VSOP Benchmark of the ASTRA Critical Facility with Special Emphasis on the Equivalent Control Rod Model

by

Diana Naidoo
Bsc (Hons), University of Natal, Durban (1997)

A dissertation presented to the Faculty of Natural Sciences
of the
NORTHWEST UNIVERSITY
for the partial fulfilment of the degree of

MAGISTER SCIENTIAE

Promoter:
Prof. Harm Moraal
School of Physics
Northwest University

Co-Promoter:
Frederik Reitsma
NECSA
Pelindaba
South Africa

POTCHEFSTROOM
2004

ABSTRACT

The ASTRA Critical Facility at the Kurchatov Institute in Moscow was used to perform critical experiments to simulate the PBMR reactor core design as being developed by PBMR (Pty) Ltd. The aim of these benchmarks is to validate the code system VSOP, currently used for PBMR core neutronics calculations. In this work these benchmark calculations are discussed and investigated.

Experiments performed at the facility include criticality experiments, control rod worth measurements and reactivity measurements. To aid in the verification, the Monte Carlo code MCNP was used to calculate some of the experiments and compared to the VSOP results.

The flux solver in VSOP uses the finite difference diffusion method to determine the neutron distribution in the core. The problem of modeling a highly absorbing region, such as control rods, in diffusion calculations is well known and many methods have been developed to accommodate the transport effects in diffusion theory. This problem is compounded in the case of the ASTRA facility (and PBMR) due to the positioning of the control rods in the side reflector and the associated directional dependence of the flux. One of the methods that can be used for the control rod model is that of equivalent cross sections. This method has been shown in the past to yield acceptable results for pebble bed type reactors. In this work the method of equivalent cross-sections is evaluated by applying it to calculations of control rod experiments performed at the ASTRA facility. It is shown that results are favorable for control rods situated within the first ring of reflector blocks, with larger error obtained for control rods situated further from the core. Additionally, a method in which an equivalent boron concentration is used to represent the absorber region is investigated. This is shown to be useful if applied correctly and with care, especially in the case of differential control rod worth.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, for his constructive input and help. He supported me through many a "battle" with the codes and methods.

I also want to acknowledge PBMR (Pty)Ltd for their support and permission to publish this work.

Thank you to my promoter for his constructive input and of course to the Northwest University.

Lastly my thank you to all those people at home and at work that supported me in my efforts and pushed me to continue and complete this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Overview

Validation of any code system used in the design of a nuclear reactor, especially when pertaining to the neutronics of the core, is of utmost importance in any project. Though often the codes have been validated for existing designs, new innovations and changes necessitate additional validation to capture these new features. This is the case for the pebble bed modular reactor (PBMR 268 MWt) [1] currently being designed by PBMR (Pty) Ltd, where the code package used for core neutronics calculations, VSOP [2] needs to be validated for the envisaged core design.

As part of the validation, a critical facility named ASTRA was set up at the Russian Research Center, Kurchatov Institute, Moscow [3], under contract from PBMR (Pty) Ltd to simulate the neutronics of the PBMR design. Various benchmark experiments were performed at the facility, and it is these experiments that are the basis of the validation under investigation in this thesis. A picture of the ASTRA facility is shown in Figure 1-1.

Figure 1-1: The ASTRA critical facility at the Russian Research Center, Kurchatov Institute Moscow

The PBMR is a high temperature gas cooled reactor type, which uses spherical fuel elements containing coated particle kernels of uranium. The core is cylindrical in shape and surrounded by a graphite reflector. Figure 1-2 shows a graphical representation of the PBMR 268 MWt core.

Two main features of the PBMR design distinguish it from previous high temperature pebble bed type designs, namely the presence of a dynamic central column of graphite

spheres in the center of the core, and, secondly the positioning of the control rods in the side reflector while still maintaining a relatively large core diameter. Previously implemented reactor designs had either noses that protruded into the core (AVR [4]) or in-core control rods (THTR [5]). The central column is included to flatten the flux profile and thereby ensure a more even distribution of the flux in the radial direction. This also provides increased leakage into the side reflector of the core, resulting in the possibility to position the control rods in the side reflector and still obtain sufficient control rod worth for reactor control.

The new features included in the PBMR design increase the importance of performing benchmark calculations to validate the usage of VSOP. The aim of this work is to provide such a benchmark in the form of comparisons between the experimental results and results calculated with the code system. Experiments investigated include critical experiments, control rod worth measurements (including integral and total worth measurements) and reactivity effects.

The flux solver in VSOP, namely CITATION [8], is based on finite difference diffusion methods. As discussed in Section 3.5, diffusion theory breaks down in the presence of a strong absorber, such as control rods. In the PBMR design this problem is compounded by the fact that there is a directional dependence in the flux at the control rod position due to the streaming of neutrons out of the core: another property that diffusion theory can not account for. The validation of current methods for control rod worth calculations is therefore one of the most pressing needs in the validation of VSOP and the main focus of this work.

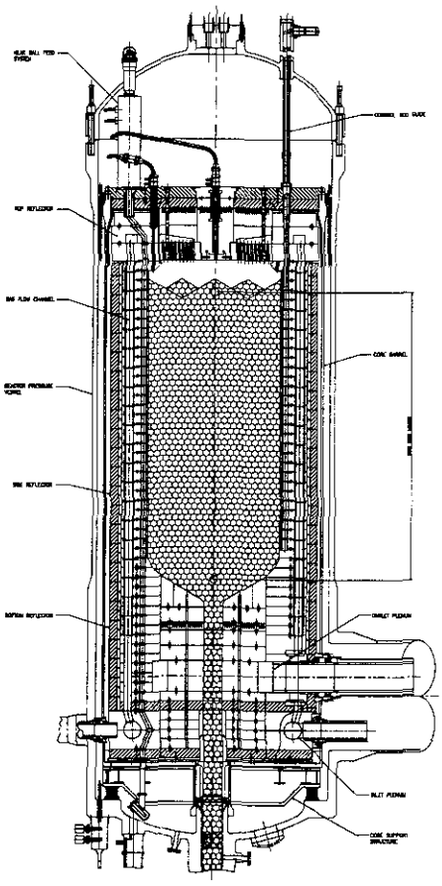

Figure 1-2: PBMR 268 MWt core layouts

In order to extend the investigation, verification of some of the results was also performed using Monte Carlo methods via the computer code MCNP [6]. Being a full 3-dimensional code, it allows for more detailed modeling of the geometry of the reactor, and, since the stochastic method used does not have the same shortcomings as the diffusion method, it provides an efficient tool for verification of the calculations.

1.2 Introduction to Core Neutronics Calculations

The design of nuclear reactors requires extensive use of computational code packages for the analysis of the core in terms of such factors as safety and economic consideration, among others. In determining these values, the most important factor is the distribution of neutrons in the core. It is the neutron distribution that determines the rate at which various

nuclear reactions occur within the reactor. Furthermore by studying the behavior of the neutron population it is possible to infer the stability of the system (i.e. the criticality). To determine the neutron distribution in the core it is required to investigate the process of neutron transport.

The roots of transport theory go back to the Boltzmann equation, first formulated in the study of the kinetic theory of gases. With the advent of the nuclear chain reaction, interest arose in its application to neutron transport.

The neutron transport equation is a rather complicated integro-differential equation, an analytical solution for which can be found only for the simplest geometries. For modern reactor designs such as the PBMR, this is certainly not possible and numerical models are required. Though direct solution methods of the neutron transport equation exist, they are restricted to rather simple geometries (often 1-D or 2-D representations). Other methods such as for example Monte Carlo [6] require long computational times that make them unsuitable for day to day design calculations. The most commonly used approximation to the neutron transport equation that is accurate and fast enough to be viable for design calculations is the diffusion approximation, in which a direct relationship between the neutron current and the neutron flux is postulated. Such a condition, however, necessitates several assumptions: a linearly isotropic angular flux, isotropic neutron sources, and a slowly time varying neutron current density as compared to the mean collision time. Considering this it is clear that diffusion theory is violated near boundaries or where material properties change dramatically over mean free path type distances, near localized sources and in strongly absorbing media.

1.3 General Definitions

This section provides definitions for some of the commonly used terms in this dissertation.

1.3.1 Criticality

In order to obtain a stable chain reaction it is required that the number of neutrons in the system remains constant from one generation to another. The ratio between the number of neutrons in one generation to the number of neutrons in the next generation is referred to as the multiplication factor k, or k-effective. If k is one, the system is referred to as critical. For a k greater than one, the system is called supercritical, and subcritical when k is less than 1.

In terms of the neutron transport equation, the k refers to the factor adjusting the fission source to obtain a solution to the steady state form of the equation.

1.3.2 Reactivity

The reactivity, ρ, of a system is a measure of the system's deviation from a multiplication factor of one.

$$\rho = \frac{k-1}{k} \tag{1-1}$$

Reactivity is often quoted in the unit of "pcm". 1 pcm is equivalent to a reactivity of 10^{-5}.

1.3.3 Control Rod Worth

The function of the control rods in a nuclear reactor core is to control the chain reaction trough absorption of neutrons. As such, the most important criterion of a control rod is its effect on the criticality value of the core. Control rod effectiveness, referred to as the control rod worth, is therefore measured in terms of its reactivity effect on the core.

Since control rod worth represents reactivity, it is often given in units of pcm or percentage. Another commonly used unit is the dollars (\$). This unit introduces a measure of the prompt criticality of the system as a result of the reactivity change. With β_{eff} being the effective fraction of delayed neutrons, the system approaches prompt criticality when the reactivity $\rho = \beta_{eff}$. One dollar represents a reactivity equal to the effective fraction of delayed neutrons.

1.4 Introduction to Equivalence Theory

In the past, many methods have been developed to include corrections due to transport effects in the diffusion theory by combined diffusion and transport calculations. They differ mainly in the parameter that is preserved between these two calculations and the method in which they are introduced into the diffusion calculation.

The most common method used is the inner boundary condition method. In this method, the ratio of the neutron net current to the neutron flux at the boundary of the absorber region is transferred from a transport calculation to the diffusion code (see Chapter 3). Azimuthal asymmetry is not taken into account and the difference between the transport and diffusion flux is ignored. Additionally, the mesh structure of the finite difference diffusion code often raises problems related to disrectization errors. Calculations and

comparisons with experiment have shown that the method is inadequate in many cases [7], but especially when used in combination with mesh centered diffusion codes.

Two methods have been developed that aim to overcome the problems associated with the inner boundary method. They are those based on response matrix methods ([9] [10] [11]) and those that employ equivalent parameters ([7] [12]) for the region under consideration.

Response matrix methods are based on the use of nonlinear boundary conditions in the diffusion calculation on the surface of regions of great absorption. These boundary conditions are the response matrices prepared by a code, based on solving the transport equation, using a 2-D transport method. Though response matrix methods have been extensively used and shown to yield good results [12], they are difficult to incorporate into a diffusion solution.

The method of equivalent cross-sections (MECS) has been applied with success in some HTR applications [12]. The principle is to model the absorber and its environment in a transport theory method (such as S_N [13]) and then extract cross-sections and diffusion parameters from the solution that will represent the absorber region accurately in the subsequent diffusion calculation. Due to the use of equivalent parameters in the diffusion solution, the method is easily incorporated into a finite difference code system. Since MECS is the method implemented in VSOP, and currently used for PBMR design calculations, it is the method that is to be validated for the design.

Another commonly used method is that of equivalent boron (MEB). In this method the worth of a control rod is determined by some means, such as, for example, equivalent cross-section calculations or Monte Carlo calculations. Boron is then inserted into the control rod region at such a concentration that the diffusion model will yield the previously determined control rod reactivity worth. Since VSOP has several restrictions in the use of equivalent parameters when it comes to the movement of the control rods, the equivalent

boron method is often used for the determination of differential worth or other problems that make the VSOP calculations unnecessarily difficult if MECS were employed. MEB is therefore another method validated in this work.

As part of the work, a code called TOTNEW [14] was developed to calculate equivalent parameters for the control rods in VSOP. The theory that is applied is discussed in Section 3.7. This code is discussed further in Section 4.6.

1.5 Motivation for this Work and Thesis Objectives

The validation of the code system used for the design of the PBMR is of vital importance to the project as a whole and forms a requirement as part of the licensing proceedings. This thesis aims to validate the flux solver of the VSOP system, using experiments performed at the ASTRA critical facility. These experiments include criticality experiments, reactivity effects of spheres added into the core, as well as control rod worths.

The most important feature that has to be validated is the effectiveness (both considering speed and accuracy) of the current methods of control-rod worth calculations, especially due to the safety implication of such calculations. The position of the control-rods in the side reflector of the reactor introduces additional uncertainties to any method that attempts to adjust the diffusion approximation for transport effects due to the directional dependence of the flux in this area. Due to these uncertainties, the validation of VSOP and, in particular, the methods used to model the control-rods in the diffusion calculations, is the main emphasis of this thesis.

1.6 Thesis Outlay

The ASTRA critical facility configuration is described in detail in Chapter 2. Chapter 3 conveys some of the theory of reactor core calculations with a special emphasis on equivalence theory and, in particular, the equivalence theory applied in the VSOP calculations.

Chapter 4 discusses the numerical codes that are used in the calculations. This section also provides an overview of the calculational path that is to be followed to perform the benchmark calculations. Following this is Chapter 5, which discusses the actual models that were set up for each of the codes.

Chapter 6 evaluates the experimental and calculational uncertainty of results. This aims to be an indication of the accuracy of the results that are shown in Chapter 7.

Finally, Chapter 8 concludes and provides recommendations for further work. This is followed by the references. An appendix gives a listing of the computer code TOTNEW, written by me for a numerical solution of the equivalent parameters for the control rod region.

2 The ASTRA Critical Facility

2.1 Introduction

The ASTRA Critical Facility at the Russian Research Center, Kurchatov Institute in Moscow was set up to investigate the neutronic behaviour of the PBMR design. With this in mind, particular care was taken that the physical configuration of the ASTRA facility allows experiments that simulate PBMR physics at the facility.

This section describes the details of the facility with special emphasis on the factors important in producing a VSOP model.

2.2 General Configuration

The ASTRA Critical Facility consists of an upright cylinder, the side reflector of 380 cm in diameter and 460 cm in height. It contains an octagonal shaped core with a maximum extent of 175 cm. Figures 2-1 and 2-2 show a schematic of the core in both longitudinal and cross-sectional perspective. A general overview of the facility dimensions and design data is presented in Table 2-1.

