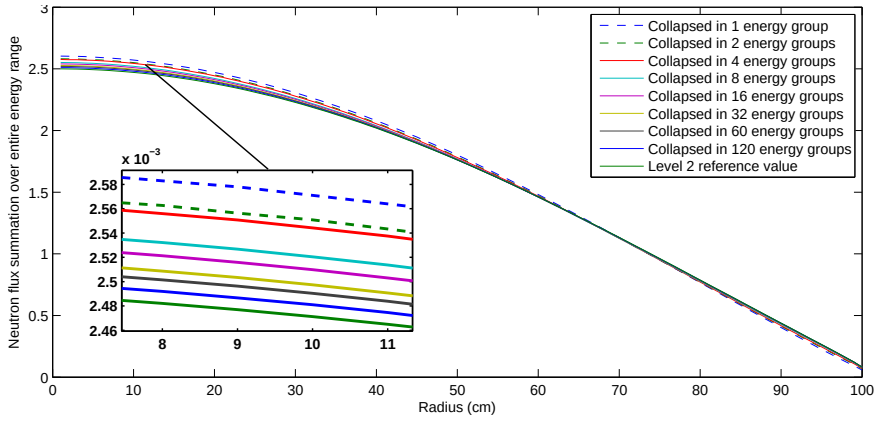

Figure 2: Effects of the number of the energy groups on the shape of the neutron flux in a thermal system

4.2 Effects of number of spatial zones

The previous results where all acquired from eigenvalue calculations using 50 zones. The negative effect of lowering the number of energy groups can be compensated by using more spatial zones. The cross sections are weighted for each spatial zone. So using more spatial zones allows the broad group cross sections to keep the same properties as the fine group cross sections better. Using less than 50 spatial zones should decrease this compensation, and the results should be worse. For the following results, the number of energy groups is fixed at 60. Now the 50 zones results are compared to results of the same calculation, but now with ten or two zones. Especially the calculation with only two zones is very inconsistent. This means that we treat the system as if the neutron flux in the most centered interval is the same as in all the intervals until the half of the radius. According to the theory, the resulting neutron flux should strongly deviate from the origninal flux. However, the k_{eff}-value does not change at all under such a large change in the amount of zones used. Also, the resulting flux shapes are quite similar, as can be seen in figure 3. The difference between the flux shapes is extremely small. Evaluating the difference between the reference flux and the third level fluxes, it becomes visible that the difference between the third level fluxes is even smaller. Also, it becomes visible that the shapes don't approach the reference shape at all as the number of zones increases. The error with the reference shape actually becomes bigger

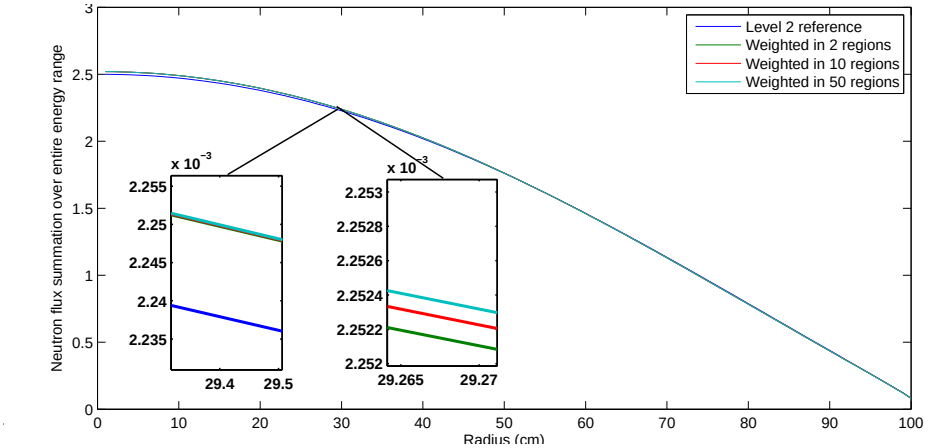

Figure 3: Effects of the number of spatial zones on the shape of the neutron flux in a thermal system, using 60 energy groups.

as the number of zones increases. Although the results of the eigenvalue calculation should depend on the number of zones used, the effects of this number is negligible. The number of energy groups used in the previous case is 60, so a quarter of the orignal 238 groups, wich is large. To check if the same behavior occurs if less energy groups are used, the same calculations were performed using only two energy groups. The resulting flux shapes can be seen in Fig 4. It is visible that the overall off-set is bigger, due to the smaller number of energy groups. The differences between the third level fluxes are still very small. Due to this, they don't compensate for the smaller number of energy groups, as they diverge from the reference flux as the number of zones increases. The conclusion drawn from these results is that this is due to the leakage-term in the NTE. This term is not influenced in the process that weights the cross sections, yet is depending on the energy group structure. If the bias is mainly due to the leakage term, it is quite logical that it will not change when increasing the number of spatial zones. Researchers from the Georgia Institute of Technology recently developed a method to circumvent this effect. This method explicitly accounts for the angular dependence of the cross sections [6].

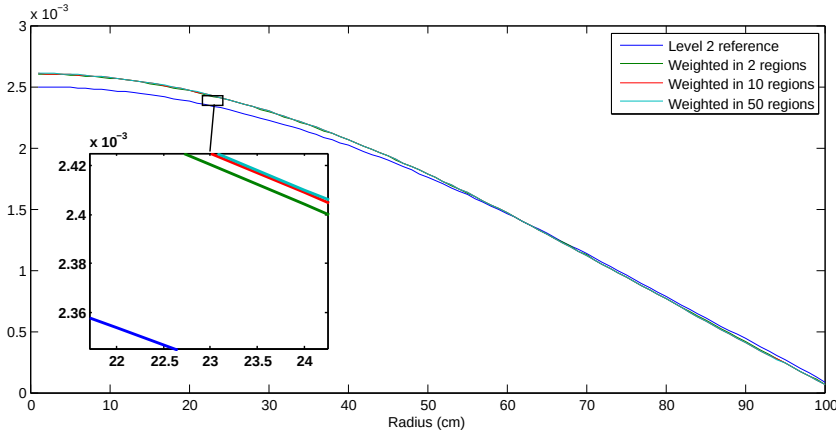

Figure 4: Effects of number of spatial zones on the shape of the neutron flux in a PWR, using two energy groups.

4.3 Infinite system eigenvalue and source-driven calculations

Now the behavior of this system in case of eigenvalue calculations is known, its results can be compared to results that are produced using source-driven calculations. To make the system sub-critical, the enrichment of the uranium was decreased. This was accomplished by making a new homogenized material in CSASi. In order to circumvent the leakage problems, all the spatial dependence was avoided by calculating the system's properties in a homogeneous infinite medium. To get a infinitely large system that is sub-critical, the enrichment had to be lowered even more, to only 0,707%. The resulting system has eigenvalue: k_{eff}=0,9000. Calculations were performd to determine the effects of source-driven calculations for the weighting and the final calculation. Four calculations where performed, using eigenvalue and source-driven calculations in different ways. All third level results where obtained using a two energy groups calculation, to maximize the effect of the different calculation method. Since the system is infinite, the flux shape can no longer be used to compare the quality of different calculations, and since source-driven calculations always lead to an eigenvalue of 1, the k_{eff}-value can't be used either. Good comparison values are the neutron flux in the two different groups, and also the fission reaction rate and the capture reaction rate. This yielded the results from using source-driven calculations to perform both the second- and third-level calculations, and the results from using a source-driven calculation to weight the material for an eigenvalue calculation and vice versa. These results can now can be compared to the "normal" method, using eigenvalue weighting calculations to perform eigenvalue final calculations. The results are shown in table 3. At the first line it is visible that with two consecutive eigenvalue calculations the results are still accurate, while the number of energy groups is collapsed from 238 to 2. In the second case, the second source calculations give results that are a bit less, but still reasonable accurate. This effect is probably due to the fact that the source that was only in the first group energy range in the first calculation, is now in the range of the entire first broadband group. Large disagreements between the third- and second-level results arise when a second level eigenvalue calculation is used to weight the cross sections for the third level source calculation, and vice versa. This makes sense, since the weighting scheme is inconsistent, using a completely different calculation than the final calculation as the weighting calculation. It is notable that in the third case the third level final results are almost exactly ten times the results of the eigenvalue calculation that was used to weight the system. To verify that this is not just an numerical coincidence, other calculations were performed. One with decreased water density, but higher enrichment of the uranium-oxide (to keep the same criticallity), and one with the source term in the third level calculation distributed over more energy groups. In these calculations, the same factor of ten still