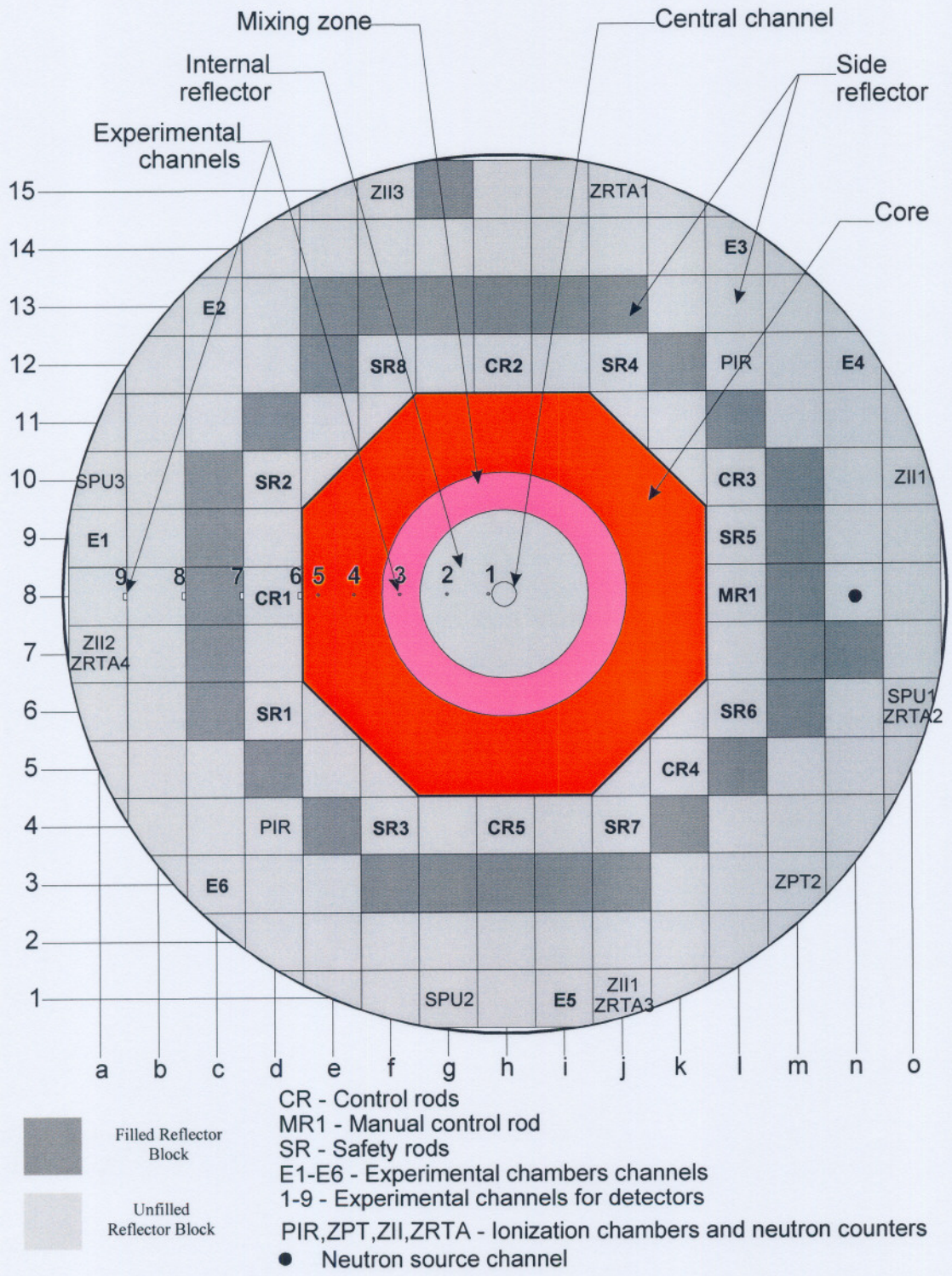

Figure 2-1: Cross-section of the ASTRA critical facility

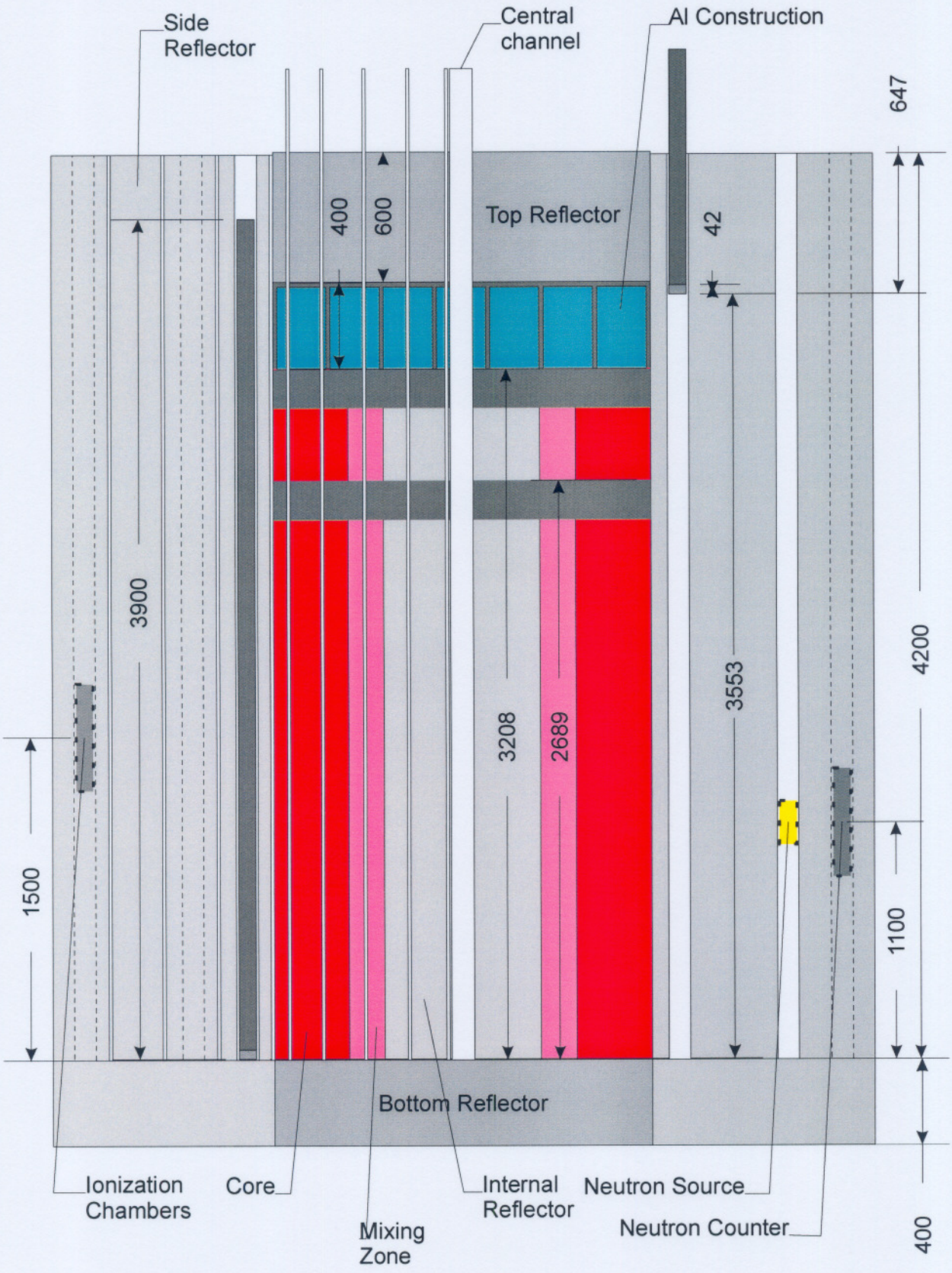

Figure 2-2: Longitudinal section of the ASTRA critical facility

Table 2-1: Overall design data of the ASTRA facility

Outer diameter	cm	380
Side reflector height	cm	460
Fuelling zones		Graphite / Mixed / Fuel
Equivalent diameter of core	cm	181
Outer diameter of mixing zone	cm	105.5
Outer diameter of inner reflector zone	cm	72.5
Inner diameter of inner reflector zone	cm	10.5
Ratio of fuel spheres (FS) to absorber spheres (AS) in fuel zone		95 / 5
Ratio of FS to AS to graphite spheres (moderator spheres – MS) in mixing zone		47.5 / 2.5 / 50
Pebble bed packing ratio		0.625
Number of control rods	#	5
Number of shutdown rods	#	8
Number of manual rods	#	1

The fuel used in the facility consists of spheres 6 cm in diameter. UO_2 kernels, 500 μm in diameter are encased in four coatings: three graphite coatings and one silicon carbide coating. Approximately 4190 of these coated kernels are then distributed in a matrix of graphite 5 cm in diameter, leading to a 2.44 g loading of Uranium at 21% enrichment. A 0.5 cm thick graphite layer then covers this graphite matrix. A graphical representation of the fuel sphere is given in Figure 2-3.

In addition to fuel spheres, two other types of spheres are used in the ASTRA facility, namely moderator spheres and absorber spheres, both 6 cm in diameter. Absorber spheres consist of graphite containing 0.1 g of Boron in the form of uniformly distributed B_4C particles in the central 4 cm diameter of the sphere. Moderator spheres consist of graphite only. Table 2-2 summarizes the different sphere types.

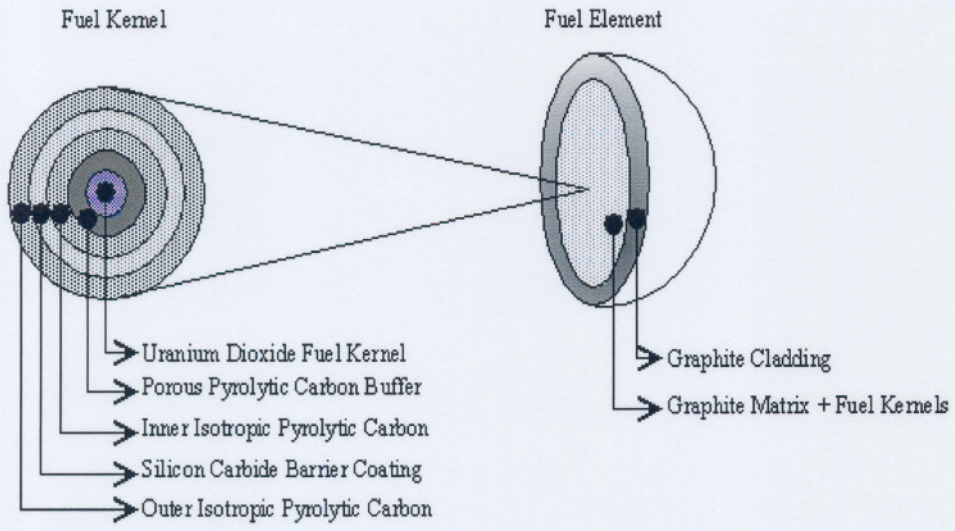

Figure 2-3: An ASTRA fuel sphere

Table 2-2: Summary of spherical elements

Sphere type	Diameter of matrix (cm)	Diameter of sphere (cm)	Density of graphite matrix (g / cm^3)	Density of graphite shell (g / cm^3)	Impurity [B(Nat) eq.] ppm by wt.	Loading (g / sphere)
Fuel sphere	5.0	6.0	1.85	1.85	1	Uranium 2.44
Moderator sphere	-	6.0	-	1.68	1	-
Absorber sphere	4.0	6.0	1.75	1.75	1	Boron 0.1

The core consists of three zones, each defined by a different ratio of sphere types. Different colors in Figure 2-1 indicate these zones. The inner reflector zone, located at the center of the core, is 72.5 cm in diameter and consists entirely of moderator spheres. Following this is a ring of 16.5 cm in thickness, the mixing zone, consisting of fuel absorber and moderator spheres, in the ratio 47.5 : 2.5 : 50 respectively. The outermost part of the core is the fuel zone, filled with fuel spheres and absorber spheres in a ratio of 95 : 5 respectively. Since the enrichment of the PBMR fuel is lower than the ASTRA fuel enrichment, the absorber spheres are added to the fuel zone in an effort to simulate the PBMR flux spectrum more closely. The random mixture of spheres at the correct ratio was obtained by mixing the appropriate quantities before loading them by hand into the relevant region. The size of each zone was ensured with the aid of a filling rig. The

measured and assumed packing ratio throughout this thesis was 0.625. The Table 2-3 lists the main features of each core zone.

Table 2-3: ASTRA core zones

Core zone	Inner diameter (cm)	Outer diameter (cm)	Sphere mixture (fuel : absorber : moderator)
Inner reflector	10.5	72.5	0 : 0 : 100
Mixing zone	72.5	105.5	47.5 : 2.5 : 50
Fuel zone	105.5	-	95 : 5 : 0

The side reflector is composed of graphite bocks 25 cm by 25 cm in cross-section with an 11.4 cm diameter central channel. This central channel is plugged for some of the blocks, as indicated by the darker areas in the side reflector in Figure 2-1. The bottom reflector is 40 cm in height. The top reflector, as indicated in the longitudinal section in Figure 2-2, is optional, and was not included in most of the experiments. Unless otherwise stated, it is assumed that the ASTRA configuration used excluded the top reflector. Table 2-4 provides general information on the reflector configuration and composition.

Table 2-4: Reflector configuration

Outer diameter of side reflector	380 cm
Equivalent inner diameter of side reflector	181 cm
Height of bottom reflector	40 cm
Height of (optional) top reflector	60 cm
Composition of reflectors	High-Purity reactor grade graphite blocks
Average effective graphite density (taking gaps into account)	1.65 g / cm^3
Equivalent natural boron concentration of impurities	0.55 ppm by weight
Square cross section of graphite blocks	25 cm x 25 cm
Axial channel diameter (hole in graphite block)	11.4 cm

The control rods (CR) and shutdown rods (SR) are situated inside certain reflector blocks as shown in Figure 2-1. Each control rod consists of a ring 76 mm in diameter, containing 15 steel rods filled with boron carbide as shown in Figure 2-4. A single manual rod (MR) is

situated in position L8. This rod does not contain any boron carbide and is present for fine reactivity adjustment requirements only. Due to its small worth, the manual rod was not considered in any of the VSOP calculations. Table 2-5 lists characteristic dimensions and properties of the control rods in the ASTRA facility.

Several experimental tubes and channels are included in the facility. They are situated both in the core region and in the side reflector (see Figure 2-1). The central experimental channel is situated in the center of the inner reflector. Experimental samples, neutron detectors and graphite plugs can be placed in this tube. Table 2-6 shows the composition of the central experimental tube.

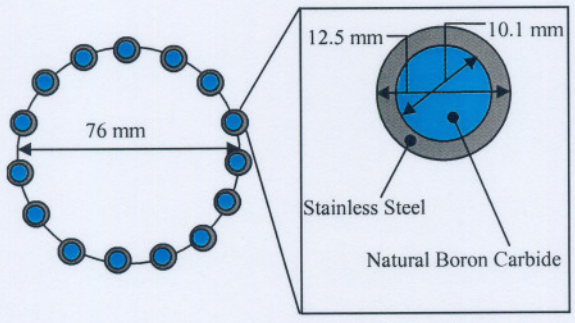

Figure 2-4: Control rod configuration

Table 2-5: Control rod configuration

Number of steel tubes	15	
Diameter of circle in which tubes are arranged (to mid-point)	7.6 cm	
Tube wall composition	Stainless Steel (12X18H10T)	
Steel density	7.9 g / cm^3	
Stainless steel composition	Element	Weight percentage
	Fe	69.1
	C	0.12
	Si	0.08
	Mn	2.0
	Cr	18.0
	Ni	10.0
	Ti	0.7
Tube wall outer diameter	1.25 cm	
Tube wall thickness	0.12 cm	
Filling material composition (filling the volume of the steel tubes)	Natural Boron Carbide	
Filling material density	1.53 g / cm^3	
Height of absorbing material	approx. 380 cm	

Table 2-6: Central experimental tube composition

Outer diameter	10.5 cm
Wall thickness	0.25 cm
Wall composition	Aluminium (Al)

A further five aluminium tubes are situated in the core along the same axial line. These tubes extend the entire height of the pebble bed and are denoted 1 to 5 in Figure 2-1.

Four small-sized experimental channels are located in the side reflector along the same radial line as the aluminium tubes. The square section of these channels is 1.5 cm x 3 cm and they are located (from the center of the core) at 88.2 cm, 113.2 cm, 138.2 cm and 163.2 cm. These channels are denoted 6 to 9 in Figure 2-1. None of the experimental channels, except for the central experimental channel, were modeled in the VSOP model of the facility. They were however included in the MCNP model. Note also that the presence of an experimental channel in position D8 causes the position of the control rod CR1 to be

shifted slightly further away from the core than the rods at what would otherwise be similar positions. This of course influences the worth measurement. Table 2-7 lists the distances from the core of each of the control rods.

Table 2-7: Control rod distances from the core

Control rod	Position	Distance from core (cm)
CR 1	D8	13.0
CR 2	h12	12.5
CR 4	K5	12.5
CR 5	H4	17.7

3 Theory

3.1 Introduction

The primary aim of neutronics calculations for nuclear reactor cores is the prediction of the power distribution for various operating conditions [16]. This problem is strictly related to the determination of the neutron flux distribution $\Phi(\underline{r}, E)$, as a function of position $\underline{r}$ and energy E in the core of specified geometry, material composition and assumed boundary condition. The basic physical model of these calculations is contained in the neutron transport equation [16], discussed in Section 3.3.

The complexity of the neutron transport equation necessitates the use of approximations to simplify the calculations. One of the most commonly used approximations is the diffusion approximation (see Section 3.5). Assumptions made here are, however, not met in and near highly absorbing regions in the core, such as for example the control rods, and as a result extensions/adjustments to the methods have to be found to include these regions. This is where the method of equivalent cross-sections, as discussed in Section 3.6, is used. Another area that provides problems in the diffusion approximation is void regions (or regions with very low material density), since here the definition of the diffusion coefficient in terms of the transport cross-section, fails. The methods used to model these regions in VSOP are discussed in Section 3.9.

3.2 Notation

The standard mathematical notation used in reactor physics is employed throughout this thesis. Although the notation is quite straight forward, some of the conventions adopted in this work are described here briefly.

Vector quantities are underlined (_) with unit vectors represented by a circumflex (^). The following quantities are often used:

Φ	scalar neutron flux distribution [$n.cm^{-2}.s^{-1}$]
D	diffusion coefficient [cm]
Σ_t	macroscopic total cross-section [cm^{-1}]
Σ_a	macroscopic absorption cross-section [cm^{-1}]
Σ_s	macroscopic differential scattering cross-section [cm^{-1}]
$\underline{r}$	position vector
$\hat{\Omega}$	direction unit vector in spherical coordinates
E	neutron energy

3.3 The Neutron Transport Equation

The main aim of any core neutronics calculation is to determine the distribution of neutrons in space, angle and energy. This detail can then yield important information such as power, reaction rates and criticality of the system. The main tool used for such an evaluation is the neutron transport equation.

The roots of transport theory go back to the Boltzmann equation, first formulated in the study of the kinetic theory of gases. The study of radiation transport in stellar atmospheres led to a number of analytical solutions of transport problems in the 1930s. Initial interest

was confined to rather simple geometries, either semi-infinite or one-dimensional. With the advent of the nuclear chain reaction, interest arose in the Boltzmann equation's application to neutron transport. Because of the complexity of reactor designs, interest was raised in the equation's solution in more complicated geometric configurations.

The derivation of the neutron transport equation is based on the conservation of neutrons in the system. The following assumptions are made:

1. Neutrons may be considered as points
2. Neutrons travel in straight lines between collisions
3. Neutron-neutron interactions may be neglected
4. Collisions may be considered instantaneous
5. The material properties are assumed to be isotropic
6. The properties of nuclei and the composition of materials under consideration are assumed to be known and time-independent unless stated otherwise
7. Only the expected or mean value of the neutron density distribution is considered. Fluctuations about the mean are not taken into account
8. Neutrons are stable

Assumption 1 is valid for all particles in which the quantum mechanical wavelength is small enough compared to the atomic diameter. This is only violated for neutrons with very low energy, of which there are very few in nuclear reactor cores. The second assumption is valid when considering neutral particle transport where there is no effect of long-range electric and magnetic forces. Since the neutron density in the reactor is small compared to the density of the atoms of the structural material, it is acceptable to assume that neutron-neutron interactions play a minor role.

Taking these assumptions into account, the angular neutron flux distribution $\Psi(\underline{r},E,\hat{\Omega},t)$ in units of neutrons.cm^{-2}.s^{-1} is governed by the neutron transport equation

$$\frac{1}{v}\frac{\partial\Psi(\underline{r},E,\hat{\Omega},t)}{\partial t}+\hat{\Omega}\cdot\nabla\Psi+\Sigma_t(\underline{r},E)\Psi(\underline{r},E,\hat{\Omega},t)= \\ +\int_{4\pi}d^2\hat{\Omega}'\int_0^{\infty}dE'\Sigma_s(\underline{r},E'\rightarrow E,\hat{\Omega}'\rightarrow\hat{\Omega})\Psi(\underline{r},E',\hat{\Omega}',t)+S(\underline{r},E,\hat{\Omega},t) \tag{3-2}$$

where $S(\underline{r},E,\hat{\Omega},t)$ is a neutron source term, with the initial condition

$$\Psi(\underline{r},E,\hat{\Omega},0)=\Psi_0(\underline{r},E,\hat{\Omega}) \tag{3-3}$$

and boundary condition

$$\Psi(\underline{r}_S,E,\hat{\Omega},t)=0 \quad for \quad \hat{\Omega}\cdot\hat{e}_S<0, \quad \underline{r}_S\in S \tag{3-4}$$

where $\underline{r}_s$ denotes a point on the surface S of the problem domain.

For many reactor calculations the details of the angular dependence of the neutron flux is not necessary. Integrating the neutron transport equation over all angles and defining the neutron flux $\Phi(\underline{r},E,t)$ as

$$\Phi(\underline{r},E,t) \equiv \int_{4\pi} d\hat{\Omega}\Psi(\underline{r},E,\hat{\Omega},t) \tag{3-5}$$

yields the neutron continuity equation

$$\begin{aligned} &\frac{1}{v}\frac{\partial\Phi(\underline{r},E,t)}{\partial t} + \nabla\cdot\underline{J}(\underline{r},E,t) + \Sigma_t(\underline{r},E)\Phi(\underline{r},E,t) = \\ &\int_0^\infty dE'\Sigma_s(\underline{r},E'\to E)\Phi(\underline{r},E',t) + s(\underline{r},E,t) \end{aligned} \tag{3-6}$$

where $s(\underline{r},E,t)$ is the angular integrated source term and $\underline{J}$ is the neutron current density defined by

$$\underline{J}(\underline{r},E,t) \equiv \int_{4\pi} d\hat{\Omega}\hat{\Omega}\Psi(\underline{r},E,\hat{\Omega},t) \tag{3-7}$$

Thus an additional unknown variable has been introduced in the process of removing the angular dependence. Since it is impossible to express $\underline{J}$ in terms of Φ in a general and exact manner, this equation cannot be solved directly.

As can be seen the neutron transport equation is an integro-differential equation. It depends on three spatial variables, two angular variables, energy and time. Spatial details in a reactor core can be very involved, with complex geometries and a large number of regions with different material parameters. Typical neutron energies range from 20 MeV down to 0 MeV, and time in a typical reactor cycle can be several months. Direct analytical

solutions are only possible for the simplest of problems. Instead, various other methods have been devised to obtain a numerical solution of the equation.

3.4 Numerical Solutions to the Neutron Transport Equation

Numerical solutions of the neutron transport equation can broadly be divided into two groups: stochastic and deterministic methods.

Stochastic or Monte Carlo methods are based on the simulation of a large but finite number of random neutrons and their motion through the system. Results such as the flux distribution, reaction rates etc. are then determined through statistical averaging of these particle histories. Monte Carlo methods have the advantage of allowing for full 3-dimensional modeling of the geometric configuration and the probabilistic nature allows for the use of continuous energy cross-section data. One of the best known Monte Carlo based codes available currently is MCNP, which is used in this work to add further verification to the VSOP results.