Calculation	Flux group 1	Flux group 2	Fission r.r.	Capture r.r.	Eigenvalue
Level 2 k_{eff}	0,3255	0,1383	0,3653	0,6334	0,9000
Level 3 k_{eff}	0,3255	0,1383	0,3652	0,6333	0,8998
Level 2 Source	7,937	3,278	8,882	15,02	1,000
Level 3 Source	7,883	3,256	8,882	14,91	1,000
Level 2 k_{eff}	0,3255	0,1383	0,3653	0,6334	0,9000
Level 3 Source	3,251	1,381	3,647	6,325	1,000
Level 2 Source	7,937	3,278	8,882	15,02	1,000
Level 3 k_{eff}	0,3389	0,1400	0,3793	0,6412	0,9570
Decreased water density and higher enrichment of the U2O					
Level 2 k_{eff}	1,404	0,01031	0,3602	0,6394	0,8999
Level 3 Source	13,91	0,1021	3,570	6,337	1,000
The source in the first broad group					
Level 2 k_{eff}	0,3255	0,1383	0,6334	0,6334	0,9000
Level 3 Source	3,502	1,677	3,570	7,366	1,000

Table 3: Flux and reaction rates in a homogeneous infinite sub-critical system, calculated through eigenvalue and source-driven calculations

exist. It was concluded that this originates in the normalization. The factor should than become ten, because the calculation has an eigenvalue of exactly 0.9.

4.4 Finite system calculations

Some interesting calculational results were obtained from the infinite system, though this isn't applicable as a model of a real power plant. Therefore, combinations of eigenvalue and source-driven calculations were performed in the same finite geometry as the eigenvalue calculations before. To keep the eigenvalue of the system at 0.900, the enrichment of the uranium-dioxide was slightly increased, to 0.796%. The source-driven calculations were performed using the equal source-terms in every interval of the system. Such a source term, or even a source term at the correct location but still in a thermal system, is unfeasible from enginering viewpoint. Yet it can provide interesting insight, because it decreases the steepness of the flux shape, allowing calculations where the leakage problems do not occur. In this case it is possible to examine the influence that the number of used spatial zones has on the combinations of source- and eigenvalue calculations. In the next two subsections, this influence is determined. First for source-driven calculations weighted using an eigenvalue calculation, and second for source-driven calculations that where correctly weighted using a source calculation.

Source-driven calculations weighted with eigenvalue calculations

In the first calculation, the material is weighted using an eigenvalue calculation. The collapsed cross sections in two energy groups are used to perform a third level source-driven calculation. The results of these calculations are shown in table 4, where some important values from the first interval are listed. There is a very specific factor between the results of the two levels. This was seen earlier in the infinite geometry, although it is not the same factor of ten as in that case. Also, it is visible that the number of weighting zones is still not very important for the accuracy of the results, since they only vary to numerical precision.

Which Calculation	Nr. of Zones	Flux group 1	Flux group 2	Fission r.r.	Capture r.r.
Level 2 k_{eff}	1	2,244E-3	8,934E-4	8,934E-4	4,212E-5
Level 3 Source	1	1,746E-2	6,969E-3	2,054E-4	3,288E-4
Level 2 k_{eff}	1	2,244E-3	8,934E-4	2,629E-5	4,212E-5
Level 3 Source	2	1,749E-2	6,980E-3	2,054E-4	3,288E-4
Level 2 k_{eff}	1	2,244E-3	8,934E-4	2,629E-5	4,212E-5
Level 3 Source	4	1,746E-2	6,969E-3	2,050E-4	3,282E-4

Table 4: First interval flux and reaction rates in a finite sub-critical system

Source-driven calculations weighted with source-driven calculations

In this finite geometry, the effects of the number of zones in the source-driven weighting calculation on the source-driven final calculation will be examined. In table 5, values of the first interval are displayed, as usual for a calculation that collapses the cross sections in two broad groups.

Calculation	Nr. of Zones	Flux group 1	Flux group 2	Fission r.r.	Capture r.r.
Level 2 Source	1	4,021E-2	1,582E-2	4,723E-4	7,448E-4
Level 3 Source	1	7,558E-2	2,923E-2	8,799E-4	1,378E-3
Level 2 Source	1	4,021E-2	1,582E-2	4,723E-4	7,448E-4
Level 3 Source	2	5,241E-2	2,059E-2	6,154E-4	9,693E-4
Level 2 Source	1	4,021E-2	1,582E-2	4,723E-4	7,448E-4
Level 3 Source	4	4,710E-2	1,853E-2	5,535E-4	8,724E-4
Level 2 Source	1	4,021E-2	1,582E-2	4,723E-4	7,448E-4
Level 3 Source	10	4,512E-2	1,770E-2	5,301E-4	8,358E-4
Level 2 Source	1	4,021E-2	1,582E-2	4,723E-4	7,448E-4
Level 3 Source	20	4,476E-2	1,760E-2	5,260E-4	8,293E-4

Table 5: Flux and reaction rates in a finite sub-critical system

The first results are the values for the second level weighting calculation, it can be seen that the second results, from the final calculation approach the second level reference values. This means that for this weighting scheme, the influence of the number of spatial zones on the accuracy of the calculations is as expected. However, the results do not converge to the exact second level values, but to a value that has a certain bias due to the fact that the cross sections are collapsed into only two energy groups. To verify that this is not just an effect in the first interval, the resulting shapes of the thermal flux are compared, see figure 5.

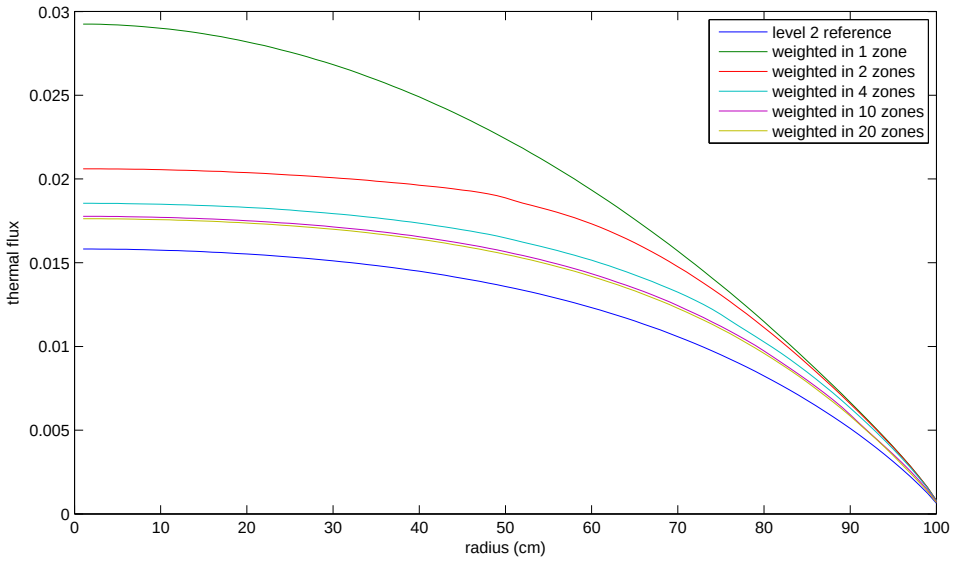

Figure 5: The neutron flux resulting from a third level source calculation in a thermal system. The cross sections are weigthed in a source-driven calculation, in different amounts of spatial zones. The cross sections are collapsed into two energy groups

In order to verify the assumption that the bias induced into this calculations originates in the weigthing procedure, the same calculations were performed, while collapsing the energy structure in 20 energy groups. As can be seen in figure 6, this seems to be a correct assumption, as the bias is smaller, and the overall convergence to the second level reference value is better.