The deterministic methods are based on the discretization of the independent variables of the transport equation in order to obtain a set of coupled equations that can easily be solved numerically by various methods such as, for example, finite difference schemes. Virtually all deterministic computational methods discretize the energy to obtain the multi-energy-group approximation [13]. Temporal variations are normally ignored, thus assuming a steady state solution validated by the use of an eigenvalue, most commonly the critical eigenvalue. In terms of the position and angular variables, several methods exist such as S_N and Collision Probability methods.

When considering a relatively simple system in one or two dimensions, it is possible to solve the neutron transport equation for discrete angles and then to apply a compatible

quadrature term to the integral. This method is commonly known as the discrete ordinates method or S_N method, where the N denotes the number of ordinates used. It is characterized by simplicity of derivation, and it leads to algorithms of notable computational efficiency, especially where memory is considered. The two-dimensional transport code used in the evaluation of the equivalent cross-sections to be used in VSOP is based on this method of solution.

3.5 Diffusion Theory Approximation

Though numerical solutions of the transport equation are possible, they are too time consuming even on modern computers to make them viable for day-to-day design calculations of the full reactor core. This necessitated the introduction of an approximation that would allow for a faster solution of the problem, while still providing solutions to the required accuracy.

The diffusion approximation is one of the approximations that has been developed to simplify the solution of the neutron transport equation. In this approximation a simple relationship between the current and the neutron flux is assumed.

As a simplification, consider the single energy approximation. The continuity equation 3-6 is then given by

$$\frac{1}{v}\frac{\partial\Phi(\underline{r},t)}{\partial t}+\nabla\cdot\underline{J}(\underline{r},t)+\Sigma_a(\underline{r})\Phi(\underline{r},t)=s(\underline{r},t) \qquad (3\text{-}8)$$

where

$$\Sigma_a(r) = \Sigma_t(r) - \Sigma_s(r) \tag{3-9}$$

If it is assumed that the angular neutron flux is only linearly anisotropic, equation 3-8 can be multiplied by $\hat{\Omega}$ and integrated over all angles to obtain

$$\frac{1}{v}\frac{\partial \underline{J}(\underline{r},t)}{\partial t} + \frac{1}{3}\nabla\Phi(\underline{r},t) + \Sigma_{tr}(\underline{r})\underline{J}(\underline{r},t) = \underline{S}_1(\underline{r},t) \tag{3-10}$$

where

$$S_1 \equiv \int_{4\pi} d\hat{\Omega}\hat{\Omega}s(\underline{r},E,\hat{\Omega},t)$$

$$\Sigma_{tr}(\underline{r}) \equiv \Sigma_t(\underline{r}) - \bar{\mu}_0\Sigma_s(\underline{r})$$

$\bar{\mu}_0$ = average scattering angle cosine

Equation 3-7 and equation 3-9 are known as the P_1 equations since the approximation of linear anisotropic angular dependence in the flux is equivalent to expanding the angular flux in Legendre polynomials and retaining the first order terms.

In principle, the P_1 equations could be used to describe the neutron behavior in the reactor. In practice two more assumptions are made to simplify the calculations. Firstly, the source term $S(\underline{r},E,\hat{\Omega},t)$ is assumed to be isotropic, an assumption usually of reasonable validity

in nuclear reactor studies. The S_1 term then vanishes in equation 3-10 for the current density. Secondly, it is assumed that the time derivative of the neutron current is small compared to the other terms in the equation. This would imply for example that

$$\frac{1}{|\underline{J}|}\frac{\partial|\underline{J}|}{\partial t} << v\Sigma_t \tag{3-11}$$

that is, that the rate of time variation of the current is much slower than the collision frequency $v\Sigma_t$. Since $v\Sigma_t$ is typically of the order of 10^5 s^{-1} or larger, only an extremely rapid time variation of the current would invalidate this assumption. With these assumptions equation 3-10 can be rewritten as

$$\frac{1}{3}\nabla\Phi(\underline{r},t)+\Sigma_{tr}(\underline{r})\underline{J}(\underline{r},t)=0 \tag{3-12}$$

Solving for the current density

$$\underline{J}(\underline{r},t)=-\frac{1}{3\Sigma_{tr}(\underline{r})}\nabla\Phi(\underline{r},t) \tag{3-13}$$

If we define the neutron diffusion coefficient D by

$$D(\underline{r})\equiv[3\Sigma_{tr}(\underline{r})]^{-1}=[3(\Sigma_t(\underline{r})-\bar{\mu}_0\Sigma_s(\underline{r}))]^{-1} \tag{3-14}$$

then

$$\underline{J}(\underline{r},t) = -D(\underline{r})\nabla\Phi(\underline{r},t) \tag{3-15}$$

Hence, in certain situations the neutron current density is proportional to the spatial gradient of the flux. This very important relation arises quite frequently in other areas of physics, where it is known as Fick's Law. Physically, the law implies that a spatial variation in the neutron flux (or density) will give rise to a current of neutrons flowing from regions of high density to low density.

By substituting equation 3-15 into equation 3-8 the one-speed diffusion equation is obtained

$$\frac{1}{v}\frac{\partial\Phi(\underline{r},t)}{\partial t} + \nabla\cdot D(\underline{r})\nabla\Phi(\underline{r},t) + \Sigma_a(\underline{r})\Phi(\underline{r},t) = s(\underline{r},t) \tag{3-16}$$

In summary, the following assumptions had to be made to obtain the one-speed diffusion approximation:

- Angular flux is linearly anisotropic
- Isotropic sources
- The neutron current density changes slowly on a time scale compared to the mean collision time

The question arises as to when the angular flux is sufficiently weakly dependent on angle for the diffusion approximation to apply. More detailed studies of the transport equation itself indicate that the assumption of weak angular dependence is violated in the following cases:

- Near boundaries or where the material properties change dramatically from point to point over distances comparable to the mean free path
- Near localized sources
- In strongly absorbing media

In fact, strong angular dependence can be associated with neutron fluxes having a strong spatial variation. This is where equivalent methods have to be introduced to be able to model such regions inside the diffusion approximation.

The above derivation can be extended to multiple energy groups, and it is in this form that it is implemented in VSOP, which uses mesh-centered finite difference methods to solve the multi-group diffusion equation.

3.6 Equivalence Theory

As is discussed in Section 3-5, diffusion theory assumes that the spatial neutron flux variation in any region is small. This is, however, not the case in and close to a strong absorber where there will be a sudden decrease in the flux. In most reactors such regions are small compared to the overall reactor size, and since the diffusion theory is valid at a distance from the absorber region, and transport methods are too time and memory

consuming to be useful for full core calculations, diffusion codes are still generally used for core design calculations today. Additionally, for graphite-moderated systems like the PBMR and ASTRA, most of the transport codes show either very slow convergence (S_N) or take an unacceptable amount of running time (Monte Carlo). Methods were therefore developed to accurately represent absorber regions in the diffusion calculations through the use of combined transport and diffusion methods.

The basic aim of the combined transport and diffusion methods is to 1) describe the absorber region and, optionally, some of its surrounding and to solve the system using a transport code; 2) extracting some macroscopic data for this region from the transport calculation; and 3) to use these data in a subsequent 2-D or 3-D diffusion calculation for the whole reactor. Since in general some manipulation of the cross-section data is performed, based on the transport solution, these are known as equivalent parameters.

Various methods exist for the determination of equivalent parameters. They differ mainly in the parameter that they try to preserve in moving between the transport and diffusion calculation, or alternatively, in the implementation of such parameters. One well-known and often used method is that of inner boundary conditions. Here the ratio of the net current to the flux at the boundary of the absorber region is determined from a transport calculation and passed to the diffusion code. This, in essence, preserves the leakage from the absorber region in the diffusion approximation. Since the method considers only the net current and average flux, no azimuthal asymmetry is taken into account. In addition, the difference between transport and diffusion flux is neglected, and the finite mesh structure of the diffusion code often raises problems related to discretization errors. It has been shown previously [7] that this method is inadequate in many cases, especially in mesh centered diffusion codes such as CITATION, the diffusion code used in VSOP, where major problems may arise.

The method of equivalent cross-sections ([7] [12]) aims to overcome some of the problems stated above. Instead of preserving the leakage out of the absorber region, it maintains the

reaction rates between the transport and diffusion methods. In its current form, the method does not account for azimuthal effects for absorbers outside the core, however, this is a restriction due to the use of a 1-D transport code, rather than inherent in the method itself. This is the method used for VSOP calculations of control rods, and is discussed in more detail in Section 3-7.

Similar to the method of equivalent cross-sections, the response matrix method ([9] [10] [11] [12]) aims to preserve the reaction rates in the transport and diffusion codes. Since a response matrix can easily be implemented to include azimuthal effects, this is taken into account, provided that a 2-D transport method is used which retains the information on the non-isotropic flux shape relative to the absorber center. The calculational effort in this case is more involved than for the method of equivalent cross-sections and since it is currently not possible to include such a response matrix in VSOP calculations, this method was not investigated further.

Both the method of equivalent cross-sections and that of response matrices have been validated to some extent for the use of high-temperature pebble bed reactors. An additional method that has been validated and proven useful is that of equivalent boron as discussed in Section 3-8.

3.7 Method of Equivalent Cross Sections

The method of equivalent cross-sections is based on preserving the reaction rate of the absorber region for the diffusion solution by relating the leakage to the reference transport solution.

The following assumptions have to be made for the method:

1. the volume of the control rod region is the same in the transport and diffusion calculation;

2. equivalence between transport and diffusion flux at some point outside the absorber region surface;

3. flux equivalence at points of equal distance from the absorber region, independent of absorber geometry.

Volume preservation is only necessary to maintain material balance. Other methods such as surface-preserving ones may be possible.

Assumption 2 asserts that the difference between the transport flux and the diffusion flux should be negligible. This means that the neighboring mesh has to be at a suitable distance and be of suitable size. Should this not be the case, the method may produce negative values for the diffusion coefficients in the rodded area.

Assumption 3 is necessary since anisotropic effects cannot be accounted for in a 1-D transport calculation. Should, at some stage, the 1-D transport code be replaced by a 2-D method, this assumption may be removed. This is an area of future investigations.

Since we are attempting to equate leakage rates in the two methods, it is important to take into account the solution method implemented in the diffusion code. In the method described here, it is assumed that a single mesh point in the diffusion calculation that is based on a mesh-centered algorithm, as is the case in CITATION, must represent the absorber region.

The ASTRA facility is modeled in cylindrical geometry (R-Z-θ) in VSOP and CITATION. However, the absorber regions, located in the outer reflector, can be approximated as a rectangular shape in R-θ meshes, located far from the center (large R) and appropriately selected angles. The following derivation, given in Cartesian geometry (X-Y) is therefore applicable to the outer reflector positions of the ASTRA facility.

In Figure 3-1 a geometrical representation of the equivalent absorber region (shaded region) and its immediate environment are given. Note that the absorber region is represented by a rectangle with dimensions (a, pa). The two neighboring meshes in the x-direction (referred to as X neighbors) both have a mesh size of (p_xa, pa) and distance Δ_x to the finite difference mesh center. The dimensions for the neighboring meshes in the y-direction (Y neighbors) are as indicated.

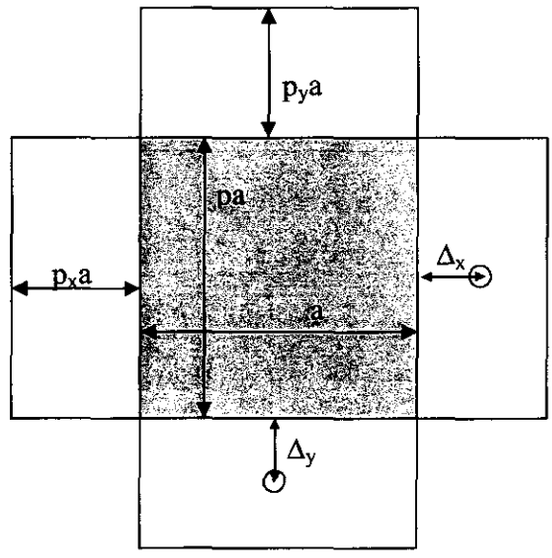

Figure 3-1: Geometric relations in CITATION

Figure 3-2 represents the equivalent control rod model in the 1-D transport calculation. Note that the shaded region in both this representation and the absorber region in the diffusion calculation (shaded region in Figure 3-1) must have the same volume ($pa*a = \pi*R_A^2$). The transport calculation must also be subdivided in such a way that mesh points exist at distances from the absorber region that are equal to the neighbor sizes Δ_x and Δ_y, indicated by R_x and R_y in Figure 3-2 respectively.

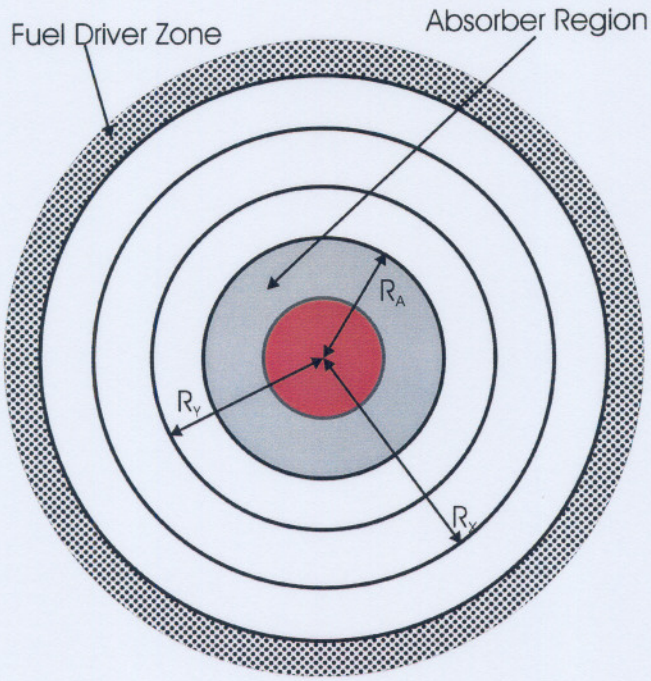

Figure 3-2: Equivalent control rod model in the 1-D transport code

The leakage in the diffusion method, as implemented in CITATION in X-Y geometry, is defined by

$$L = \frac{F_y\left(\Phi_x - \Phi_a\right)}{\frac{\delta_x}{D_a} + \frac{\Delta_x}{D_0}} + \frac{F_x\left(\Phi_y - \Phi_a\right)}{\frac{\delta_y}{D_a} + \frac{\Delta_y}{D_0}} \qquad (3\text{-}17)$$

where

$\Phi_x = \Phi$ (x-neighbor)

$\Phi_y = \Phi$ (y-neighbor)

$\Phi_a = \Phi$ (absorber region)

Φ = diffusion flux from CITATION

F_y = surface to x-neighbor $= 2pa$

F_x = surface to y-neighbor $= 2a$

δ_x = distance from absorber center to x-surface $= \frac{1}{2}a$

δ_y = distance from absorber center to y-surface $= \frac{1}{2}pa$

Δ_x = distance from x-surface to x-neighbor center $= \frac{1}{2}p_x a$

Δ_y = distance from y-surface to y-neighbor center $= \frac{1}{2}pp_y a$

a, p, p_x, p_y = parameters of CITATION mesh geometry (see Figure 3-1)

D_a = diffusion coefficient in the absorber region

D_0 = diffusion coefficient in neighbors

Making this leakage equal to the transport leakage and using assumption 2 at points R_x and R_y such that

$$\Phi_x = \Psi(R_x) \text{ and } \Phi_y = \Psi(R_y), \tag{3-18}$$

where

Ψ = transport flux from 1-D S_N calculation,

R_x, R_y = radii in S-N calculation equivalent to distance of diffusion neighboring mesh point to absorber mesh,

leads to the equivalent diffusion coefficient

$$\frac{1}{D_a} = 2p\frac{\Phi_x - \Phi_a}{L} + \frac{2}{p}\frac{\Phi_y - \Phi_a}{L} - \frac{p_x + p_y}{2D_0} + \sqrt{R} \tag{3-19}$$

where

$$R = \left(2p\frac{\Phi_x - \Phi_a}{L} + \frac{2}{p}\frac{\Phi_y - \Phi_a}{L} + \frac{p_x - p_y}{2D_0}\right)^2 + 4p\left(p_y - p_x\right)\frac{\Phi_x - \Phi_a}{L}\frac{1}{D_0} \tag{3-20}$$

The macroscopic absorption and removal cross-sections for the equivalent absorber region are determined from flux-volume weighting of the selected regions in the 1-D transport model.

3.8 Method of Equivalent Boron Concentration

The use of the equivalent boron concentration method is one of the simplest methods to model absorber regions in diffusion methods, and the most easily implement in any diffusion code. It assumes that the worth of the absorber region in terms of its effect on the system as a whole is known, or can be determined through the use of a transport code.

Boron is then added into the absorber region of the diffusion calculation at such a concentration as to preserve the same reactivity worth. By definition this method preserves the effectiveness of the absorber region; however the flux distribution and reaction rates from the diffusion calculation are not the correct ones. The advantage of this method over MECS however is that it is possible to have subdivisions of the control-rod region and that no restrictions are placed on the sizes of the neighboring meshes.

3.9 Void Area Treatment by Diffusion Theory

Similar to control-rod regions, void regions also provide a problem in diffusion calculations. The definition of the diffusion coefficient in terms of the inverse of the transport cross-section fails in the case where the transport cross-section is zero. It is, therefore, again necessary to find a relation between a transport solution and the diffusion solution to model these regions.

Several methods exist that can be used to model void areas. The simplest one is to model the void area as containing a material at very low density. The problem with this method is that there is a limit to how dilute the material can be defined before the diffusion method fails. An alternative to using a low-density material is to use a response matrix as determined from a transport solution. The problem with this method, as was the case for the response matrix method for the control rods, is that VSOP cannot handle the insertion of a response matrix into the calculation. This option, though found to be efficient in other investigations [12], can therefore not be implemented in VSOP without major changes to the code. Since VSOP can use manual input of cross-section data and diffusion coefficients, another method was developed that equates the transport and diffusion currents under certain very limiting circumstances [30]. This method is discussed in more detail in Section 3.9.1. The restrictions under which the aforementioned theory is valid include a relatively large void cavity and the fact that the void area must be completely surrounded by graphite. Since this is not the case in any of the void areas in the ASTRA facility, a different method had to be investigated. Milgram's method [17] of estimating the axial diffusion coefficient by analog Monte Carlo methods, as discussed in Section 3.9.2

was therefore used. Note that this method is for now only applied to the axial diffusion coefficients

3.9.1 Treatment of the Void Area by Diffusion Theory

Consider a void cylindrical cavity delimited by the pebble bed core at the bottom and by graphite on all other sides. Two dimensional transport calculations have shown that the flux in such an area is fairly uniform. The scattering and reaction cross-sections in the cavity are zero. Assuming that "diffusion coefficients" exist for the region, the diffusion equation in two-dimensional cylindrical geometry is

$$\left(D_z \frac{\partial^2}{\partial z^2} + D_r \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial}{\partial r} \right) \right) \Phi(r,z) = 0 \tag{3-21}$$

where D_z and D_r are anisotropic diffusion coefficients to be determined. Considering a flat flux distribution, a Taylor expansion of the flux can be used, given to second order by

$$\Phi(r,z) = \Phi_0 + \Phi_1 z + \Phi_2 \left(\frac{z^2}{2D_z} - \frac{r^2}{4D_r} \right), \tag{3-22}$$

where the constants Φ_i can be determined by boundary conditions. They can be interpreted in terms of flux averaged values, axial current, axial in-streaming, radial leaving current, etc.