We can now conclude that, using source calculations to weight the material for final source-driven calculations, XSDRN behaves as expected. The results will be better when collapsed into more energy groups, and weighted in more spatial zones. When a eigenvalue calculation is used to weight the material for a source-driven final calculation, the influence of the number of spatial zones is still negligible. It was concluded that this is due to the same leakage problems as encountered before.

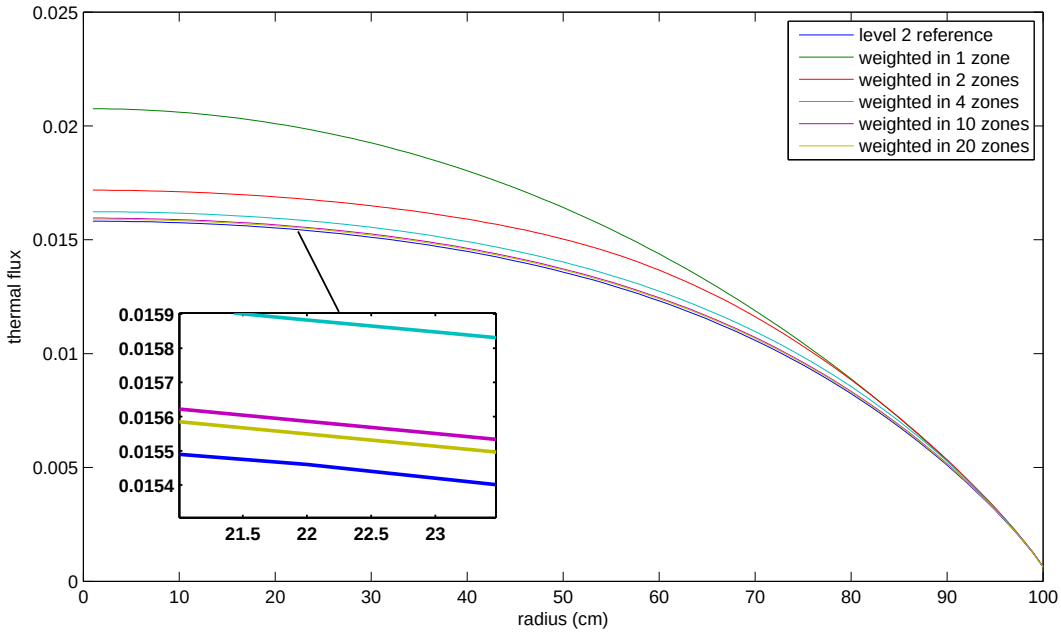

Figure 6: The neutron flux resulting from a third level source calculation in a thermal system. The cross sections are weigthed in a source-driven calculation, in different amounts of spatial zones. The cross sections are collapsed into 20 energy groups

5 Fast metal cooled reactor

The experimental nuclear reactor VENUS-F, located in Mol, Belgium, is an example of accelerator driven sub-critical systems. The reactor core consists of 89 rectangular fuel assemblies, grouped together forming approximately a cylinder with diameter 0,96 m and height 1,11 m. This cylinder is incapsulated by a lead reflector. The reactor can be operated as both a critical reactor as well as a accelerator driven system. In order to operate the reactor in accelerator driven mode, the inner four fuel assembly's are replaced with a deuterium beam tube (DTB) [5]. The next section will be on how this set-up was modeled in this research.

5.1 Model of the VENUS-F reactor

In order to approach this set-up with a geometry that is suitable for one-dimensional calculations, this reactor core was modeled as a sphere. The geometry of this sphere was designed to keep the properties of the sytem the same as in the real VENUS-F reactor. The sphere consist of three parts. The first part is located in the most inner part of the sphere. It represents the DBT. It has a radius of 12 cm, and is filled with lead. When performing source-driven calculations in this geometry, the source term will be solely in this part. The second part is a spherical shell with thickness 30 cm. In this part, the fissionable material is placed. It represents the fuel assemblies in the original VENUS-F geometry. The last part of the model is a second shell, this time with a tickness of 33 cm. This part is filled with lead, to represent the lead reflector that is surrounding the reactor core in the original geometry. The total sphere has a diameter of 75 cm, and is divided into 100 sperical shells, each acting as a computational interval.

5.2 Different kinds of weighting calculations

In the design process of the emerging acellerator driven systems, conclusions are based on a certain weighting scheme. This weighting scheme exists of using a final source calculations after weighting the cross sections using an eigenvalue calculation. Results from these kind of calculations should have major errors, as an inconsistent weighting calculation is used. But apparently this weighting method gives the right results, as it is applied in the industry. This section will be on investigating why this method is applicable. This is done by comparing the results of two third level source-driven calculations.

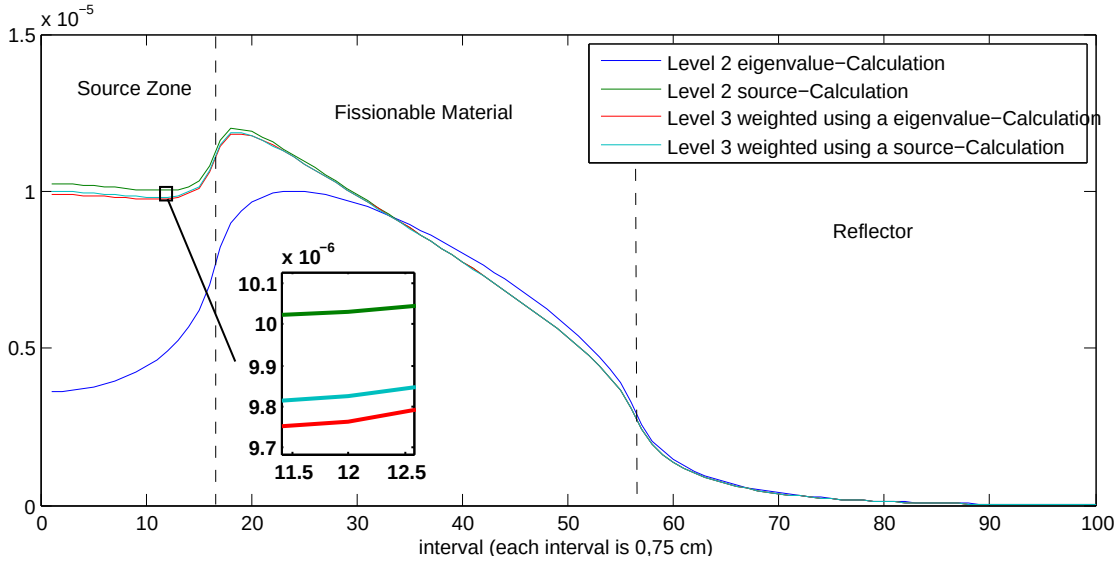

Figure 7: Shape of the third broad group flux for various calculations in the venus-F geometry

The first one is performed using cross sections that where acquired through second level eigenvalue calculations, and the second is based on correctly weighted cross sections from second level source-driven calculations. The weighting calculation is used to collapse the energy structure into 60 broad groups. All results are normalized to one fission per second, for the entire reactor in all energy groups. For the source-driven calculations, the source term is placed in the first 16 intervals, in the first broad group. It's a fast reactor, so the most interesting energy groups are the first ones. But the source term is in the first broad group, so using this group does not clarify the aims of this study. Therefore, the first two energy groups will not be used. Instead, the third broad group will be used to evaluate the flux shape throughout the core. In figure 7 the spatial flux shapes for the second and third level calculations are shown for the entire radius of the reactor. It shows that the second level eigenvalue calculation yields significantly different results than the source calculation, especially in the source zone. But the third level source calculations are surprisingly similar. They are different then the second level source calculation because of the smaller number of energy groups, but the difference between them is quite small. This, although one of them is weighted using a different kind of calculation.