It is now assumed that the transport and diffusion fluxes correspond to each other when the current directed into the cavity is the same at each point of the surface. In diffusion theory, the incoming current is defined by

$$J_{in}(r)=\frac{\Phi(r)}{4}+\frac{J(r)}{2} \tag{3-23}$$

where $J(r)$ is the net current. In transport theory the definition becomes more complicated, since the angular transport flux distribution must also be considered. It is, however, possible to obtain an expression for the z-component of the current, provided it is assumed that the transport solution is rather isotropic, which is the case when the cavity is built up from a large amount of graphite. The diffusion incoming current (3-22) derived from the Taylor expansion (3-21) can then be inserted into this transport solution to obtain a complicated expression dependent on the Taylor expansion coefficients and on the position r. Similarly, the Taylor expansion of the flux can be used to determine the diffusion current.

In general there is no agreement between these two solutions, and it is therefore necessary to optimize the agreement of both theories for the first Φ_i with respect to an integral goal value, which is taken to be the net current between the surface of the core and the cavity. Equality of the diffusion and transport value of this current for the linear component Φ_1 gives a determining equation for the z-directed diffusion constant.

$$D_z = R[(h+4)(h^2+4)^{1/2}-8+4h-h^2]/12h \tag{3-24}$$

where R is the radius of the cavity and h is the ratio between its height H and radius R. This expression depends only on the geometric properties of the cavity.

Similarly, using the quadratic term Φ_2

$$D_r = \frac{\frac{3}{2} RD_z h^2 (h(h^2+4)^{1/2} - h^2 - 2 + 4\ln\{[h+(h^2+4)^{1/2}]/2h\})}{\left((24+14h+2h^2+h^3)(h^2+2)^{1/2} - 48 - 28h - 16h^2 - 2h^3 - h^4 + 24\ln\{[h+(h^2+4)^{1/2}]/2\}\right)}$$

The values of D_z / R and D_r / R for a range of h between 0.1 and 2.0 are shown in Figure 3-3.

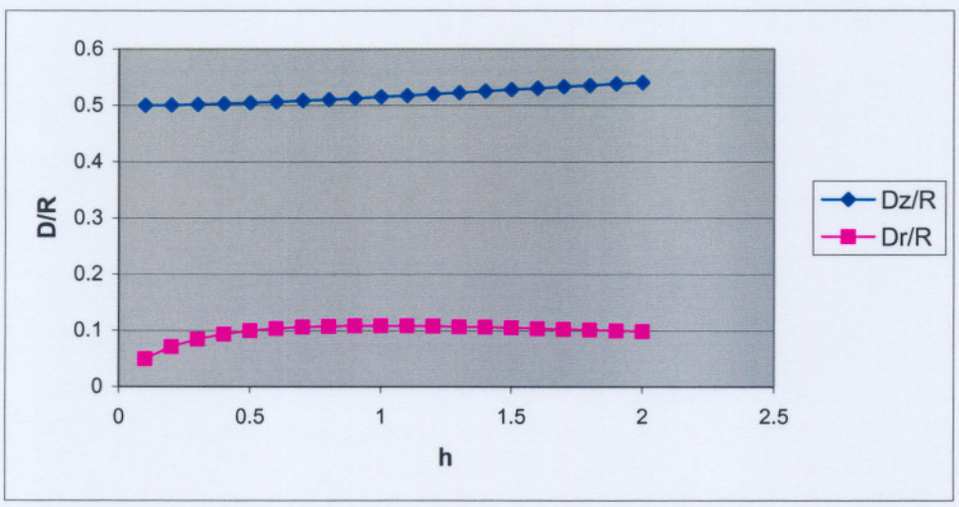

Figure 3-3: Radial and axial diffusion coefficients

The axial diffusion coefficient in this range is close to 0.5 and the radial coefficient has an average value of around 0.1. It must be noted that the general trends observed here do not apply once the ratio of height to radius increases well beyond 2.5.

3.9.2 Determination of Axial Diffusion Coefficients Using Analogue Monte Carlo Methods

The method discussed in Section 3.9.1 only applies in very specific situations. If we consider the ASTRA facility, a graphite reflector does not terminate the cavity above the core, except in the few cases when the top reflector is included in the configuration. The same applies for the control rod channels and central experimental channel, with the added difficulty that due to the height of these channels, the assumption of a flat flux profile in the axial direction is incorrect. In the radial direction this problem is not as pronounced.

As a result of the above-mentioned problems, the diffusion coefficient in the axial direction was determined by another method employing Monte Carlo techniques.

Consider the diffusion approximation. The basic assumption is that there is a relationship between the current and the flux given by (3-14)

$$\underline{J}(\underline{r}) = -D(\underline{r})\nabla\Phi(\underline{r}) \tag{3-25}$$

where $D(\underline{r})$ is defined by (3-13). Since $D(\underline{r})$ is dependent only on material composition at point $\underline{r}$ and only changes its value as a function of position to the extend that material changes occur at different positions, the above equation can be rewritten as

$$\underline{J}_i = -D_i(\nabla\Phi)_i \tag{3-26}$$

where the subscript i refers to the ith homogeneous material region defining the surface over which $\underline{J}_i$ and $(\nabla\Phi)_i$ are evaluated and averaged, and D_i becomes a constant over surface i. This can now be generalized to the union of many of many regions (area A_i) of differing materials such that

$$\underline{J} = \sum_i \underline{J}_i A_i = \sum_i D_i A_i (\nabla\Phi)_i \tag{3-27}$$

Considering only a single direction, the net current J is defined by

$$J = \sum_i J_i A_i = \sum_i D_i A_i (\nabla\Phi)_i \tag{3-28}$$

A Monte Carlo method such as MCNP can now be used to determine the current and flux gradient in (for example) the axial direction for a region and hence determine the axial diffusion coefficient.

There are two important points that have to be noted:

1. The net current J is the difference between the opposing currents crossing a surface. As a result any determination of J will be subject to considerable error because of statistical fluctuations in the partial currents.

2. The gradient of the flux is not in general one of the results that can be obtained from a Monte Carlo code and must therefore be obtained from a 'best' fit to a model flux distribution.

Thus, the determination of D requires a physical model that will generate both a set of currents crossing a surface such that a difference can be computed with a meaningful accuracy, and a flux that can be fitted with sufficient accuracy to obtain a gradient.

The method described above provides a means to determine diffusion coefficients in void areas without the assumptions and restrictions imposed by the previously described method (Section 3.9.1). In particular, it allows for the determination of diffusion coefficients in the voids of the control rod regions, which clearly do not meet the requirements of the previous method.

4 Numerical Codes

4.1 Introduction

In Chapter 3 the theory required to perform core neutronics calculations, and in particular to determine equivalent parameters for control rod models, was discussed. In this chapter the numerical codes used to perform such calculations are considered in more detail. In Chapter 5 the actual models used for each of the codes are discussed.

4.2 Calculational Path

Figure 4-1 shows the calculational path that is followed in general to perform a VSOP calculation without control rods included, as well as the calculational path required to follow when control rods are considered in the VSOP calculation.

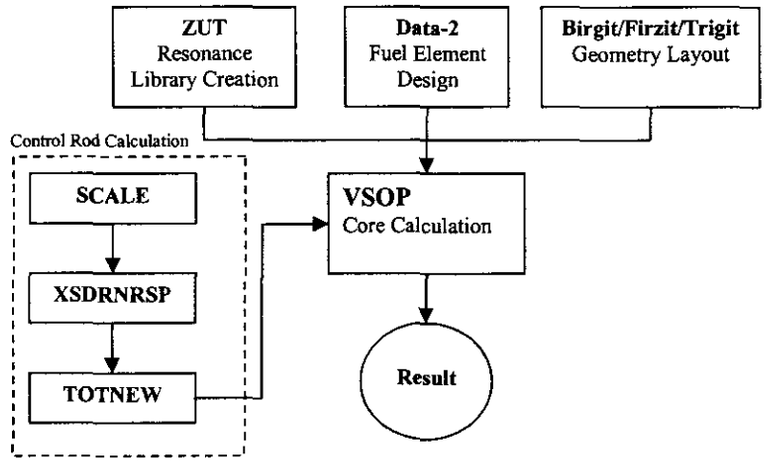

Figure 4-1: Calculational path

In performing a general VSOP calculation, three codes have to be run prior to the main part, namely ZUT, BIRGIT/TRIGIT/FIRZIT and DATA2. The ZUT code generates a resonance integral library to be used in the spectrum calculations. BIRGIT, FIRZIT and TRIGIT set up the geometry specification. The code used will depend on whether the calculation involves cylindrical or Cartesian coordinates, and on the number of dimensions. Once these have been executed, the main VSOP program can be run, which has several internal subsections as discussed in detail in Section 4.3.

In adding the data for the equivalent control rod cross-sections, further steps have to be taken. First, a working library has to be created using the SCALE system. Following this, the control rod model is calculated in the XSDRNRSP code. The results from the calculation are passed to TOTNEW, where the equivalent parameters are calculated. These are then manually entered into the VSOP input and the VSOP run is performed as discussed above.

4.3 VSOP

VSOP (Very Superior Old Programs) [2] is a system of computer codes for the numerical simulation of nuclear reactor physics performance. It has special features that allow for the simulation of high temperature gas cooled reactors with spherical fuel elements.

The version of VSOP used in this dissertation is VSOP (94), though it includes additions that were made by various contributors in order to model the PBMR core.

As the name suggests, VSOP consists of several, relatively old, and as a result well-tested computer codes, which have been linked together to form a complete package. Figure 4-2 gives an indication of the data flow in VSOP through these various codes.

The main code in the package is itself called VSOP. VSOP combines several modules, including the finite difference diffusion code CITATION and the spectrum codes GAM and THERMOS. VSOP allows following the reactor life from its start-up through the running-in phase towards an equilibrium cycle. It handles the fuel shuffling through the use of so-called batches that are moved through the core and stored in storage bins to be either removed or to re-enter the system. From now on VSOP refers to the code itself, whereas the entire code system will be referred to as the "VSOP package".

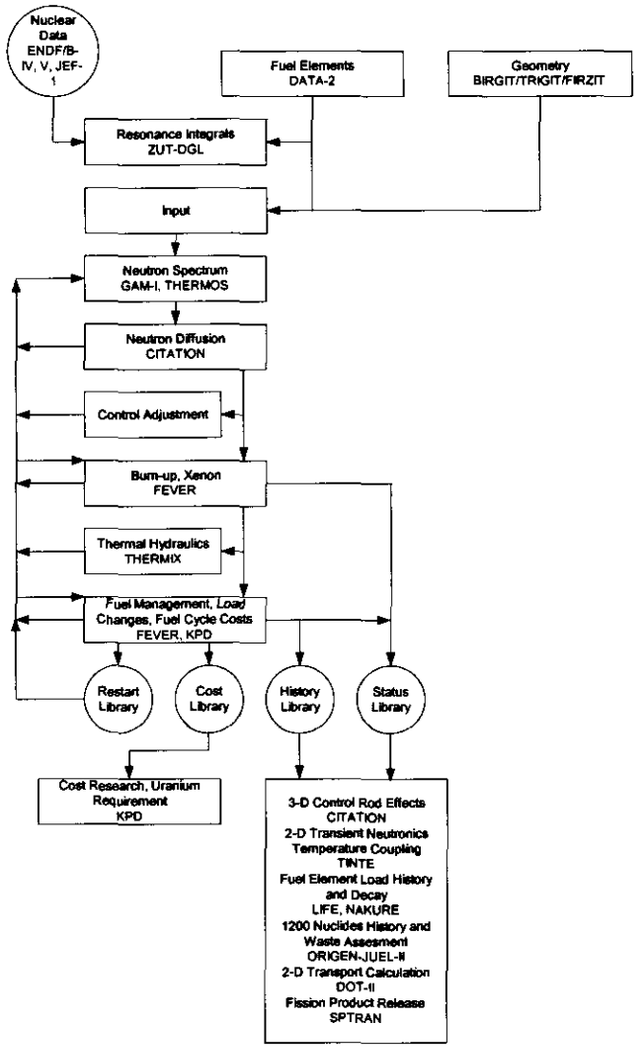

Figure 4-2: Data and program flow in VSOP

The geometric design of the reactor under consideration is set up using one of the codes BIRGIT (2-D cylindrical geometry), TRIGIT (3-D Cartesian geometry) or FIRZIT (3-D cylindrical geometry). These codes handle, for example, the evaluation of the flow channels, as well as the generation of a fine mesh over the specified user mesh for the flux calculation, which is then passed to VSOP. The difficulty in modeling a pebble bed reactor is twofold. For one, there is a mixture of pebbles of different burnup in the core at any particular time. Secondly, the pebbles are moving downward in a certain flow pattern, which will be reactor dependent at all times. Both these features place extra difficulties on the geometry specification. In the VSOP package the core is modeled as consisting of channels which have a predefined shape, depending on the flow pattern. Each channel is then subdivided into a given number of layers. The downward motion of each of these layers simulates fuel movement. Each layer in turn is made up of one or more batches, each batch having one particular composition. This allows for the mixing of fuel of different burnup inside the core. Note that the reflector region or any other none-core areas are also specified by what are known as layers. For the neutron flux calculation the material data for each layer is mixed and passed on from VSOP to the CITATION code. CITATION only allows for Cartesian or cylindrical geometry and can, as a result, not model the flow channels explicitly. BIRGIT/FIRZIT/TRIGIT is therefore responsible for providing an interface between the actual reactor layout and the CITATION model in the form of a volume matrix.

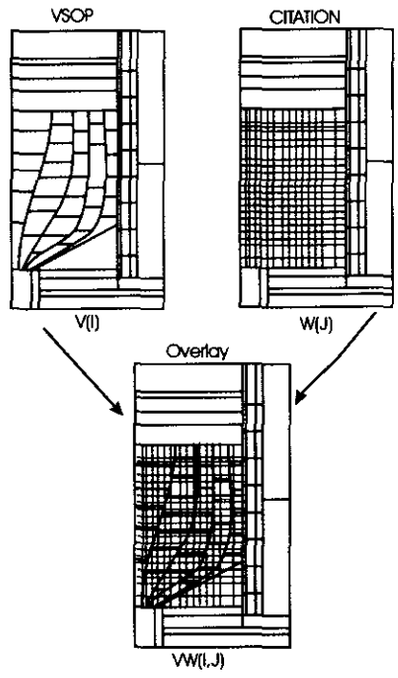

Figure 4-3: Volume matrix creation in VSOP

FIRZIT initially generates the VSOP layers V(I) as shown in Figure 4-3, as well as the CITATION coarse mesh W(J). Note that this coarse mesh is further subdivided into a fine mesh for the actual finite difference calculation. FIRZIT then fits the VSOP layer mesh over the CITATION mesh to generate a volume matrix VW(I,J) that allows for material data to be transferred between VSOP and CITATION. Macroscopic cross-section data is generated for the VSOP batches and thereafter for the VSOP layers, Σ(I). These are then converted to macroscopic cross-sections Σ(J) of the CITATION composition by

$$\Sigma(J)=\frac{\sum_{I}\Sigma(I)\cdot VW(I,J)}{W(J)}. \qquad (4\text{-}1)$$

Neutron fluxes $\Phi(J)$ of the CITATION composition are transformed to fluxes $\Phi(I)$ of the VSOP mesh by

$$\Phi(I) = \frac{\sum_J \Phi(J) \cdot VW(I,J)}{V(I)} \tag{4-2}$$

This flux is then applied for burnup calculations of the VSOP batches.

The fuel element design is set up using the code DATA-2. This code was specially designed for spherical fuel elements. Information from DATA-2 is passed to the ZUT code to obtain resonance integrals for the isotopes ^{232}Th and ^{238}U at various temperatures for all required fuel types. Due to the double heterogeneity of the fuel, special features had to be added to ZUT to correctly calculate these integrals. A standard resonance integral calculation, employing the Nordheim integral treatment [18] involves the solution for the energy dependence of the neutron in a material region containing a resonance absorber and a maximum of two admixed moderators. The presence of more than one absorber lump in the moderator medium is accounted for through the use of a Dancoff factor, related to the first flight escape probability from the absorber region given by

$$P(E) = \frac{P_0(E) \cdot (1-C)}{1 - \left(1 - \bar{l}\Sigma_a(E) P_0(E)\right) \cdot C}, \tag{4-3}$$

in which C is the Dancoff factor and $P_0(E)$ is the probability for a neutron to escape the lump of its birth. This method has proven to be a good approximation for lumps of all degrees of grayness. In the case of the coated particle fuel however, due to the small size of the kernels, the first-flight escape probability $P_0(E)$ is close to 1 even for neutrons close to the peak of a strong resonance. In addition, a neutron may travel through many coated particles without undergoing a collision. In fact, there are as many as 8 possibilities in which the neutron can escape from the coated particle, as indicated by W1 to W8 in Figure 4-4. Note that the top diagram indicates escape probabilities from a single kernel to either the cladding of its own sphere or regions in surrounding spheres. The bottom diagram

indicates how all these escape probabilities, as well as the ones inside a single kernel and its cladding, can be represented in a single fuel sphere.

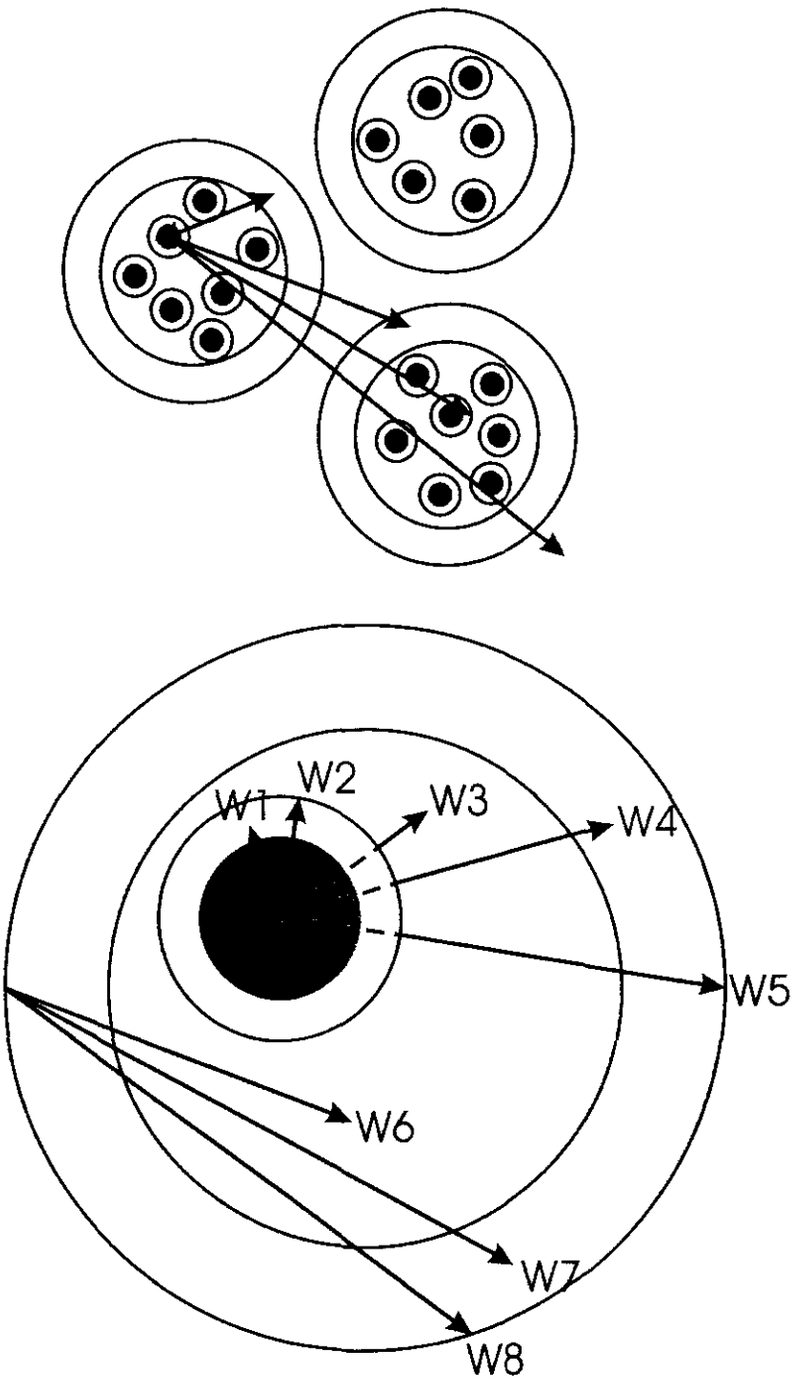

Figure 4-4: Escape probabilities

Under these circumstances equation 4-3 is no longer a reasonable approximation. A more rigorous numerical treatment [19] is thus required, relating to these 8 escape probabilities, which need to be evaluated. The geometric escape probability as calculated in ZUT is given by

$$P(E) = W_1 + W_2(W_3 + W_4) + W_2 W_5 \frac{W_6 + W_7}{1 - W_8}. \qquad (4\text{-}4)$$

The resonance integral calculations are performed at different temperatures for each fuel sphere type and stored. The dependence of cross-sections on temperature is achieved by linear interpolation between the respective resonance integral sets.

Neutron spectrum calculations are performed by the codes GAM-I and THERMOS, which are included in the VSOP code. The basic libraries used for both these codes have been extracted from ENDBF-V and JEF-I. Spectrum calculations are done for each spectral zone, consisting of a user-defined number of layers. The resulting neutron fluxes are applied to form broad group cross-sections for the subsequent diffusion calculation. In general, between 2 and 8 broad groups are used for a VSOP calculation. Only one thermal group can be chosen however, since up-scattering is not included in the cross-section transfer.

GAM-I performs epithermal spectrum calculations. It uses the materials homogeneously distributed and applies the P_1-approximation [13]. Optionally, self-shielding factors, as calculated by other codes, can be included to allow for heterogeneous effects. Both resolved and unresolved resonance integrals are passed to GAM from the ZUT code. An adjustment to the infinite homogeneous slowing-down spectrum is made by utilizing feedback from the core calculation. The CITATION module provides leakage terms L_{SI} for the different spectrum zones. For GAM these are converted into buckling terms according to

$$B_{SI}^2 = \frac{L_{SI}}{D_{SI}\Phi_{SI}V_S}, \tag{4-5}$$

which are needed for the P_1 approximation. The GAM library is given in 68 energy groups, ranging from 10 MeV to 0.414 eV, and presently contains 181 materials.

THERMOS performs 1-D cell calculations for thermal neutrons. The effect of the coated particle grain structure of the fuel is included in the code by evaluation of the collision probability for a neutron, which travels through the coated particle, analogously to ZUT. The neutron exchange with the other spectrum zones is accounted for by finding albedos at the surface of the spectrum zone out of leakage terms given by

$$A = \frac{1 - 2\dfrac{L_S}{\Phi_S V_S}\dfrac{V_C}{S_C}}{1 + 2\dfrac{L_S}{\Phi_S V_S}\dfrac{V_C}{S_C}}, \tag{4-6}$$

where the subscripts C and S refers to cell and spectrum zone respectively.

The THERMOS library is a 30-group library in the energy range from 0 to 2.05 eV. This library is actually created from a 96-group library, the THERMALIZATION library. The collapsing is done in VSOP with a special input that takes into account a typical thermal neutron spectrum for the reactor and fuel elements under consideration.

As stated, VSOP is the main code of the VSOP package. It includes the flux solver as well as the spectrum codes, and handles the intercommunication between the different modules. The flux solver used in VSOP, called CITATION, is a 2-D and 3-D finite difference neutron diffusion code in Cartesian and cylindrical geometry.

CITATION uses a mesh cantered finite difference method for its solution of the neutron diffusion equation. The validation effort of this work is mainly focused on this part of the VSOP package. Since the ASTRA facility is described in cylindrical geometry, only the cylindrical geometry option is tested. In addition, only a full 3-D model was set up for the ASTRA facility. This is necessary since there is an asymmetry in the control rod positions.

The reactor lifetime simulation for general cores is an alternation between burnup calculations and fuel shuffling, with one burnup cycle being placed between two shuffling steps. Each burnup cycle can be divided into a number of large burnup steps at which the diffusion calculation can be repeated. Optionally, spectrum calculations and control poison adjustment can also be repeated. A large time step can be subdivided into small time steps. In each small time step, the neutron flux shape is kept unchanged, but its absolute value is adjusted to the requested power production of the core, thus compensating for the depletion of fissile isotopes. Since no fuel shuffling or burnup was required in the ASTRA calculations, this option was not tested.

Other codes are included in VSOP, such as the thermal hydraulics code THERMIX, fuel management and cost analysis codes FEVER and LBD amongst others, but none of these were considered in the validation.

4.4 The SCALE System

The SCALE system [15] was developed in answer to the need for a standardized method of analysis for the evaluation of nuclear fuel facility and package designs. In its present form, the system has the capability to perform criticality, shielding, and heat transfer analysis using well-established functional modules. In this benchmark, only the sequence that provides problem dependent cross sections for stand-alone codes, in the form of what is known as a working library, was used. This working library was then used in XSDRNRSP [24] (see Section 4.5) to create the input for TOTNEW [14] (see Section 4.6) to perform equivalent cross-section calculations.

The parts of the system that were used in creating a problem-dependent working library were BONAMI [20], NITAWL-II [21], XSDRNPM [22] and ICE [23]. BONAMI and NITAWL-II perform resonance self-shielding of cross-sections with Bondarenko factors and resolved resonance data, respectively. XSDRNPM is a general 1-D discrete-ordinates code, in this case used for zone weighting of cross-sections. The utility module ICE is simply used for mixing cross-sections.

4.5 XSDRNRSP

The code XSDRNRSP [24] is a modified version of the one-dimensional discrete ordinates code XSDRNPM-S [25] of the SCALE-3 [26] calculational system. It solves the one-dimensional Boltzmann transport equation in slab, cylindrical or spherical coordinates. The difference to XSDRNPM-S is that is produces an interface file called an RSP file, containing cross-section and flux data for the different zones in the calculation. It is this file that is required as input for the subsequent TOTNEW calculation to determine the equivalent parameters. XSDRNRSP was used to perform 1-D spectral geometry calculations utilizing the ASTRA working library. The specific model is discussed in Section 5.2.

The input library for the spectral calculation is in 238 groups. Even though TOTNEW can handle the group collapsing that is required to reduce the equivalent parameters to the 6 energy groups (1 thermal and 5 fast groups) used in the VSOP calculation, this was rather done in the XSDRNRSP calculation itself. Only the collapsed data was therefore passed to the TOTNEW code.

Multi-group eigenvalue calculations were performed with a tight point-wise flux convergence criterion (10^{-5}) and high order flux expansion (S_{16}) to ensure accurate reference transport results for the equivalent parameter calculation to follow. Anisotropic scattering was taken into account utilizing P_3 approximations.

For each cell in the 1-D spectral geometry model, group cross-sections, cell averaged fluxes, surface currents and volumes are written to the interface file. Note that no spatial homogenization of the cells was performed in the XSDRNRSP calculation, but rather left for the TOTNEW code to handle.

4.6 TOTNEW

TOTNEW [14] was written to implement the method of equivalent cross-sections as discussed in Section 3.7. It uses the results of the S_N code XSDRNRSP to determine the cross-sections and diffusion constants for the control rod geometry under investigation. The TOTNEW program was written in FORTRAN-90 and a listing of the source can be found in Appendix A.

TOTNEW can perform group collapsing of the input data to the required number of energy groups in VSOP, though this feature was not used in the benchmark discussed.

Spatial homogenization of the cells in the spectral calculation is done to obtain homogenized cross-sections for the control rod region.

TOTNEW requires, as part of its input, the dimensions of the control rod in the CITATION geometry, as well as of the neighboring meshes. It compares this to the input of the cells making up the control rod region and of the cells specifying the distance to the neighboring mesh. A warning is given if these dimensions do not correspond.

Diffusion coefficients for the neighboring mesh can be given in the input, or alternatively the diffusion coefficients as calculated from the transport code may be used. The first option is the one recommended and used in this benchmark.

4.7 MCNP

MCNP [6] stands for Monte Carlo n-particle. It is a Monte Carlo based code that can handle neutron, photon and electron transport, including the capability to calculate eigenvalues for critical systems. The code can handle complex 3-D geometry with various material compositions, which makes it extremely versatile. MCNP provides a sufficiently different second method for the verification of the VSOP results. Details of the MCNP model of the ASTRA facility are given in Section 5-5.

5 ASTRA Models

5.1 Introduction

This chapter discusses the various models that were set up to perform calculations for the ASTRA critical facility.

The model used for the control rods in XSDRNRSP and TOTNEW is discussed in Section 5.2. Section 5.3 discusses the method of equivalent boron concentration. Section 5.4 looks at the diffusion coefficients used in the void areas. The actual VSOP model is discussed in detail in Section 5.5, followed by a short discussion on the MCNP model in Section 5.6.

5.2 The ASTRA Control Rod Model in XSDRNRSP and TOTNEW

The control rods in the ASTRA facility are situated in the side reflector, inside the axial channels of the graphite blocks, as shown in Figure 2-1. All these rods have the same configuration, as shown in Figure 2-4. Since a 1-D transport code is used to determine the equivalent parameters, this geometry set-up has to be approximated in one-dimensional cylindrical co-ordinates. The diffusion calculation was set up in such a way as to conserve the diameter of 11.4 cm of the channel in the reflector blocks into which the control rods are placed.

As discussed in Section 2.2 the ASTRA control rods consist of several hollow tubes of steel filled with boron carbide arranged in a circle. This cannot be modeled in a 1-D geometry. In order to perform a sensitivity study of the representation of the control rod in XSDRNRSP and TOTNEW, several models were set up for the CR2 control rod, equivalent parameters determined and used in VSOP calculations to calculate the resultant control rod worth. This worth was then compared to the experimental data. In each of the models the total mass of steel and boron carbide found in the control rods was preserved. The results are shown in Table 5-1.

From Table 5-1 it is clear that the main factor of importance appears to be the surface area of boron exposed. The closer this is to the actual surface area of the boron in the control rods, the closer the worth comparison. Currently the restriction of the control rod hole size to 11.4 cm in width in the VSOP calculation prevents the full surface area to be used in the TOTNEW calculation. However, increasing the absorber size will have other effects that may influence the calculation. Shielding of the tubes from each other may have to be taken into account and the sparse spacing of the tubes may imply that the effective area is somewhat smaller than the total area exposed. This will be investigated in future work. It is also evident that the steel plays an important role in the final result. Placing all of the steel in a ring on the outside drastically reduces the accuracy of the calculation as compared to a steel ring of the appropriate thickness (compare number 3 and number 8 in Table 5-1). This is most likely due to the effect of the steel on the neutron spectrum.

The above calculations are by no means a full investigation into all the factors that can influence the calculation. However, it provides some indication of the representation that can be used. All the models, except number 5 and number 8, fall within an 8% accuracy range and most have less that 2% difference from each other, which falls within the experimental and calculational uncertainty (see Section 6).

Table 5-1: Control rod worth of the CR2 control rod for different 1-D representations

Setup	Short description	Description	Worth	Percentage difference to experiment
1	Conserve average diameter	B_4C in 1.01 cm ring around 7.6 cm diameter with steel of 0.12 cm thickness on outside and the remaining steel on the inside	-2.36	-7.52
2	Conserve outer diameter	Stainless steel of 0.12 cm thickness in a ring of outer diameter of 8.85 cm followed by a ring of boron and then the remaining steel	-2.38	-6.54
3	Conserve ½ B_4C surface area	B_4C in a ring with outer surface area equal to ½ the actual surface area. Stainless steel of 0.12 cm thickness on outside and the remainder on the inside.	-2.39	-5.94
4	Conservation of ½ stainless steel surface area	Steel in a ring with outer surface area equal to ½ total surface area and 0.12 cm thickness. Followed by a layer of B_4C and then the remaining steel.	-2.44	-4.47
5	Conservation of volume (no inner air)	B_4C cylinder of volume equal to the total B_4C volume followed by a layer of steel	-0.14	-94.66
6	Conservation of volume (no outer air)	A ring of B_4C followed by a ring of steel	-2.49	-2.29
7	Conservation of ½ steel surface area (1 steel layer)	A ring of B_4C followed by a ring of steel	-2.37	-6.70
8	Conservation of ½ B_4C surface area (1 steel layer)	A ring of B_4C followed by a ring of steel	2.26	-11.2
9	Maximum B_4C surface area	Ring of steel followed by B_4C and a ring of steel without surrounding air	-2.52	-1.14

Figure 5-1 represents the control rod model in TOTNEW that will be used in all subsequent calculations. It corresponds to setup number 9 in Table 5-1, consisting of an outer ring of steel, 11.4 cm in outer diameter and 0.12cm in thickness followed by a boron carbide ring and the remaining steel in a ring on the inside with void in the center. Two rings of graphite and a fuel driver zone surround this control region. Note that the neighbor size is taken to be 6.9 cm in width and height.

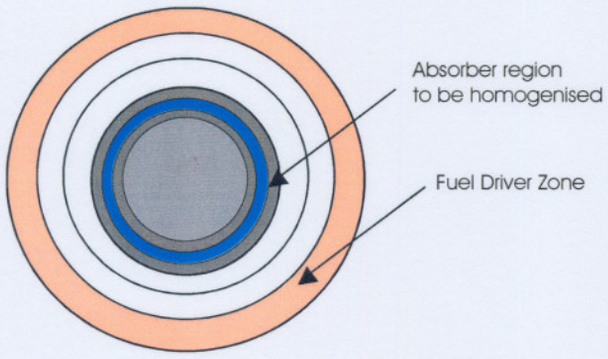

Figure 5-1: Control rod representation in XSDRNRSP / TOTNEW

It must be noted that the above evaluation of the control-rod model used, only applies to the ASTRA facility and can by no means be seen as a valid method to be used, for example, for the control rods in PBMR. The optimal method is very much design related and it is suggested that MCNP calculations be performed to determine the best model to use.

A major problem that exists with this control rod model is the fact that there is an overlap in the meshes between control rods at different positions. This is discussed further in Section 5.5.

5.3 Applying the Method of Equivalent Boron Concentration

When using the method of equivalent boron for the control rods, one would normally insert a boron concentration as to equal the worth of the transport calculation. However, since in the case of the ASTRA facility experimental worths are known for the rods fully inserted, this was used to determine the number density of the boron. Further calculations, such as differential control rod worth, were then performed using this density. Table 5-2 lists the equivalent boron concentrations used for each of the control rods.

Table 5-2: Equivalent boron concentrations for the different control rods

Control rod	Experimental worth ($)	Equivalent boron (^{11}B) concentration (10^{+24} atoms / cm^2)
CR1	-2.46	1.51057E-03
CR2	-2.53	2.48922E-03
CR5	-2.55	2.48922E-03
CR4	-1.95	4.09702E-04

As expected the boron concentration decreases with decreasing control rod worth (CR4 having the smallest concentration of the four). The accuracy of the equivalent boron worth is restricted to the accuracy of the number density of boron that can be inserted (6 significant figures). As a result CR5 and CR2 have the same concentration, which falls within the closest accuracy of the experimental worth.

5.4 Diffusion Coefficients in Void Regions

As discussed in Section 3.9, special methods have to be employed to model void areas in the diffusion approximation. In VSOP, approximate diffusion coefficients, as obtained from the method discussed in Section 3.9.1, are traditionally used. However, none of the void areas in the ASTRA facility comply with the requirement of this method, namely a cavity above the pebble bed core surrounded by graphite. Clearly the ASTRA control rod regions, being situated in the side reflector, cannot be model. Similarly, the void above the core does not form a cavity in the cases when the top reflector is not included in the configuration, as is usually the case. The central experimental channel runs through the center of the core and therefore does not meet the requirement either.

In order to obtain diffusion coefficients for these void areas, the method discussed in Section 3.9.2 was used. Note that only the axial diffusion coefficients were adjusted since they were deemed to have a larger effect on the core. Radial diffusion coefficients will have to be investigated at a later stage.

Table 5-3 shows the MCNP axial diffusion coefficients obtained and compares them to the ones calculated in the traditional manor. It can be seen that the difference is substantial, especially in the case of the control rod region and central experimental channel. Considering that the area above the core, even without the top reflector in place, comes closest to resembling the situation assumed in the traditional method, it was to be expected that the difference between these two diffusion coefficients is less than for the other regions.

Table 5-3: Axial diffusion coefficients comparison

Region	MCNP axial diffusion coefficient / R	Traditional method axial diffusion coefficient / R	% difference
Area above core (top reflector not present)	0.708783	0.532155	-29.122
Control rod region (no control rod inserted)	0.099444	0.660003	-90.056
Central experimental channel	0.064362	0.660520	-98.564

Since the determination of axial diffusion coefficients using analogue Monte Carlo methods does not suffer from the same restrictions as the traditional method, these are the ones that were used in the VSOP model of the ASTRA facility.

5.5 VSOP Model of the ASTRA Critical Facility

The ASTRA facility was modeled in VSOP using full 3-D cylindrical geometry. As a result of the cylindrical geometry, the octagonal core had to be approximated by a volume preserving cylinder 181 cm in diameter. To evaluate the effect of this assumption, two MCNP models were set up, one with an octagonal and one with a cylindrical core. To avoid sphere-cutting effects at the edge of the core, homogenized material was used in both cases. Under these conditions the MCNP showed a relatively small effect of 0.1% on the k-effective value in the case of no control rods inserted. With all control rods inserted the cylinderization effect was much larger at 0.5%. For the experimental layouts considered with a maximum of 3 control rods inserted the effect should be considerably smaller.