5.3 Analysis on the energy spectrum in the fuelzone

Although the second level weighting calculations are quite different, the third level final calculations agree quite well with each other. To investigate what causes this, energy spectra were extracted for the second level source calculations in the fuel zone. Figure 8 shows such an energy spectrum.

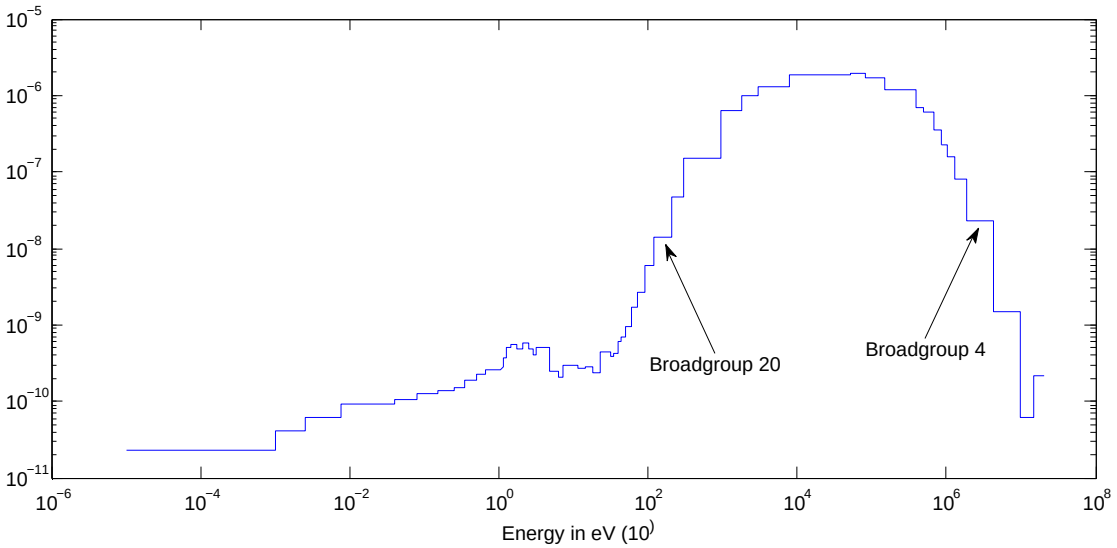

Figure 8: Energy spectrum in the fuel zone for the source calculation. The flux intensity in broad group 4-20 is much higher than in the other broad groups.

Notice that this is plotted on a logarithmic scale, and since the flux intensity in broad groups 4 to 20 is so much higher than in the rest of the groups, errors in the weighted cross sections for those other groups are not very important for the quality of the third level calculation. To evaluate if the second level calculations better agree in the more important broad groups, figure 9 represents the flux shape of the 10th broad group. It shows a much bigger resemblance between the two second level calculations. Also, it shows that almost all of the difference is in the source zone. The flux shapes agree very well in the fuel zone. The fact that the errors are mainly in the source zone is very important, because the quality of the cross sections in the fuel zone does not depend on weighting errors in the source zone. The accuracy of the third level calculations is strongly dependent on the quality of the cross sections for the fuel zone, and not so dependent on the errors in the cross sections of the source zone. This is so because fission only

occurs in the fuel zone, and the neutron flux for the entire systems depends on the way fissions occur in

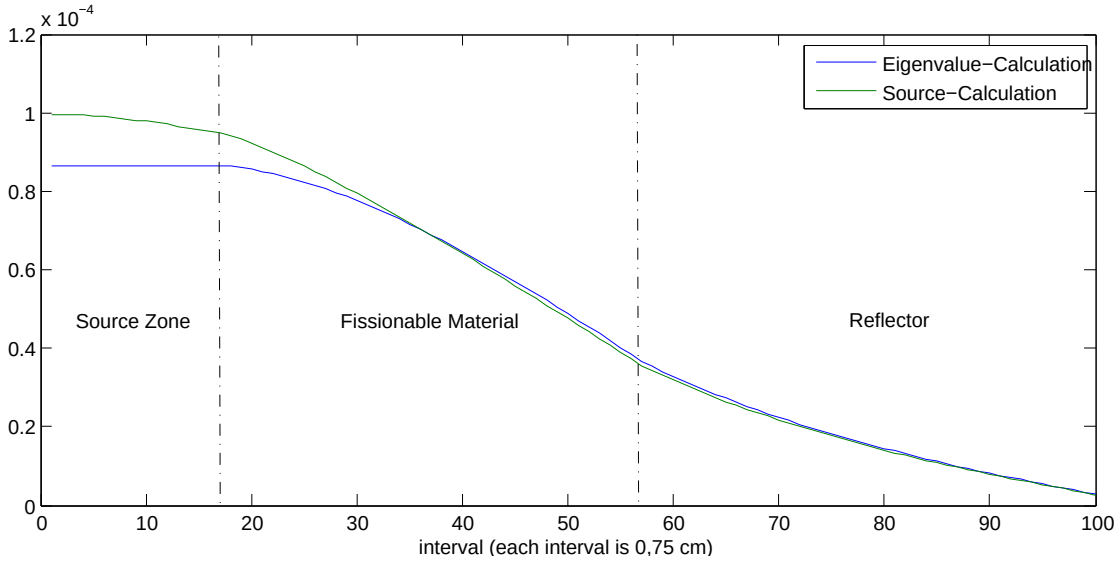

Figure 9: Spatial shape of the 10th broad group flux for different calculations in the venus-F geometry

the system. This means that for a correct weighting calculation, the only thing that is important is that the energy spectrum in the fuel zone is correct, especially in the energy range 10^2 eV - 10^6 eV. To examine the correctness of an eigenvalue weighting calculation in this zone and this energy range, figure 10 shows the difference in percentage between energy spectra in the middle of the fuel zone, with both the results of a eigenvalue calculation and a source-driven calculation. The spectra do differ from each other, about 10% even in the first broad group. But the difference in the range 10^2 eV - 10^6 is extremely small. And since this is the range in which the vast majority of all neutrons are, third level calculations performed

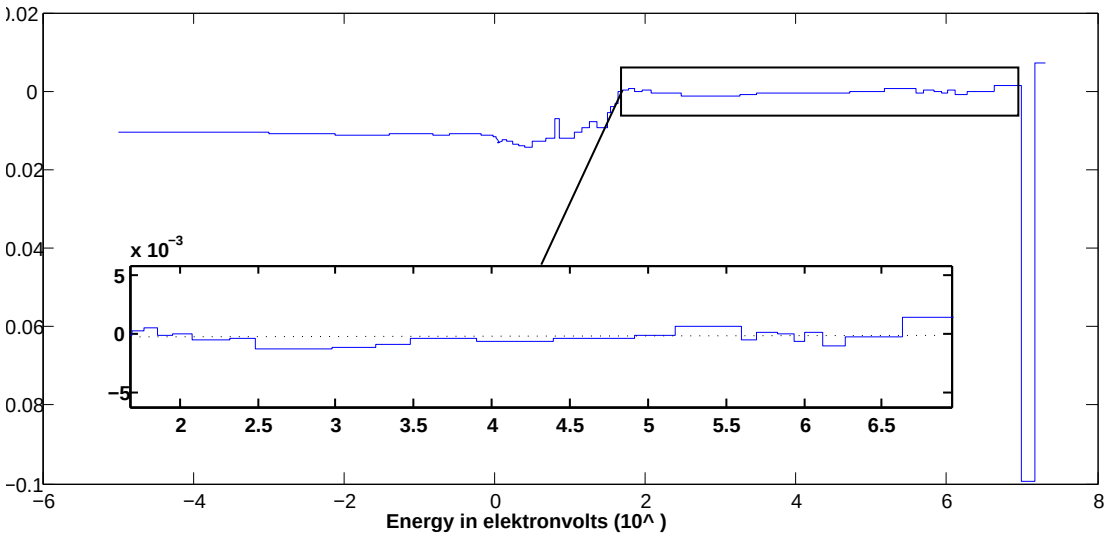

Figure 10: Difference between the energy spectra of a source-calculation and a eigenvalue calculation, in the fuel zone of the Venus-F geometry. The difference is extremely small in the energy range 10^2 eV - 10^6 eV.

using cross sections from these calculations will almost not differ one from another. So this is in fact the reason why it is possible to do a third level source calculation with eigenvalue-weighted cross sections. Eigenvalue calculations can be used to weight the cross sections for source-driven calculations, because although the calculation is quite different than the appropriate source-driven weighting calculation, the difference in the fuel zone is extremely small for the energy groups that are important.

5.4 Eight Energy group structure

It is clear how it is possible to use this weighting scheme for collapsing the energy-structure in 60 broad groups. But this is still quite a large number of groups. To see if this explanation still holds for collapsing in fewer groups, the same calculations have been performed, but now for collapsing into only 8 groups. The energy spectrum for the source-driven calculation is shown in figure 11. In this case, the flux shape is much more vigorous, only two groups are present in reasonable amount. Other groups have fluxes that are over a hundred times smaller. For these two groups, the difference between the eigenvalue calculation and the source calculations turns out to be 0.12 and 0.10 percent. So also for collapsing in few groups, the cross sections that matter are weighted with only marginal errors. So also for this type of calculations, the weighting scheme is applicable. The fact that an eigenvalue calculation based weighting is used to create the cross sections doesn't have a big influence on the quality of the final source-driven calculation.

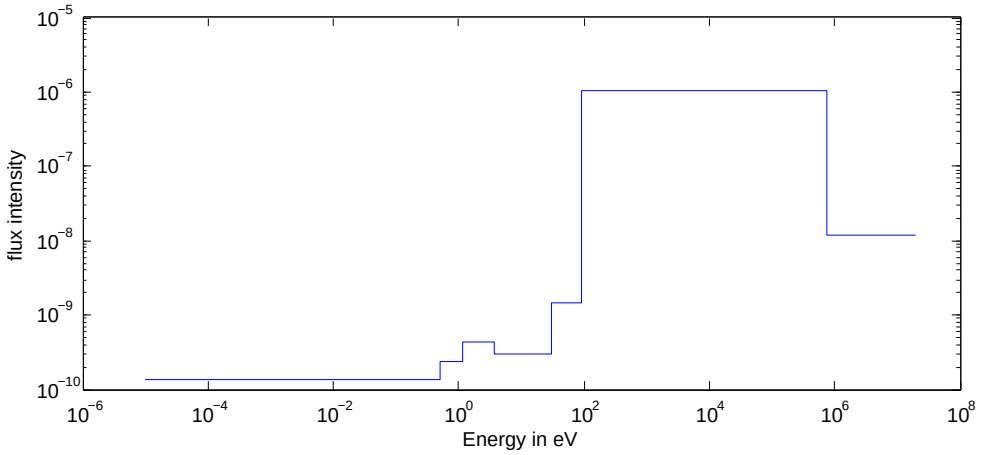

Figure 11: The energy spectrum of a 8 group source-calculation, in the fuel zone of the Venus-F geometry

6 Conclusions

In the set-up of this project, the investigation on the thermal system was just meant as a way to get reference values from a well known system. But actually, some interesting results were also gathered in this part of the research. Because of this, the conclusions drawn from results of this part of the reseach will be descbribed in a seperate section. The results of the fast metal cooled reactor also lead to some interesting conclusions, solving the question that was stated as main goal of this thesis.

6.1 Thermal System

The first conclusions on the termal system are for the cross sections that are weighted using a eigenvalue calculation. If these cross sections are used to perform a eigenvalue final calculation, the influence of the number of energy groups used is as predicted by the theory. The compensating influence of an increase in number of spatial zones was not observed. It was concluded that this is due to local leakage problems. That means that for such a system, the standard discrete ordinates approach can not be applied. Instead, another method should be used, for example the method developed by the Georgia Institute of Technology. This method explicitly accounts for the angular dependence of the total cross sections.

For the same reason, these cross sections can't be used to perform final source-driven calculations. When the crossections are weighted using a source-driven calculation, these problems do not occur. The final source-driven calculations show the predicted behaviour. The results become worse under a decrease in the number of energy broad groups. Yet this off-set can partially be compensated by using more spatial zones.

6.2 Fast metal cooled reactor

In the geometry of the VENUS-F reactor, eigenvalue calculations can be used to weight the cross sections for final source-driven calculations. This is due to the fact that the resulting energy spectrum for these types of calculations is quite similar in the fuelzone. This causes that the cross sections are weighted correctly, although the weighting scheme is incosistent. This could be a hint that this method is applicable for all accelerator-driven subcritical systems. Because of this, a recommendation for further research is to investigate if this method works for different ADS, or that the similarity of the energy spectrum is a coincidence, and only exists in this specific geometry.

References

[1] James J. Duderstadt and Louis J. Hamilton. *Nuclear reactor analysis.* John Wiley and Sons, Inc., 1976.

[2] Oak Ridge National Laboratory. *SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation* Nuclear Science and Technology Division, January 2009.

[3] N.M. Greene and L.M. Petrie. *XSDRNPM: a one-dimensional discrete-ordinates code for transport analysis.* Nuclear Science and Technology Division, January 2009.

[4] S. Goluoglu D. F. Hollenbach N. F. Landers L. M. Petrie J. A. Bucholz C. F. Weber C. M. Hopper *THE MATERIAL INFORMATION PROCESSOR FOR SCALE* Nuclear Science and Technology Division, January 2009.

[5] Wim Uyttenhove, Peter Baeten, Anatoly Kochetkov. *Description of the VENUS-F Reactor for Neutronic Calculations* Version 1, March 8, 2010

[6] Steven Douglass, Farzad Rahnema. *Consistent generalized energy condensation theory* Annals of Nuclear Energy 40, 2012

[7] Nikolay Fil. *Status and perspectives of VVER nuclear power plants* OKB Gidropress. IAEA. September 2011