An additional problem of the cylindrical model is the position of the control rods. Maintaining exact positions relative to the origin at the core center would lead to incorrect distances between the control rod and the core edge. As expected and confirmed by experiment, the core-to-control rod distance plays an important role in the worth of the control rod (see Section 7.3). The rods were therefore modeled in positions corresponding to their equivalent distance from the core edge, i.e. the distance into the reflector. As discussed in Section 3.7, the method of equivalent cross-sections imposes a restriction on

the mesh structure of any control rod region. Since control rods in ASTRA are positioned at differing distances from the core, an exact model of the rods at the required position is not possible without having overlapping control rod regions (see Figure 5-2)

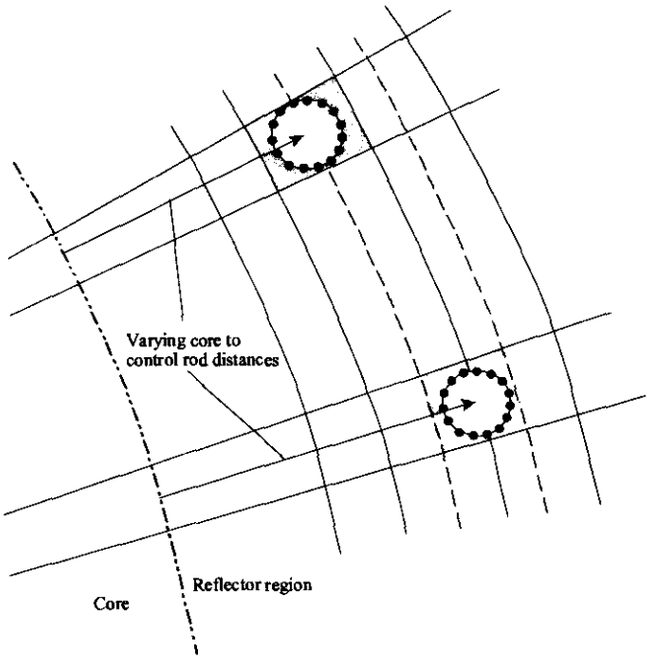

Figure 5-2: Representation of problem of overlapping control-rod regions

The control rods were therefore placed either at the position of the rod under consideration, or at an average position from the core in the case where more than one control rod is considered.

In the axial direction the model consists of 14 layers, one for the bottom reflector, three for the top reflector or void area, and 10 for the side reflector. The core is divided into 89 angles, chosen in such a way as to obtain the absorber region volume for each control rod and a 6.9 cm neighbor mesh on both sides, and then subdividing the regions in between. The model is represented in Figure 5-3.

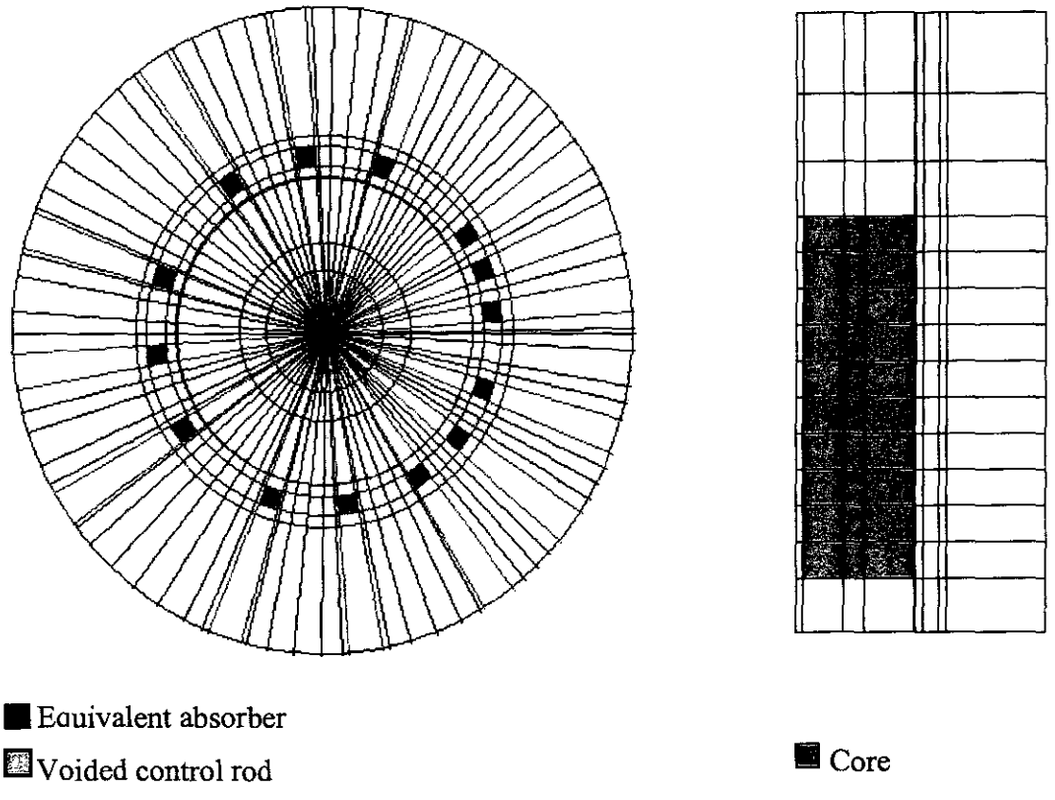

Figure 5-3: Axial (right) and radial (left) representation of the VSOP model

The core is modeled as three channels consisting of one layer of three batches, each chosen in such a way as to equal the correct loading of fuel to absorber to moderator spheres in each channel. Each channel forms its own spectral zone. The central experimental channel, side reflector, bottom reflector and top void area, or top reflector, also form a spectral zone each. Unfilled control rods form one spectral zone with the side reflector with filled control rods a separate one. Each filled control rod is surrounded by a separate spectrum zone to enable the determination of diffusion coefficients in the surrounding area as an input to the method of equivalent cross-sections. In all, nine spectrum zones are used in the calculation.

Calculations were performed in six energy groups, one thermal group and 5 fast groups with the group boundaries shown in Table 5-4.

Table 5-4: Energy group structure for the VSOP model

Group number	Upper energy limit	Lower energy limit
1	10 MeV	1 MeV
2	1 MeV	4 keV
3	4 keV	0.4 keV
4	0.4 keV	29 eV
5	29 eV	1.86 eV
6	1.86 eV	0 eV

The convergence for the CITATION calculation was set to 0.00001 (1 pcm) in k-effective and 0.0001 for the normalized flux.

5.6 MCNP Model of the ASTRA Critical Facility

One of the important features to be taken into account in pebble bed type reactors is the heterogeneity of the fuel, with its internal micro-kernels. In the MCNP model of ASTRA, the micro kernels were modeled explicitly, and placed on a square lattice structure, which was then used to fill a fuel sphere. The B_4C kernels of the absorber spheres used in ASTRA were modeled in a similar fashion.

The spheres used in ASTRA (fuel, absorber and moderator) were then filled into the geometric MCNP model of ASTRA by placing them on a body centered tetragonal lattice (see Figure 5-4) with which the core is filled. Grouping several body-centered tetragonal structures together into a lattice with the correct ratios ensured the correct ratio of fuel, absorber and moderator spheres. It has been shown [27] that this method of modeling the ASTRA facility provides good results.

Figure 5-4: The body-cantered tetragonal structure

A cross-section through the core of the MCNP model is shown in Figure 5-5.

```
  05/19/04 10:11:12
c cyclindrical model of extra -
 homogenized materials

probid =   05/19/04 09:52:03
basis:
( 1.000000,  0.000000,  0.000000)
( 0.000000,  1.000000,  0.000000)
origin:
(     0.00,      0.00,    100.00)
extent = (   200.00,    200.00)
```

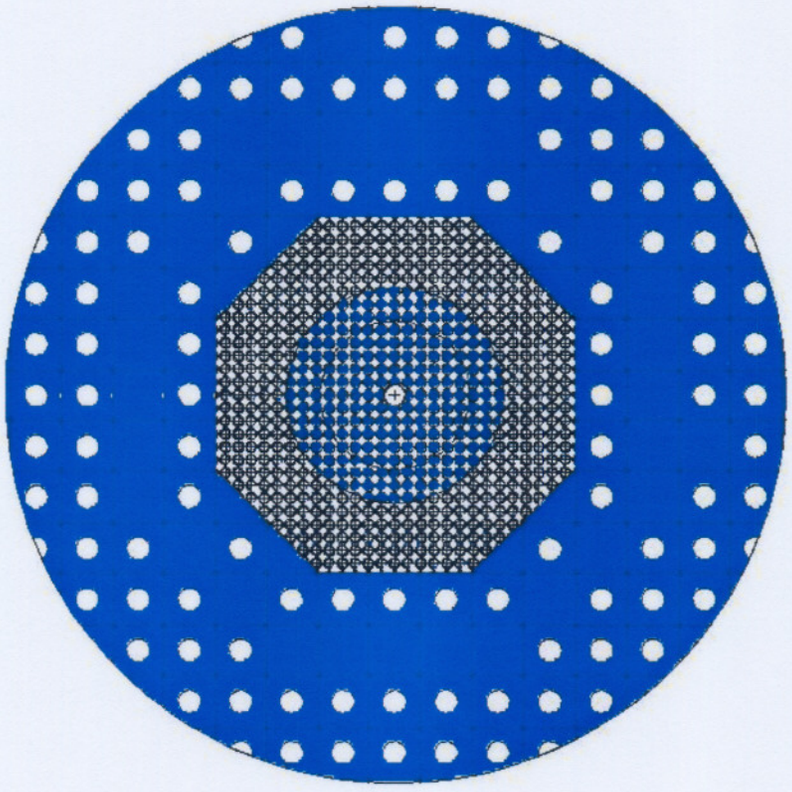

Figure 5-5: Cross-section of the MCNP model at a height of 100 cm

6 Evaluation of the Experimental and Calculational Uncertainties

6.1 Introduction

An evaluation of the uncertainty inherent in the results obtained for both the experimental benchmark and the results of the calculation forms an important part of an evaluation of the validity of a computational code. Should the calculated results fall outside the expected error margin, a more detailed investigation is required as to why this is the case. However, it is very difficult, if not impossible, to assign an error margin to the results of an extensive code like VSOP, or to most experiments.

6.2 Experimental Uncertainty

A thorough experimental uncertainty evaluation was not supplied from the Russian Research Institute where the experiments were performed. Such an evaluation would have provided a clearer framework from which to perform a validation.

From the documentation [28] it is merely stated that the uncertainty in the control rod worth is experimentally determined as ±2.5%. This figure is therefore used in all control rod calculations.

In terms of criticality experiments, the uncertainty is associated with the actual height of the pebble bed. Before any height measurement, the surface of the bed was padded down to level it as best possible. However, the very nature of the stochastic filling of the pebbles implies that any measured value will be but an indication of the average height. Since experiments have shown that a single layer of pebbles can have a reactivity worth of as much as \$0.54 it can be expected that the uncertainty in the height will cause an uncertainty of at least half this on the calculated result, or in other words, a difference in k of about 0.19%.

6.3 Calculational Uncertainty

To determine an accurate measure of the calculational uncertainty is an almost impossible task because of the number of possible influences, from uncertainty in the cross-section libraries to approximations made in the solution method. Several approximations were made in the model of the ASTRA facility, and the inherent approximations of the diffusion theory must also be considered.

In terms of the method of calculation, the convergence in the k was set to 0.00001, setting the lowest limit of the accuracy.

As discussed in Section 5.5 the approximation of the core of the facility as a cylinder instead of an octagon introduces a difference in the k from between 0.1% in the case of no control rods inserted and no control rod positions modeled, to 0.5% for all control rods inserted at core-to-control distance conserved. It can therefore be assumed that the best accuracy that can be obtained must lie somewhere between these two extreme values at about 0.35% in the k value.

The modeled control rod positions, compared to the actual positions is another point that has to be considered because of the problem in modeling overlapping control rods in the method of equivalent cross-sections. Comparing the value in the k-effective between using empty control rod positions at an average distance, closest distance, and furthest distance showed a maximum change of 0.06% in the value of k.

Considering all the uncertainties and assumptions, it can be concluded that the uncertainty in the calculational result should be in the region of 0.60% in the value of k. This falls well within the accepted norms of 1% in k for criticality calculations when compared to experiment.

7 Results and Discussion of VSOP Benchmark

7.1 Introduction

The following describes and discusses the results obtained from VSOP calculations of the ASTRA experiments. Section 7.2 discusses results of calculation for experiments involving the criticality of the facility. Section 7.3 discusses the control rod worth results. This is followed by an investigation of reactivity effects of main components.

The uncertainty for the calculational result is assumed to be approximately 0.6% in the value of k as discussed in Section 6.2. The reactivity worths are given in dollars where the experimentally measured effective delayed neutron fraction β_{eff} of 0.0072 [29] was assumed.

7.2 Initial Criticality and Investigation of Critical Parameters with Varying Height of the Critical Assembly Pebble Bed

In the first experiment performed at the ASTRA facility, the pebble bed height was built up slowly and the height at which criticality was reached noted. In further experiments the height of the pebble bed was built up further and various critical parameters recorded.

7.2.1 Attainment of Criticality

The first experiment performed at the facility was attainment of criticality. In the VSOP calculations, a full 3-D model of the ASTRA facility was set up, and the height set to the height at which criticality was attained. The k calculated was 1.0053±0.6%, a difference of 530 pcm to the experimental result and within the given uncertainty. The full detail MCNP calculation resulted in a k value of 1.00400±0.00175, and thus in very good agreement with the VSOP result.

7.2.2 Variation in Pebble Bed Height

In this experiment the pebble bed height was increased in layers, and the reactivity margin noted by using the CR5 control rod. The results of the calculation as well as the experiment are shown in Table 7-1. In the VSOP model the height of the pebble bed was increased and the mesh structure adjusted where appropriate, to maintain the number of meshes in the axial direction.

Table 7-1: Variation in the assembly reactivity margin [$] with increasing height of the pebble bed

Layer no.	Pebble bed height	Reactivity margin (VSOP)	Reactivity margin (experiment)	Percentage difference	Reactivity margin for layer (VSOP)	Reactivity margin for layer (experiment)	Percentage difference
0	268.9						
1	274.4	0.50	0.64	-21.9	0.50	0.64	-21.9
2	281.4	1.12	1.14	-1.8	0.62	0.50	24.0
3	285.9	1.52	1.58	-3.8	0.40	0.44	-9.1
4	291.8	1.99	2.01	-1.0	0.47	0.43	9.3
5	297.4	2.44	2.43	0.4	0.45	0.42	7.1
6	303.5	2.91	2.85	2.1	0.47	0.42	11.9
7	309.4	3.33	3.26	2.2	0.42	0.41	2.4
8	315.2	3.70	3.66	1.1	0.37	0.40	-8.1
9	320.8	4.08	4.03	1.2	0.38	0.37	0.7

The calculational results compare well with experiment. However, no consistency seems to exist in the percentage error obtained, though variations between experiment and VSOP results are small and well within the uncertainty, so that the possibility exists of experimental uncertainty causing the fluctuations.

7.2.3 Various Critical Configurations

While building up the pebble bed to different heights, various configurations were determined at which the facility reaches criticality. These configurations, as well as the associated VSOP results in k are shown in Table 7-2. Again, VSOP agrees very well with experiment.

Table 7-2: Various critical ASTRA configurations

No	Configuration	Average height of the pebble bed (cm)	Critical position of control rods, (cm)				k_{eff}		
			CR1	CR2	CR5	MR1	Experiment	VSOP Calculation	% difference
1	Without the top reflector and supporting structure	268.9	*↑	↑	↑	326.5±26	1.000	1.0053	0.53
2	Without the top reflector and supporting structure	285.8	↑	↑	355±2	↑	1.000	1.0017	0.17
3	Without the top reflector and supporting structure	320.8	**↓	↑	325±1	↑	1.000	1.0046	0.46
4	With the top reflector and supporting structure	320.8	↑	↓	376±1	↑	1.000	0.99174	-0.826

*↑ = Fully Extracted **↓ = Fully Inserted

The results for all configurations without a top reflector fall well within the 0.6% uncertainty. The configuration with the top reflector shows a slightly larger difference and in addition, it is the only one for which the calculated reactivity was less than one thus showing deviation from the general trend of the VSOP calculations providing a too high k value.

For these calculations, equivalent boron was used for the control rods as determined from the control rod worth at fully inserted, as discussed in Section 5.3. The manual rod is not modeled. Its reactivity worth at fully inserted for the pebble bed at 268.9 cm height is,

according to experiment, only –0.157$. Considering the uncertainty that is associated with the experiment, as well as the assumptions made in the calculations, this is a very small effect.

The use of equivalent boron may imply that the spectrum is different enough in the case with the top reflector present so that the calculated boron concentration no longer applies. Spectrum effects on the boron concentration are discussed further in Section 7.3.3.

It must be noted that in order to model the control rods at the correct positions, the mesh structure as used in VSOP had to be changed for each case through changing the size of one or more of the axial meshes. This, together with experimental uncertainty, may explain the spread of 0.4% observed in the results for the cases without the top reflector.

7.3 Control Rod Worths and Validation of the Method of Equivalent Cross-Sections

The control rod experiments performed at the ASTRA facility, and used for comparison, can be classified into three groups namely (1) the reactivity worth of a single control rod fully inserted, (2) the interference effects of two or more control rods inserted simultaneously, and (3) the differential reactivity worth of a control rod. In the VSOP calculations the control rod reactivity worths were calculated by performing k calculations, with control totally inserted and totally extracted. In the differential worth calculation the insertion depths were restricted to the finite difference axial mesh boundaries.

7.3.1 Method of Equivalent Cross-sections and Single Control Rod Reactivity Worths

In Table 7-3 the reactivity worths of individual control rods are given for both the measured and the VSOP calculation, making use of MECS. Results of calculations for experiments where certain control rods were moved to positions further from the core-edge are also given. In each case the shortest distance from the core edge to the center of the control rod or equivalent rod region is quoted.

Table 7-3: Reactivity worth of individual control rods represented by MECS

CR and block position		Distance from core (cm)	Measured [$]	VSOP [$] (MECS)	Percentage difference
CR5	h4	12.5	-2.53	-2.50	-1.9%
CR2	h12	12.5	-2.55	-2.48	-1.8%
CR1	d8	13.0	-2.46	-2.43	-1.3%
CR4	k5	17.7	-1.95	-2.00	2.8%
CR2	h13	37.5	-0.88	-0.76	-13.5%
CR2	h14	62.5	-0.22	-0.20	-9.7%
CR2	h15	87.5	-0.03	-0.04	-34.2%

The calculated results compare favorably with the measured values for the control rods positioned in the closest reflector positions (results above the double line). For the three rods positioned closest to the core, the reactivity worths are all under-predicted and lie within a small band (0.4%) around –1.7%. For CR4, positioned an additional 5cm away from the core, the predicted worth overestimates the experiment by 2.8%. When the results for increasing distances are considered (results below the double line), the differences are generally larger, and increase significantly to -34.2% as the control is moved to the outermost rod position. At larger distances from the core the error made by diffusion theory is expected to increase but it is evident that the reactivity effect of the control rod also decreases with an associated increase in the relative error. Note that in the last case (h15) the absorber region was positioned so close to the VSOP outer boundary that the mesh size prescribed by MECS could not be satisfied. The discrepancy in the sign seen in the result of CR4 / k5 can possibly be attributed to the presence of the two shutdown rod positions in close proximity to the control rod and the approximation used to account for the streaming effects in these two void areas. This will be investigated further in future.

Although the results with the method of equivalent cross-sections are very promising, it should be noted that for each core-to-control distance, a separate model with radial diffusion meshes, corresponding to the distances of concern, was used. As explained in Section 5.5, this presents a dilemma if multi-rod reactivity worth calculations are to be performed, since the diffusion meshes required to accurately represent the individual control rods will overlap. In most HTR designs and also in the current PBMR design, all the control and shutdown system positions can be accommodated within a single set of radial meshes so that method of equivalent cross-sections is applicable.

7.3.2 Method of Equivalent Cross-sections: Parameters and Sensitivities

When MECS is applied, three sets of parameters are calculated and used in VSOP to represent the equivalent absorber region. The calculated absorption and removal (defined as total out-scatter) cross sections, as well as the equivalent diffusion coefficients are given and compared for all the cases listed in Table 7-4, within the 6-group structure. In Tables 7-5 and 7-6 the values for the four control rod cases in close proximity to the core (CR2, CR1 and CR4) are given. Note that the values (or differences) for CR5 are not included, since they are identical to that of CR2. This is expected since the two control rods (CR5, CR2) are both located 12.5 cm from the core and therefore the same 1-D transport model and CITATION meshes were used in both cases. In all cases the calculated CR-2 control rod values are used as the reference result.

Table 7-4: Absorption cross-sections [cm^{-1}] and differences for CR2, CR1 and CR4

Group	Control rod		
	CR2 (reference)	CR1	CR4
1	6.25E-04	-0.4%	-1.7%
2	4.32E-03	0.0%	1.9%
3	2.13E-02	-0.5%	-2.5%
4	5.06E-02	-0.6%	-3.0%
5	8.72E-02	-0.8%	-3.9%
6	1.05E-01	-0.9%	-4.5%

Table 7-5: Removal cross-sections [cm^{-1}] and differences for CR2, CR1 and CR4

Group	Control rod		
	CR2 (reference)	CR1	CR4
1	6.78E-03	1.6%	8.4%
2	2.22E-03	2.4%	16.5%
3	5.66E-03	1.7%	9.6%
4	6.82E-03	1.8%	10.2%
5	6.75E-03	1.7%	10.7%
6	1.62E-05	-0.2%	-9.9%

Table 7-6: Diffusion coefficients [cm] and differences for CR2, CR1 and CR4

Group	Control rod		
	CR2 (reference)	CR1	CR4
1	1.26E+01	51.0%	386.4%
2	2.75E+00	-0.9%	-23.4%
3	3.60E+00	-1.6%	-7.8%
4	3.11E+00	-1.0%	-2.6%
5	2.45E+00	-0.7%	-1.6%
6	2.47E+00	-0.9%	-1.4%

The variation in the absorption cross sections is quite small with a maximum change of 4.5% in the thermal group (< 1.86 eV). The differences in the removal cross sections are considerably larger, with differences up to 16.5% seen for CR4. Since the absorption and removal cross sections are only a function of the 1-D transport model, and since the same absorber volume was used in all calculations, the differences can only be attributed to the fine group spectra variations of the representative 1-D transport models. This explains why the largest effects are noticed for CR4 positioned 17.7 cm from the core, compared to the 12.5 cm for the selected reference CR2. This is confirmed below, when the variation in the MECS parameters for CR2 for various distances are compared.

The equivalent diffusion parameters presented in Table 7-6 show a much larger variation in values with the group one value for CR4 nearly a factor four larger than the selected reference CR2. The diffusion coefficient is, of course, an adjusted or equivalent value dependent not only on the flux spectra, but also on the leakage, mesh sizes and diffusion coefficients of the neighboring regions.

In Table 7-8 and Table 7-9, the values for CR2 at four control-rod-to-core distances are given. Similar to the previous discussion, the values for CR2 in position h12 (12.5 cm) were selected as reference, and only differences are given for the other three cases, h13 at 37.5 cm, h14 at 62.5 cm and h15 at 87.5 cm. For these cases the differences in the cross sections are considerably larger. The large variation in the control-rod-to-core distances and the corresponding larger variation in the spectrum is the reason for this. In most cases the differences, compared to the reference position h12, become systematically larger as the distance increases and as the spectrum softened (see Table 7-10). It must be kept in mind that the fine group spectrum changes in the absorber region of the 1-D transport calculation cause the six group cross section variations and not the six-group spectrum variations presented here for comparison.

Table 7-7: Variation in absorption cross-section [cm-1] from MECS with increasing distance from the core

Group	Control rod			
	h12 (reference)	h13	h14	h15
1	6.25E-04	3.4%	7.4%	10.0%
2	4.32E-03	17.7%	29.4%	36.2%
3	2.13E-02	1.4%	2.7%	2.8%
4	5.06E-02	1.4%	2.6%	2.6%
5	8.72E-02	1.1%	2.0%	1.5%
6	1.05E-01	0.7%	1.1%	-0.2%

Table 7-8: Variation in removal cross-section [cm-1] from MECS with increasing distance from the core

Group	Control rod			
	h12 (reference)	h13	h14	h15
1	6.78E-03	-4.2%	-13.9%	-21.0%
2	2.22E-03	32.5%	56.4%	77.9%
3	5.66E-03	6.4%	11.6%	18.5%
4	6.82E-03	8.3%	15.0%	23.4%
5	6.75E-03	12.4%	22.0%	32.9%
6	1.62E-05	-62.9%	-87.9%	-96.0%

Table 7-9: Variation in diffusion coefficients [cm] from MECS with increasing distance from the core

Group	Control rod			
	h12 (reference)	h13	h14	h15
1	1.26E+01	-82.4%	-77.1%	155 %
2	2.75E+00	-63.7%	-66.2%	-63.1%
3	3.60E+00	-22.8%	-26.5%	-50.6%
4	3.11E+00	14.7%	38.0%	-30.0%
5	2.45E+00	18.3%	49.6%	-18.7%
6	2.47E+00	37.5%	143 %	-8.8%

Table 7-10: Neutron energy spectrum (lethargy [dimensionless]) in absorber region from the 1-D transport results

Group	Control Rod (CR2) Position			
	h12	h13	h14	h15
1	0.122	0.043	0.012	0.003
2	0.328	0.169	0.054	0.013
3	0.252	0.211	0.094	0.028
4	0.131	0.134	0.070	0.024
5	0.088	0.115	0.073	0.029
6	0.080	0.327	0.698	0.905

The equivalent diffusion coefficients show large variations for the various cases, with differences up to 155 %. Compared to the results presented in Table 7-9, the larger differences are not only restricted to the fast groups, with large differences also seen in the thermal group. Since these differences could be due to various factors, no further attempt was made to analyze the specific reasons for these effects. A sensitivity study was performed to determine how sensitive the reactivity worth of a control rod is on the selected method of equivalent cross-sections parameters. To test this, the method of equivalent cross-sections parameters of cases h12 to h15 were substituted in VSOP calculations for cases h12 to h14. These results are given in Table 7-11 and show the differences compared to the reference result obtained when the appropriate method of equivalent cross-section set was used (h12 method of equivalent cross-section parameters in case h12 and so on).

Table 7-11: Sensitivity of reactivity worth to method of equivalent cross-section parameters

CR position	Method of Equivalent Cross-section set used			
	h12	h13	h14	h15
h12	-	1.1%	4.8%	-1.5%
h13	-8.0%	-	5.0%	-5.9%
h14	-7.6%	-9.0%	-	-7.6%

The effects on the reactivity worth are surprisingly small, especially since such large differences in the MECS parameters were found. This provides some comfort since the MECS diffusion coefficients in particular were found to be sensitive to changes in the 1-D transport model and mesh size selection.

7.3.3 Usage of the Equivalent Boron Concentration Method

Since the manual input option is used to introduce the equivalent cross sections in VSOP, the stepping out of the control rods can only be performed with the help of restart calculations (one required for each control rod step). This adds a practical difficulty in using the method of equivalent cross-sections for incremental control rod VSOP calculations. Extraction to a distance not on a mesh interval is also not possible with the MECS. Therefore, an equivalent Boron concentration (EBC) often replaces the equivalent cross sections. In this case the rod movement can be defined in a single VSOP run and no restarts are required.

A single equivalent Boron concentration was determined to conserve the experimentally obtained worth of CR2 in position h12. The control rod was then moved away from the core to positions h13, h14 and h15. In the VSOP model the same equivalent boron concentration was used to represent the rod when moved to the other positions further from the core edge, as indicated in Table 7-12. The calculated VSOP results are compared to measurements.

Table 7-12: Reactivity worth of CR2 at different positions represented by a single EBC

Block position (see Figure 1)	Distance from core (cm)	Measured [$]	VSOP [$] Boron equivalent	Percentage difference
h13	37.5	-0.88	-0.77	-12.1 %
h14	62.5	-0.22	-0.19	-14.1 %
h15	87.5	-0.03	-0.008	-72.5 %

It is immediately apparent that the reference calculated boron concentration of position h12 cannot be applied to other control regions. A similar effect was seen when control rod CR4 was moved from position k5 to l4. A possible reason is that the diffusion error increases for increased distances from the core, or simply that the relative error becomes larger as the reactivity worth value decreases. Also, since the calculated boron concentration was determined to reproduce the reference values, any deficiencies in the diffusion calculation, such as transport or self-shielding effects, are also taken into account when determining the reference boron concentration. In other words, the EBC is sensitive to the neutron spectrum in which it is determined. These effects will be different for each control position, especially when positioned further away from the core. The use of a single set of EBC for control rods of the same design and at equal distances from the core should however be acceptable.

7.3.4 Interference of Control Rods

In another set of calculations, the combined reactivity worth of several combinations of rods inserted in ASTRA was compared with the measured result. In this case the EBC for each control rod was determined, and applied in the VSOP calculations. The differences shown in Table 7-13 are therefore due to the combined control rod effects, also called the interference effects. The results obtained are satisfactory and it can be concluded that the EBC method can be used to represent the reactivity effects of control rods, provided that an EBC is calculated for each "type" of control rod (in the sense of design or placement in the core).

Table 7-13: Reactivity worth of combination of control rods each represented by its equivalent boron concentration

Control rod combination	Measured [$]	VSOP [$] Boron equivalent	Percentage difference
CR1 + CR5	-5.16	-5.12	-0.9%
CR2 + CR5	-5.57	-5.53	-0.7%
CR4 + CR5	-4.31	-4.34	0.8%
CR1 + CR2 + CR5	-8.42	-8.43	0.1%
CR1 + CR4 + CR5	-7.15	-7.31	2.2%
CR2 + CR4 + CR5	-7.57	-7.76	2.5%

7.3.5 Differential Control Rod Worth

As a final example, the differential worth of the CR5 control rod (s-curve) was determined. In the experiments, the worth of the control rod was measured at various heights of extraction. In the VSOP calculations the EBC for the control rod at full insertion was determined and then used for the subsequent differential worth calculations. As a validation, MCNP calculations were also performed. The results obtained, as shown in Figure 7-1, are satisfactory, and it can be concluded that EBC can be used for differential control rod worth calculations.

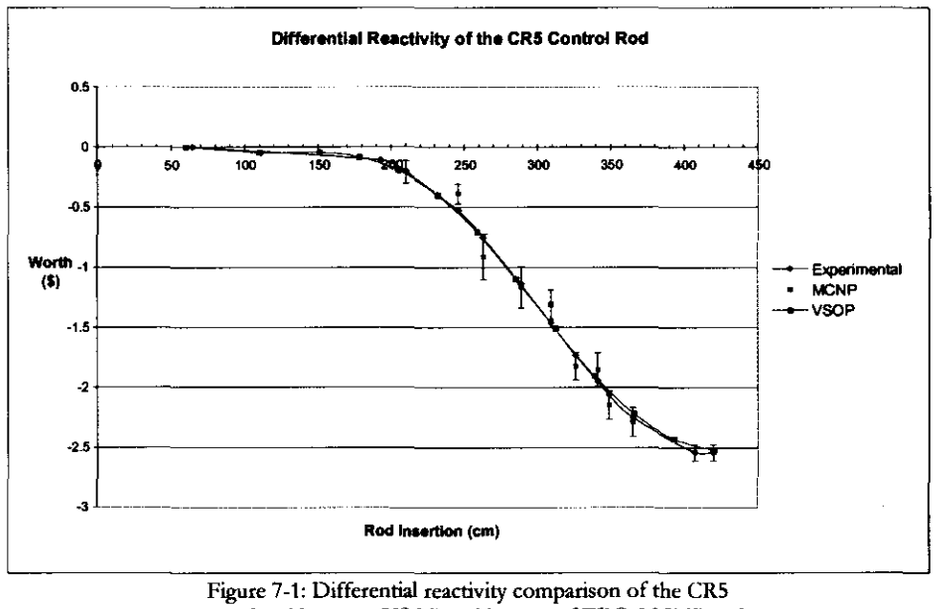

Figure 7-1: Differential reactivity comparison of the CR5 control rod between VSOP making use of EBC, MCNP and experiment

7.4 Reactivity Effects of Main Components and Materials

Reactivity effects of single spheres of all types, as well as of groups of spheres, were measured at the ASTRA facility in the central experimental channel of the core. Only the combinations of spheres were calculated using VSOP. The results are shown in Table 7-14.

VSOP does not allow for sphere data (from DATA-2) to be inserted in a region that is not specified as core. As a result, the materials were inserted into the region by setting the appropriate number densities of the relevant isotopes. The grain structure is, however, taken into account in the spectrum calculation where the fuel sphere cell calculations is used.

The experiment used a rod, which contained graphite plugs. In the VSOP model the entire channel was filled with graphite, except for the region where the spheres were placed. As a result, it is not possible to compare experimental worth of the experimental rod in its entirety with the VSOP result, but only the relative worths of the sphere combinations. In addition, there is more graphite in the core than there was in the experiment, which can be a possible explanation for the large percentage difference in the calculated worth of the moderator spheres, compared to the experimental value.

For the different sphere combinations care was taken to adjust the THERMOS calculations accordingly. This seemed to have an especially large effect in case of the fuel spheres and the absorber spheres. The moderator spheres were not as strongly affected.

Table 7-14: Reactivity effects for groups of spheres

Group Composition	Reactivity Effect ($)	Experimental Reactivity Effect ($)	Percentage Difference
20 MS*	-0.004	-0.007±0.003	-42.86%
19 MS + 1 AS**	-0.052	-0.044±0.003	18.18%
20 AS	-0.440	-0.602±0.003	-26.91%
20 FS***	0.138	0.141±0.003	-2.13%
18 FS + 2 MS	0.110	0.126±0.003	-12.70%
19 FS + 1 MS	0.117	0.141±0.003	-17.02%

* MS – Moderator Spheres

** AS – Absorber Spheres

*** FS – Fuel Spheres

8 Conclusions and Recommendations

8.1 Conclusions

The results of these calculations have shown that VSOP is indeed able to model the ASTRA facility.

In terms of critical parameters, in Section 7.2 it is shown that for the initial criticality of the facility the calculated k differed from the experimental value of one by 530 pcm, within the expected 0.6% uncertainty level. It is also shown that VSOP provides accurate results on the reactivity effects of adding layers of spheres, with all reactivity margins falling within 22% of the experimental values. However, there appears to be no consistency in the difference between experiment and calculation, leading to the conclusion that this is caused by experimental uncertainty, especially in the cases where the reactivity margin is relatively small. Various critical configurations are also shown to produce VSOP results in the k, which are within the 0.60% uncertainty margin for all configurations that do not include the top reflector. Once this reflector is included, the k effective value differs from the experiment by 0.826%. The reason for this is the influence of the top reflector on the neutron spectrum, thus influencing the EBC of the control rods modeled.

For the control rod calculations as discussed in Section 7.3, it is shown that the MECS can calculate the control rod worth to within 1.9% of the experimental value for control rods close to the core. This falls within the experimental uncertainty of 2.5%. However, as the distance between the control rod and the core is increased, the difference between experiment and calculation increases to as much as 34.2%. This can be attributed to a number of effects, most notably the increase in the error made in diffusion theory with increasing distance from the core. It must also be noted that the restriction of the mesh structure in MECS restricts the control rod placement in the model due to the overlap of

control rod regions. This can have a difference of as large as 0.5% in the k. In the usage of EBC it is shown that the method can yield very good results when considering such effects as interference of control rods (calculational results were within 2.5% of experiment) and differential control rod worth. However, EBC becomes ineffective in cases when there is a large spectrum change as compared to the spectrum in which the EBC was calculated. Most notably, EBC cannot be used when the control rod is moved further away from the core. In this case, the difference between the calculated and experimental worth increased to as much as 73% for the most distant position.

Reactivity effects of different sphere types introduced into the central experimental channel are shown in Section 7.4 to be within 27% of experiment. Because of the restrictions of VSOP, several assumptions had to be made here, such as the inclusion of more graphite, which contribute to the larger calculational difference.

Overall, VSOP has been shown to provide acceptable answers to the experiments, provided care is taken in several aspects, especially concerning the control rod model with EBC. It has been shown that the method of equivalent cross sections can be applied highly successfully for this type of configuration. Control rod worths calculated were well within the uncertainty margin of 2.5% in worth, when compared to experiment. The inability to have overlapping control rod regions does however limit the accuracy that can be modeled with VSOP for the ASTRA facility. The method of equivalent boron has been shown to be highly effective and useful for calculations provided that care is taken in their use. It must be assured that the spectrum at which the boron concentration was calculated does not differ substantially from that in which the concentration is to be used.

8.2 Recommendations for Future Work

The MECS has been shown to yield favorable results for the ASTRA facility. Since it was found that the limitation of a single mesh in the rodded regions can cause modeling problems due to overlaps, it will be of an advantage to investigate changes to the method to overcome this deficiency. Future improvement may also include the extension of the transport code to two dimensions, and the associated introduction of asymmetry in the method. This, coupled with directionally dependent diffusion coefficients in CITATION in the radial and azimuthal direction, will provide greater accuracy in the calculation.

As more experiments are being performed at the ASTRA facility, it is recommended that work proceed in comparing experimental results to VSOP calculations as a highly efficient means to validate the code and ultimately provide validation for PBMR calculations.

9 References

[1] **E.J. Mulder**, "*Core Basic Design Report for the PBMR Nuclear Power Plant*", PBMR Document Number 002979-34 (2000)

[2] **E. Teuchert, et al.**, "*VSOP (94), Computer Code System for Reactor Physics and Fuel Cycle Simulation Input Manual and Comments*", Jül-2897 (1994)

[3] "*Services to be Performed at the ASTRA Critical Facility for the PBMR Plant*", Work under contract between ESKOM and A/O TECHSNABEXPORT

[4] **W. Scherer, et al.**, "*Reaktorphysikalishe Experimente am AVR-Kraftwerk und deren Analyse*", Lecture and der Jahrestagung Kerntechnik, Frankfurt, FRG (1984)

[5] **H.J. Neef, D Wahl**, Reactortagung 1973, Karlsruhe

[6] **J.F. Briesmeister,** Editor, "*MCNP – A General Monte Carlo N-Particle Transport Code*", Los Alamos National Laboratory, Report LA-12625-M, Version 4C

[7] **W. Scherer, H.J. Neef**, "*Determination of Equivalent Cross Sections for Representation of Control Rod Regions in Diffusion Calculations*", JÜL-1311 (1976)

[8] **T.B. Fowler, D.R. Vondy, G.W. Cunningham**, "*Nuclear Reactor Analysis Code: CITATION*", ORNL-TM-2496, Rev. 2 (1971)

[9] **L.L. Briggs, E.E. Lewis**, "*A comparison of Constrained Finite Elements and Response Matrices as One Dimensional Transport Approximation*", Nucl. Sci. and Eng.: 63, 225-235 (1977)

[10] **Z. Weiss**, "*An Improved Response Matrix Method for Calculating Neutron Flux Distributions*", Nucl. Sci. and Eng.: 63, 457-492 (1977)

[11] **T. Takeda, K.Azecura, T.Ohnishi**, "*An improved Response Matrix Method for Calculating Neutron Flux Distributions*", Nucl. Sci. and Eng.: 62, 709-715 (1977)

[12] **V. Fen, M. Lebedev, V. Sarytchev, W. Scherer,** "*Modelling of Neutron Absorbers in High Temperature Reactors by Combined Transport-Diffusion Methods*", JÜL-2573 (1992)

[13] **E.E. Lewis, W.F. Miller, Jr**, "*Computational Methods of Neutron Transport*", John Wiley & Sons,

[14] **D. Naidoo,** "*TOTNEW User Manual*"

[15] "*SCALE-4.4: Modular Code System for Performing Criticality Safety Analyses for Licensing Evaluation*", RSICC CCC-545 (2000)

[16] **J.J. Duderstadt, L.J. Hamilton**, "Nuclear Reactor Analysis", John Wiley & Sons (1976)

[17] **M.S. Milgram**, "*Estimation of Axial Diffusion Process by Analog Monte-Carlo: Theory, Tests and Examples*", Ann.Nucl.Energy,Vol. 24,No. 9,pp. 671-704 (1997)

[18] **L.W. Nordheim**, "*The Theory of Resonance Absorption*", Proceedings of Symposia in Applied Mathematics, Vol. XI, p. 58, G. Birkhoff and E. P. Wigner, Eds., Am. Math. Soc. (1961).