A XSDRN input shell for simulations in the thermal sytem

```
#/bin/sh

for egrps in 2 8 20 60 120; do
for zones in 1 2 4 10 20; do
for lvltwee in 0 1; do
for lvldrie in 0 1; do

if [ $egrps = 2 ];
then
lvl2p51="202r1 36r2e";
elif [ $egrps = 8 ];
then
lvl2p51="28r1 29r2 29r3 29r4 29r5 29r6 29r7 36r8";
elif [ $egrps = 20 ];
then
lvl2p51="11r1 11r2 12r3 12r4 12r5 12r6 12r7 12r8 12r9 12r10
12r11 12r12 12r13 12r14 12r15 12r16 12r17 12r18 12r19 12r20";
elif [ $egrps = 60 ];
then
lvl2p51="3r1 3r2 4r3 4r4 4r5 4r6 4r7 4r8 4r9 4r10 4r11 4r12
4r13 4r14 4r15 4r16 4r17 4r18 4r19 4r20 4r21 4r22 4r23 4r24
4r25 4r26 4r27 4r28 4r29 4r30 4r31 4r32 4r33 4r34 4r35 4r36
4r37 4r38 4r39 4r40 4r41 4r42 4r43 4r44 4r45 4r46 4r47 4r48
4r49 4r50 4r51 4r52 4r53 4r54 4r55 4r56 4r57 4r58 4r59 4r60";
else
lvl2p51="1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10
11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20
21 21 22 22 23 23 24 24 25 25 26 26 27 27 28 28 29 29 30 30
31 31 32 32 33 33 34 34 35 35 36 36 37 37 38 38 39 39 40 40
41 41 42 42 43 43 44 44 45 45 46 46 47 47 48 48 49 49 50 50
51 51 52 52 53 53 54 54 55 55 56 56 57 57 58 58 59 59 60 60
61 61 62 62 63 63 64 64 65 65 66 66 67 67 68 68 69 69 70 70
71 71 72 72 73 73 74 74 75 75 76 76 77 77 78 78 79 79 80 80
81 81 82 82 83 83 84 84 85 85 86 86 87 87 88 88 89 89 90 90
91 91 92 92 93 93 94 94 95 95 96 96 97 97 98 98 99 99 100 100
101 101 102 102 103 103 104 104 105 105 106 106 107 107 108 108
109 109 110 110 111 111 112 112 113 113 114 114 115 115 116 116
117 117 118 118 119 120";
fi

if [ $zones = 1 ];
then
lvl2p13="1"; lvl2p14="100"; lvl2p15="1"; lvl2p36="100r1";
lvl2p39="f1";lvl2p49="1 1"; lvl2p50="18 102";
lvl3p13="1"; lvl3p14="1100"; lvl3p15="1"; lvl3p36="100r1";
lvl3p39="1"; lvl3p49="1 1"; lvl3p50="18 102";
elif [ $zones = 2 ];
then
lvl2p13="1"; lvl2p14="100"; lvl2p15="1"; lvl2p36="50r1 50r2";
lvl2p39="f1"; lvl2p49="1 1"; lvl2p50="18 102";
lvl3p13="1 2"; lvl3p14="1100 2100"; lvl3p15="1 1";
lvl3p36="50r1 50r2"; lvl3p39="1 2";
lvl3p49="1 1 2 2"; lvl3p50="18 102 18 102";
elif [ $zones = 4 ];
then
lvl2p13="1"; lvl2p14="100"; lvl2p15="1";
lvl2p36="25r1 25r2 25r3 25r4";
lvl2p39="f1"; lvl2p49="1 1"; lvl2p50="18 102";
lvl3p13="1 2 3 4"; lvl3p14="1100 2100 3100 4100";
lvl3p15="1 1 1 1"; lvl3p36="25r1 25r2 25r3 25r4";
lvl3p39="1 2 3 4";lvl3p49="1 1 2 2 3 3 4 4";
lvl3p50="18 102 18 102 18 102 18 102";
elif [ $zones = 10 ];
then
lvl2p13="1"; lvl2p14="100"; lvl2p15="1";
lvl2p36="10r1 10r2 10r3 10r4 10r5 10r6 10r7 10r8 10r9 10r10";
lvl2p39="f1"; lvl2p49="1 1"; lvl2p50="18 102";
lvl3p13="1 2 3 4 5 6 7 8 9 10";
lvl3p14="1100 2100 3100 4100 5100 6100 7100 8100 9100 10100";
lvl3p15="1 1 1 1 1 1 1 1 1 1";
lvl3p36="10r1 10r2 10r3 10r4 10r5 10r6 10r7 10r8 10r9 10r10";
lvl3p39="1 2 3 4 5 6 7 8 9 10";
lvl3p49="1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10";
lvl3p50="18 102 18 102 18 102 18 102 18 102 18 102 18 102
 18 102 18 102 18 102";
else
lvl2p13="1"; lvl2p14="100"; lvl2p15="1";
lvl2p36="5r1 5r2 5r3 5r4 5r5 5r6 5r7 5r8 5r9 5r10
5r11 5r12 5r13 5r14 5r15 5r16 5r17 5r18 5r19 5r20"; lvl2p39="f1";
lvl2p49="1 1"; lvl2p50="18 102";
lvl3p13="1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20";
lvl3p14="1100 2100 3100 4100 5100 6100 7100 8100 9100 10100
11100 12100 13100 14100 15100 16100 17100 18100 19100 20100";
lvl3p15="1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1";
lvl3p36="5r1 5r2 5r3 5r4 5r5 5r6 5r7 5r8 5r9 5r10
5r11 5r12 5r13 5r14 5r15 5r16 5r17 5r18 5r19 5r20";
lvl3p39="1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20";
lvl3p49="1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10
11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20";
lvl3p50="18 102 18 102 18 102 18
102 18 102 18 102 18 102 18 102 18 102 18 102 18 102 18 102
102 18 102 18 102 18 102 18 102 18 102 18 102 18 102 18 102";
fi

if [ $lvltwee = 0 ];
then p30="30\$$ 1 99r1"; p31="31** 1 237r0"; ter="T"; p3r2lvl2=" 1"
else p30="'30\$$ 1 99r1"; p31="'31** 1 237r0"; ter="'T"; p3r2lvl2=" 0"
fi

egrpsmin1=`awk 'BEGIN{print -1.0+'$egrps';}'`

if [ $lvldrie = 0 ];
then p30n="30\$$ 1 99r1"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
p3r2lvl3=" 1"
else p30n="'30\$$ 1 99r1"; p31n="'31** 1 0"; tern="'T"; p3r2lvl3=" "
fi

nract=`awk 'BEGIN{print 2.0*'$zones';}'`
rvlvltwee=`awk 'BEGIN{print 1.0-'$lvltwee';}'`
rvlvldrie=`awk 'BEGIN{print 1.0-'$lvldrie';}'`


cat >lvl23fin-${egrps}grp-${zones}zn-type${lvltwee}${lvldrie}.input
<<ENDXX
=shell
cp \$RTNDIR/outputfin.ampx ft04f001
end
=xsdrn
level 2 ${egrps}grps ${zones}zns type$lvltwee calculation
1\$$ 2 $zones 100 1 0 1 1 8 2 $lvltwee 10 1000 e
2\$$ a2 0 a4 1 a7 0 0 e
3\$$ 1$p3r2lvl2 a9 2 1 1 e
4\$$ 0 $egrps 3 -1 e
5** 1e-6 1e-7 e
T
13\$$ $lvl2p13
14\$$ $lvl2p14
15**  $lvl2p15
T
$p30
$p31
$ter
33## f1
T
35** 99i0.00 100.00 e
36\$$ $lvl2p36 e
39\$$ $lvl2p39
49\$$ $lvl2p49 e
50\$$ $lvl2p50 e
51\$$ $lvl2p51 e
T
end
=shell
cp ft03f001
\$RTNDIR/fin-${egrps}grp${zones}zones-type${lvltwee}${lvldrie}.ampx
end
=shell
cp \$RTNDIR/fin-${egrps}grp${zones}zones-type${lvltwee}${lvldrie}.ampx
ft04f001
end
=xsdrn
level 3 ${egrps}grps ${zones}zns type$lvldrie calculation
1\$$ 2 $zones 100 1 0 $zones $zones 8 2 $lvldrie 10 1000 e
3\$$ 0$p3r2lvl3 a9 $nract 1 1 e
5** 1e-6 1e-7 e
T
13\$$ $lvl3p13
14\$$ $lvl3p14
15**  $lvl3p15
T
$p30n
$p31n
$tern
33## f1
T
35** 99i0.00 100.00 e
36\$$ $lvl3p36 e
39\$$ $lvl3p39
49\$$ $lvl3p49 e
50\$$ $lvl3p50 e
T
end
ENDXX

echo buissy calculating
lvl23fin-${egrps}grp-${zones}zn-type${lvltwee}${lvldrie}
batch6
lvl23fin-${egrps}grp-${zones}zn-type${lvltwee}${lvldrie}.input
echo done calculating
lvl23fin-${egrps}grp-${zones}zn-type${lvltwee}${lvldrie}
cat lvl23fin-${egrps}grp-${zones}zn-type${lvltwee}${lvldrie}.msgs
nedit
lvl23fin-${egrps}grp-${zones}zn-type${lvltwee}${lvldrie}.output &

done
done
done
done
```

B XSDRN input shell for simulations in the VENUS-F geometry

```
#/bin/sh

for egrps in 120 60 20 12 10 8 2; do
for zones in 1 2 4 10 20; do
for lvltwee in 0 1; do
for lvldrie in 0 1; do

egrpsmin1=`awk 'BEGIN{print -1.0+'$egrps';}'`
zonesplus2=`awk 'BEGIN{print 2.0+'$zones';}'`
nract=`awk 'BEGIN{print 2.0*'$zones';}'`
rvlvltwee=`awk 'BEGIN{print 1.0-'$lvltwee';}'`
rvlvldrie=`awk 'BEGIN{print 1.0-'$lvldrie';}'`
if [ $egrps = 2 ];
then
lvl2p51="202r1 36r2e";
elif [ $egrps = 8 ];
then
lvl2p51="28r1 29r2 29r3 29r4 29r5 29r6 29r7 36r8";
elif [ $egrps = 10 ];
then
lvl2p51="26r1 26r2 26r3 26r4 26r5 24r6 24r7 24r8 18r9 18r10";
elif [ $egrps = 12 ];
then
lvl2p51="21r1 21r2 20r3 20r4 20r5 20r6 20r7 20r8 20r9 20r10
18r11 18r12";
elif [ $egrps = 20 ];
then
lvl2p51="11r1 11r2 12r3 12r4 12r5 12r6 12r7 12r8 12r9 12r10
12r11 12r12 12r13 12r14 12r15 12r16 12r17 12r18 12r19 12r20";
elif [ $egrps = 60 ];
then
lvl2p51="3r1 3r2 4r3 4r4 4r5 4r6 4r7 4r8 4r9 4r10 4r11 4r12
4r13 4r14 4r15 4r16 4r17 4r18 4r19 4r20 4r21 4r22 4r23 4r24
4r25 4r26 4r27 4r28 4r29 4r30 4r31 4r32 4r33 4r34 4r35 4r36
4r37 4r38 4r39 4r40 4r41 4r42 4r43 4r44 4r45 4r46 4r47 4r48
4r49 4r50 4r51 4r52 4r53 4r54 4r55 4r56 4r57 4r58 4r59 4r60";
else
lvl2p51="1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10
11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20
21 21 22 22 23 23 24 24 25 25 26 26 27 27 28 28 29 29 30 30
31 31 32 32 33 33 34 34 35 35 36 36 37 37 38 38 39 39 40 40
41 41 42 42 43 43 44 44 45 45 46 46 47 47 48 48 49 49 50 50
51 51 52 52 53 53 54 54 55 55 56 56 57 57 58 58 59 59 60 60
61 61 62 62 63 63 64 64 65 65 66 66 67 67 68 68 69 69 70 70
71 71 72 72 73 73 74 74 75 75 76 76 77 77 78 78 79 79 80 80
81 81 82 82 83 83 84 84 85 85 86 86 87 87 88 88 89 89 90 90
91 91 92 92 93 93 94 94 95 95 96 96 97 97 98 98 99 99 100 100
101 101 102 102 103 103 104 104 105 105 106 106 107 107 108 108
109 109 110 110 111 111 112 112 113 113 114 114 115 115 116 116
117 117 118 118 119 120";
fi
if [ $zones = 1 ];
then
lvl2p13="1 2"; lvl2p14="1001 1004"; lvl2p15="f1";
lvl2p36="16r1 40r2 44r3"; lvl2p39="2 1 2";lvl2p49="2 2";
lvl2p50="18 102";lvl3p13="1 2 3"; lvl3p14="11004 21001 31004";
lvl3p15="f1"; lvl3p36="16r1 40r2 44r3"; lvl3p39="1 2 3";
lvl3p49="2 2"; lvl3p50="18 102";
elif [ $zones = 2 ];
then
llvl2p13="1 2"; lvl2p14="1001 1004"; lvl2p15="f1";
lvl2p36="8r1 8r2 20r3 20r4 22r5 22r6"; lvl2p39="2 2 1 1 2 2";
lvl2p49="3 3 4 4"; lvl2p50="18 102 18 102"; lvl3p13="1 2 3 4 5 6";
lvl3p14="11004 21004 31001 41001 51004 61004"; lvl3p15="f1";
lvl3p36="8r1 8r2 20r3 20r4 22r5 22r6"; lvl3p39="1 2 3 4 5 6";
lvl3p49="3 3 4 4"; lvl3p50="18 102 18 102";
elif [ $zones = 4 ];
then
llvl2p13="1 2"; lvl2p14="1001 1004"; lvl2p15="f1";
lvl2p36="4r1 4r2 4r3 4r4 10r5 10r6 10r7 10r8 11r9 11r10 11r11
11r12"; lvl2p39="2 2 2 2 1 1 1 1 2 2 2 2";
lvl2p49="5 5 6 6 7 7 8 8";
lvl2p50="18 102 18 102 18 102 18 102";
lvl3p13="1 2 3 4 5 6 7 8 9 10 11 12";
lvl3p14="11004 21004 31004 41004 51001 61001 71001 81001 91004
101004 111004 121004"; lvl3p15="f1";
lvl3p36="4r1 4r2 4r3 4r4 10r5 10r6 10r7 10r8 11r9 11r10 11r11 11r12
       ";
lvl3p39="1 2 3 4 5 6 7 8 9 10 11 12"; lvl3p49="5 5 6 6 7 7 8 8";
lvl3p50="18 102 18 102 18 102 18 102";
elif [ $zones = 10 ];
then
lvl2p13="1 2"; lvl2p14="1001 1004"; lvl2p15="f1";
lvl2p36="2r1 2r2 2r3 2r4 2r5 2r6 2r7 2r8 4r9
4r10 4r11 4r12 4r13 4r14 4r15
        4r16 4r17 4r18 4r19 4r20 4r21
        4r22 4r23 4r24 4r25 4r26 4r27 4r28 4r29";
lvl2p39="2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
       2";
lvl2p49="9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18
       18";
lvl2p50="18 102 18 102 18 102 18 102 18 102
18 102 18 102 18 102 18 102 18 102";
lvl3p13="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";
lvl3p14="11004 21004 31004 41004 51004 61004 71004 81004 91001
       101001
        111001 121001 131001 141001 151001 161001 171001
        181001 191004 201004
        211004 221004 231004 241004 251004 261004 271004 281004
               291004";
lvl3p15="f1"; lvl3p36="2r1 2r2 2r3 2r4 2r5 2r6 2r7 2r8
4r9 4r10 4r11 4r12 4r13 4r14 4r15
        4r16 4r17 4r18 4r19 4r20 4r21 4r22 4r23
        4r24 4r25 4r26 4r27 4r28 4r29";
lvl3p39="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";
lvl3p49="9 9 10 10 11 11 12 12 13 13 14 14
15 15 16 16 17 17 18 18";
lvl3p50="18 102 18 102 18 102 18 102 18
102 18 102 18 102 18 102 18 102 18 102";
else
lvl2p13="1 2";
lvl2p14="1001 1004";
lvl2p15="f1";
lvl2p36="14i1 16 2r17 2r18 2r19 2r20 2r21 2r22 2r23 2r24 2r25 2r26
        2r27 2r28 2r29 2r30 2r31 2r32 2r33 2r34 2r35 2r36 2r37 2r38
        2r39 2r40 2r41 2r42 2r43 2r44 2r45 2r46 2r47 2r48 2r49 2r50
        2r51 2r52 2r53 2r54 2r55 2r56 2r57 2r58";
lvl2p39="2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
        1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
               2";
lvl2p49="17 17 18 18 19 19 20 20 21 21 22 22 23 23 24 24 25 25 26
       26
        27 27 28 28 29 29 30 30 31 31 32 32 33 33 34 34 35 35 36
               36";
lvl2p50="18 102 18 102 18 102
        18 102 18 102 18 102 18 102 18 102 18 102 18 102 18 102 18
        102 18 102 18 102 18 102 18 102 18 102 18 102 18 102 18
               102";
lvl3p13="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 40 41 42 43
        44 45 46 47 48 49 50 51 52 53 54 55 56 57 58"
lvl3p14="11004 21004 31004 41004 51004 61004 71004 81004 91004
       101004
        111004 121004 131004 141004 151004 161004 171001 181001
               191001
        201001 211001 221001 231001 241001 251001 261001 271001
        281001 291001 301001 311001 321001 331001 341001 351001
               361001
        371004 381004 391004 401004 411004 421004 431004 441004
               451004
        461004 471004 481004 491004 501004 511004 521004 531004
               541004
        551004 561004 571004 581004";
lvl3p15="f1";
lvl3p36="14i1 16 2r17 2r18 2r19 2r20 2r21 2r22 2r23 2r24 2r25 2r26
        2r27 2r28 2r29 2r30 2r31 2r32 2r33 2r34 2r35 2r36 2r37 2r38
                2r39 2r40
        2r41 2r42 2r43 2r44 2r45 2r46 2r47 2r48 2r49 2r50 2r51 2r52
                2r53 2r54
        2r55 2r56 2r57 2r58";
lvl3p39="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 40 41 42 43 44 45
                46 47
        48 49 50 51 52 53 54 55 56 57 58";
lvl3p49="17 17 18 18 19 19 20 20 21 21 22 22 23 23 24 24 25 25 26
       26
        27 27 28 28 29 29 30 30 31 31 32 32 33 33 34 34 35 35 36
               36";
lvl3p50="18 102 18 102 18 102 18
        102 18 102 18 102 18 102 18 102 18 102 18 102 18 102 18 102 18
        102 18 102 18 102 18 102 18 102 18 102 18 102 18 102 18
               102";
fi
if [ $lvltwee = 0 ];
then
if [ $lvldrie = 0 ];
then
if [ $egrps = 2 ];
then p30="30\$$ 16r1 84r0"; p31="31** 202r1 36r0"; ter="T";
       p3r2lvl2=" 1";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 8 ];
then p30="30\$$ 16r1 84r0"; p31="31** 28r1 210r0"; ter="T";
       p3r2lvl2=" 1";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 10 ];
then p30="30\$$ 16r1 84r0"; p31="31** 26r1 210r0"; ter="T";
       p3r2lvl2=" 1";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 12 ];
then p30="30\$$ 16r1 84r0"; p31="31** 21r1 210r0"; ter="T";
       p3r2lvl2=" 1";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 20 ];
then p30="30\$$ 16r1 84r0"; p31="31** 11r1 227r0"; ter="T";
       p3r2lvl2=" 1";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 60 ];
```

```
then p30="30\$$ 16r1 84r0"; p31="31** 3r1 235r0"; ter="T"; p3r2lvl2
       =" 1";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
else p30="30\$$ 1 99r0"; p31="31** 2r1 236r0"; ter="T"; p3r2lvl2="
       1";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
fi
else
if [ $egrps = 2 ];
then p30="30\$$ 16r1 84r0"; p31="31** 202r1 36r0"; ter="T";
       p3r2lvl2=" 1";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 8 ];
then p30="30\$$ 16r1 84r0"; p31="31** 28r1 210r0"; ter="T";
       p3r2lvl2=" 1";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 10 ];
then p30="30\$$ 16r1 84r0"; p31="31** 26r1 210r0"; ter="T";
       p3r2lvl2=" 1";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 12 ];
then p30="30\$$ 16r1 84r0"; p31="31** 21r1 210r0"; ter="T";
       p3r2lvl2=" 1";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 20 ];
then p30="30\$$ 16r1 84r0"; p31="31** 11r1 227r0"; ter="T";
       p3r2lvl2=" 1";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 60 ];
then p30="30\$$ 16r1 84r0"; p31="31** 3r1 235r0"; ter="T"; p3r2lvl2
       =" 1";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
else p30="30\$$ 16r1 84r0"; p31="31** 2r1 236r0"; ter="T"; p3r2lvl2
       =" 1";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
fi
fi
else
if [ $lvldrie = 0 ];
then
if [ $egrps = 2 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 8 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 10 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 12 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 20 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
elif [ $egrps = 60 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
else p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="30\$$ 16r1 84r0"; p31n="31** 1 ${egrpsmin1}r0"; tern="T";
       p3r2lvl3=" 1";
fi
else
if [ $egrps = 2 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 8 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 10 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 12 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 20 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
elif [ $egrps = 60 ];
then p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
else p30="'30\$$ 1 99r0"; p31="'31** 202r1 36r0"; ter="'T";
       p3r2lvl2=" ";
p30n="'30\$$ 1 99r0"; p31n="'31** 1 ${egrpsmin1}r0"; tern="'T";
       p3r2lvl3=" ";
fi
fi
fi
if [ $zones = 1 ];
then totzns=3
elif [ $zones = 2 ];
then totzns=6
elif [ $zones = 4 ];
then totzns=12
elif [ $zones = 10 ];
then totzns=29
else totzns=58
fi
if [ $lvldrie = 0 ];
then accconlvl3="1"
else accconlvl3="0"
fi
if [ $lvltwee = 0 ];
then accconlvl2="1"
else accconlvl2="0"
fi
cat >lvl23venus-${egrps}grp-${zones}zn-type${lvltwee}${lvldrie}.
       input
<<ENDXX
=shell
cp \$RTNDIR/reactormats238grp.ampx ft04f001
end
=xsdrn
level 2 ${egrps}grps ${zones}zns type$lvltwee calculation
1\$$ 3 $totzns 100 1 0 2 2 16 2 $lvltwee 100 1000 e
2\$$ a2 0 a4 1 a7 0 0 e
3\$$ 1$p3r2lvl2 a9 $nract 1 $accconlvl2 e
4\$$ 0 $egrps 3 -1 e
5** 1e-7 1e-8 e
T
13\$$ $lvl2p13
14\$$ $lvl2p14
15**  $lvl2p15
T
$p30
$p31
$ter
33## f1
T
35** 99i0.00 75.00 e
36\$$ $lvl2p36 e
39\$$ $lvl2p39 e
49\$$ $lvl2p49 e
50\$$ $lvl2p50 e
51\$$ $lvl2p51 e
T
end
=shell
cp ft03f001
\$RTNDIR/venus-${egrps}grp${zones}zones-type${lvltwee}${lvldrie}.
       ampx
end
=shell
cp \$RTNDIR/venus-${egrps}grp${zones}zones-type${lvltwee}${lvldrie
       }.ampx
ft04f001
end
=xsdrn
level 3 ${egrps}grps ${zones}zns type$lvldrie calculation
1\$$ 3 $totzns 100 1 0 $totzns $totzns 16 2 $lvldrie 100 1000 e
3\$$ 0$p3r2lvl3 a9 $nract 1 $accconlvl3 e
5** 1e-7 1e-8 e
T
13\$$ $lvl3p13
14\$$ $lvl3p14
15**  $lvl3p15
T
$p30n
$p31n
$tern
33## f1
T
35** 99i0.00 75.00 e
36\$$ $lvl3p36 e
39\$$ $lvl3p39 e
49\$$ $lvl3p49 e
50\$$ $lvl3p50 e
T
end
ENDXX

done
done
done
done
```