[19] **U. Hansen, E. Teuchert**, "*Influence of Coated-Particle Structure in Thermal Neutron Spectrum Energy Range*", Nucl. Sci. Eng. 44,12 (1971)

[20] **N.M Greene, "***BONAMI: Resonance Self Shielding by the Bondareko Method***"**, NUREG/CR-0200, Revision 6, Volume 2, Section F1, ORNL/NUREG/CSD-2/V2/R6

[21] **N.M. Greene, L.M. Petrie, R. M. Westfall**, "*NITAWL-II: SCALE System Module for Performing Resonance Shielding and Working Library Production*", NUREG/CR-0200, Revision 6, Volume 2, Section F2, ORNL/NUREG/CSD-2/V2/R6

[22] **N.M. Greene, L.M. Petrie**, "*XSDRNPM: A one-Dimensional Discrete-Ordinates Code for Transport Analysis*", NUREG/CR-0200, Revision 6, Volume 2, Section F3, ORNL/NUREG/CSD-2/V2/R6

[23] **N.M. Greene, L.M. Petrie, S.K. Fraley**, "*ICE: Module to Mix Multigroup Cross Sections*", NUREG/CR-0200, Revision 6, Volume 2, Section F8, ORNL/NUREG/CSD-2/V2/R6

[24] **E.Z. Müller**, "*XSDRNRSP: A One-Dimensional Ordinates Code with Improved Cross Section Weighting Facilities Utilizing Transport Corrections*", PEL 317 (1992)

[25] **N.M. Greene, L.M. Petrie**, "*XSDRNPM-S: A One-Dimensional Discrete-Ordinates Code for Transport Analysis*", NUREG/CR-0200, Volume 2, Section F3 (1983)

[26] "*SCALE-3: Modular Code System for Performing Criticality Safety Analyses for Licensing Evaluation*", RSICC CCC-0466

[27] **Z. Karriem, C.C. Stoker, F. Reitsma**, "*MCNP Modelling of HTGR Pebble-Type Fuel*", MC2000 Conference (2000)

[28] **V.P. Garin et al**, "*Stage A-1.2.3. Investigation of Critical Parameters with Varying Height of the Pebble Bed, Worth of Control Rods depending on their Position in the Side Reflector, and Differential Reactivity of the Control Rod Depending on the Depth of its Insertion, with Calculational Analysis of the Results Obtained*", RRC KI, Moscow (2000)

[29] "*Task A1-2: Investigation of the Annular Core Neutronics at the ASTRA Facility at Room Temperature*", RRC KI, Moscow (2002)

[30] **H. Gerwin, W. Scherer**, "*Treatment of Upper Cavity in a Pebble-Bed High-Temperature Gas-Cooled Reactor by Diffusion Theory*", Nuclear Science and Engineering, 97, 9-19 (1987)

[31] **F. Reitsma, D. Naidoo, Z. Karriem**, "*An Evaluation of the Control Rod Modelling Approach used in VSOP by Comparing its Results to Experiments Performed at the ASTRA Critical Facility*", HTR-2002 (2002)

[32] **D. Naidoo, J. Jevremovich**, "*Validation of VSOP'94 using Experiments Performed at the ASTRA Facility*", ANS Summer Meeting, Hollywood, Florida (2002)

10 Appendices

APPENDIX A: TOTNEW SOURCE

TOTNEW.F90

```
!
! Main for TOTNEW program
!

PROGRAM totnew

! Modules Used
USE versio_mod  ! Version Information
USE error_mod   ! Error handling routines
USE readinp_mod ! Routines for reading the totmos input file
USE readrsp_mod ! Routines for readin the rsp file
USE mixdata_mod ! Routines for mix type
USE collaps_mod ! Routines for collapsing xsection
USE xsec_mod    ! Routines for determining xsecs in rod region
USE diff_mod    ! Routines for calculating diffusion coefficients

IMPLICIT NONE

WRITE(*,*)
WRITE(*,100) VERSIO
WRITE(*,*)

CALL CPU_TIME(cpu_start)

!
! Read input
!
CALL readinfile

WRITE(OUTFILE,*)
WRITE(OUTFILE,100) VERSIO
WRITE(OUTFILE,*)

WRITE(OUTFILE,*)
```

```
WRITE(OUTFILE,*) '0000000000000000000000000000000000000000000000000000000000000000000000000000'
WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'FILE DATA:'
WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'Input Filename : ', trim(inputfile)
WRITE(OUTFILE,*) 'Output Filename: ', trim(outputfile)
WRITE(OUTFILE,*) 'RSP Filename   : ', trim(rspfile)
WRITE(OUTFILE,*)
WRITE(OUTFILE,*) '0000000000000000000000000000000000000000000000000000000000000000000000000000'
WRITE(OUTFILE,*)

WRITE(*,*)
WRITE(*,*) '0000000000000000000000000000000000000000000000000000000000000000000000000000'
WRITE(*,*)
WRITE(*,*) 'FILE DATA:'
WRITE(*,*)
WRITE(*,*) 'Input Filename : ', trim(inputfile)
WRITE(*,*) 'Output Filename: ', trim(outputfile)
WRITE(*,*) 'RSP Filename   : ', trim(rspfile)
WRITE(*,*)
WRITE(*,*) '0000000000000000000000000000000000000000000000000000000000000000000000000000'
WRITE(*,*)

!
! Read rsp file - already done in readinfile
!
! CALL readrspfile

WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'MIXTURE DATA ON RSP FILE'
WRITE(OUTFILE,*)

CALL show_all_mixtures( 1 )

!
! Collaps the Cross Sections to broad groups
!
CALL collaps(ngb,ngbfng)

IF (ngb > 0) THEN

WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'COLLAPSED MIXTURE DATA'
WRITE(OUTFILE,*)

!
! Show the basic data
!
 CALL show_all_mixtures( 1 )

END IF
```

```
!
! Reduce to zone data
!
CALL create_zone_data

!
! Show Zone data
!

WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'ZONE DATA'
WRITE(OUTFILE,*)

CALL show_all_zones( 1 )

!
! Calculate siga and sigr for control rod region
!
CALL rod_xsec( cnumz, czones )

!
! Calculate diffusion constant
!
CALL rod_d

!
! Show results
!
CALL show_xsecs

!
! Deallocate Everything before program termination
!
CALL deallocate_mixtures
CALL deallocate_zones
CALL deallocate_input

!
! Write completion comment
!

CALL CPU_TIME(cpu_end)

WRITE(OUTFILE,*)
WRITE(OUTFILE,101) (cpu_end - cpu_start)
WRITE(OUTFILE,*)

WRITE(*,*)
WRITE(*,101) (cpu_end - cpu_start)
WRITE(*,*)
```

```
CLOSE(OUTFILE)

! FORMATS
 100  FORMAT('TOTNEW (VERSION=',F4.2,')')
 101  FORMAT('TOTNEW COMPLETED SUCESSFULLY ( CPU Time = ', F6.2, ' s )')

END PROGRAM totnew
```

VERSIO_MOD.F90

```
MODULE versio_mod

IMPLICIT NONE

 REAL, PARAMETER :: VERSIO = 1.01D0

END MODULE versio_mod
```

ERROR_MOD.F90

```
MODULE error_mod
! module for error handling inside totnew

IMPLICIT NONE

CHARACTER(LEN=256), PRIVATE :: errmsg
REAL                :: cpu_start, cpu_end
INTEGER,PARAMETER       :: OUTFILE=84       ! Output File Unit

CONTAINS

 SUBROUTINE errset( msg )

   IMPLICIT NONE
   CHARACTER(*), INTENT(IN) :: msg

   CALL enter_sr( 'errset' )

   errmsg = msg

   CALL leave_sr( 'errset' )
```

```
END SUBROUTINE errset

SUBROUTINE warning( msg )

 IMPLICIT NONE
 CHARACTER(*), INTENT(IN), OPTIONAL :: msg
 LOGICAL :: isopen

 CALL enter_sr( 'warning' )

 IF (PRESENT(msg)) THEN
   errmsg = msg
 END IF

 errmsg = TRIM(errmsg)

 INQUIRE(UNIT=OUTFILE,OPENED=isopen)
 IF (isopen) THEN
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'WARNING: ', errmsg
 ELSE
  WRITE(*,*)
  WRITE(*,*) 'WARNING: ', errmsg
 ENDIF

 CALL leave_sr( 'warning' )

END SUBROUTINE warning

SUBROUTINE error( msg )

 IMPLICIT NONE
 CHARACTER(*), INTENT(IN), OPTIONAL :: msg
 LOGICAL :: isopen

 CALL enter_sr( 'error' )

 IF (PRESENT(msg)) THEN
   errmsg = msg
 END IF

 errmsg = TRIM(errmsg)

 INQUIRE(UNIT=OUTFILE,OPENED=isopen)

 IF (isopen) THEN
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'ERROR: ',errmsg
  WRITE(OUTFILE,*) 'Aborting TOTNEW'
 ELSE
```

```
  WRITE(*,*)
  WRITE(*,*) 'ERROR: ',errmsg
  WRITE(*,*) 'Aborting TOTNEW'
 ENDIF

 CALL leave_sr( 'error' )

 STOP

END SUBROUTINE error

SUBROUTINE enter_sr( name )

 CHARACTER(LEN=*) :: NAME
 LOGICAL :: isopen

 RETURN

 CALL CPU_TIME(cpu_end)

 INQUIRE(UNIT=OUTFILE,OPENED=isopen)

 IF (isopen) THEN
  WRITE(OUTFILE,'(A21,A,A,F10.2,A)') 'ENTERING SUBROUTINE: ', name, &
                  ' (CPU Time = ', &
                  (cpu_end - cpu_start), ' s )'
 ELSE
  WRITE(*,'(A21,A,A,F10.2,A)') 'ENTERING SUBROUTINE: ', name, &
                  ' (CPU Time = ', &
                  (cpu_end - cpu_start), ' s )'
 ENDIF

END SUBROUTINE enter_sr

SUBROUTINE leave_sr( name )

 CHARACTER(LEN=*) :: NAME
 LOGICAL :: isopen

 RETURN

 CALL CPU_TIME(cpu_end)

 INQUIRE(UNIT=OUTFILE,OPENED=isopen)

 IF (isopen) THEN
  WRITE(OUTFILE,'(A21,A,A,F10.2,A)') 'LEAVING SUBROUTINE: ', name, &
                   ' (CPU Time = ', &
                   (cpu_end - cpu_start), ' s )'
 ELSE
  WRITE(*,'(A21,A,A,F10.2,A)') 'LEAVING SUBROUTINE: ', name, &
```

```
                    ' (CPU Time = ', &
                    (cpu_end - cpu_start), ' s )'
  ENDIF

 END SUBROUTINE leave_sr

END MODULE error_mod
```

READINP_MOD.F90

```
MODULE readinp_mod
! Module to read the totmos input file

USE error_mod
USE readrsp_mod
USE mixdata_mod

IMPLICIT NONE

CHARACTER(LEN=256)            :: inputfile='in.dat' ! Input Filename
CHARACTER(LEN=256)            :: outputfile       ! Output filename
INTEGER,PARAMETER             :: INFILE=80        ! Input File Unit
INTEGER                       :: cnumz            ! Number of zones in CR
INTEGER,DIMENSION(:),ALLOCATABLE :: czones        ! Array of zone numbers
INTEGER                       :: ngb              ! Number of broad groups
INTEGER,DIMENSION(:),ALLOCATABLE :: ngbfng        ! Broad group structure
INTEGER                       :: rx,ry            ! Zone numbers for rx,ry
REAL                          :: a,pa,pxa,ppya    ! Geometruc Dimensions
LOGICAL                       :: ownd=.FALSE.     ! D's give in input
REAL,DIMENSION(:),ALLOCATABLE :: ds               ! input D's
LOGICAL                       :: totx=.FALSE.     ! rx  as in totmos?
LOGICAL                       :: toty=.FALSE.     ! ry as in totmos?

CONTAINS

!
!
!
 SUBROUTINE deallocate_input
!
! Deallocates any memory created in the input reading
!

  CALL enter_sr( 'deallocate_input' )

  IF (ALLOCATED(czones)) THEN
```

```
   DEALLOCATE(czones)
  END IF

  IF (ALLOCATED(ngbfng)) THEN
   DEALLOCATE(ngbfng)
  END IF

  CALL leave_sr( 'deallocate_input' )

 END SUBROUTINE deallocate_input

!
!
!
 SUBROUTINE getfilename( comment, filename, argnum )
!
! Requests and reads the input filename if not give on the command line
!
  CHARACTER(LEN=*)   :: comment
  CHARACTER(LEN=256) :: filename
  INTEGER :: argnum
  INTEGER :: numargs=0
  INTEGER :: IARGC


!
! Get command line
!

  numargs = IARGC()

  IF (numargs >= argnum) THEN

   CALL GETARG(argnum,filename)

  ELSE

   CALL enter_sr( 'getfilename' )

   WRITE(*,*) comment

   READ(*,*) filename

  END IF

  CALL leave_sr( 'getfilename' )

 END SUBROUTINE getfilename

!
!
```

```
!
 SUBROUTINE readinfile
!
! Reads the input file
!

  INTEGER :: i,nond,j,k
  REAL    :: nd
  CHARACTER(LEN=256) :: str

  CALL enter_sr( 'readinfile' )
!
! Get the input filename
!
  CALL getfilename( 'Please enter input filename: ', inputfile, 1)
!
! Open File for reading
!
  OPEN(INFILE, FILE=inputfile,STATUS='OLD',ACTION='READ',ERR=999)
!
! Read output filename
!
  READ(INFILE,*,ERR=998) outputfile
!
! Open outputfile for writing
!
  OPEN(OUTFILE,FILE=outputfile,STATUS='NEW',ACTION='WRITE', &
     ACCESS='SEQUENTIAL',FORM='FORMATTED',ERR=997)
!
! Read rsp filename
!
  READ(INFILE,*,ERR=998) rspfile
!
! Read rsp file
!
  CALL readrspfile
!
! Read number of broad groups
!
  WRITE(OUTFILE,*) 'Reading number of energy groups ...'
  READ(INFILE,*,ERR=998) ngb
!
! If we want collapsing, then read the group data
!
  IF (ngb > 0) THEN
   IF (ngb > mixtures(1)%neg) THEN
    CALL error( 'NGB greater than number of energy groups on RSP file' )
   END IF
   ALLOCATE(ngbfng(ngb),STAT=i)
   IF (i /= 0) THEN
    CALL error( 'Can not allocate ngbfng' )
```

```
    END IF
!
! Read group data
!
    WRITE(OUTFILE, *) 'Reading group data ...'
    READ(INFILE,*,ERR=998) (ngbfng(i),i=1,ngb)
!
   END IF
!
! Read number of zones in control rod
!
   WRITE(OUTFILE, *) 'Reading number of zones in control rod ...'
   READ(INFILE,*,ERR=998) cnumz
!
! Read the zone data
!
   WRITE(OUTFILE, *) 'Reading zone data ..'
   ALLOCATE(czones(cnumz),STAT=i)
   IF (i /= 0) THEN
    CALL error( 'Can not allocate czones' )
   END IF
   READ(INFILE,*,ERR=998) (czones(i),i=1,cnumz)
!
! Sort the zones (just to be sure)
   CALL sort(czones, cnumz)
!
! Read the zone numbers for rx and ry
!
   WRITE(OUTFILE, *) 'Reading zone numebrs for rx and ry ...'
   READ(INFILE,*,ERR=998) rx,ry,totx,toty
!
! Read dimension data
!
   WRITE(OUTFILE, *) 'Reading dimension data ...'
   READ(INFILE,*,ERR=998) a,pa,pxa,ppya
!
! Read wether or not to read D data
!
   WRITE(OUTFILE, *) 'Reading wether or not to read D ...'
   READ(INFILE,*,ERR=998) ownd

!
! If yes, read own D's
!
   IF (ownd) THEN
    WRITE(OUTFILE, *) 'Reading reading D ..'
    IF (ngb > 0) THEN
!write      write(*,*) ngb
     ALLOCATE(ds(ngb))
     READ(INFILE,*,ERR=998) ds
    ELSE
```

```
!write      write(*,*) mixtures(1)%neg
      ALLOCATE(ds(mixtures(1)%neg))
      READ(INFILE,*,ERR=998) ds
     END IF
    END IF
!
! Temporary - reading in number densities for each material
!
    WRITE(OUTFILE, *) 'Reading number of materials ..'
    READ(INFILE,*,ERR=998) nond
    IF (nond /= imix) THEN
     WRITE(str,*) 'Number of mixtures on input file (', nond, &
            ') not equal to number on RSP file (', imix, ')'
     CALL error(str)
    END IF

    DO i=1,nond
     WRITE(OUTFILE, *) 'Reading material data ..'
     READ(INFILE,*,ERR=998) j,nd
     DO k=1,imix
      IF (mixtures(k)%isotop == j) THEN
       mixtures(k)%ndens = nd
      END IF
     END DO
    END DO
!
! Everything read OK - close the file and exit
!
    CLOSE(INFILE)
    CALL leave_sr( 'readinfile' )
    RETURN
!
! Output File open error
!
997 CONTINUE
    CALL error( 'Can not open output file '// outputfile )
    CALL leave_sr( 'readfile' )
    RETURN
!
! Reading error
!
998 CONTINUE
    CALL error( 'Error reading input file '// inputfile )
    CALL leave_sr( 'readfile' )
    RETURN
!
! File open error
!
999 CONTINUE
    CALL error( 'Can not open input file '// inputfile )
    CALL leave_sr( 'readinfile' )
```

```
  RETURN

 END SUBROUTINE readinfile

!
!
!
 SUBROUTINE sort( arr, size )
!
! Sorts an integer array
!

  INTEGER,DIMENSION(:),INTENT(INOUT) :: arr
  INTEGER,INTENT( IN ) :: size
  INTEGER :: I,J,changed=1
  INTEGER :: temp

  CALL enter_sr( 'sort' )

  DO WHILE(changed == 1)
   changed = 0
   DO I=1,size-1
    IF (arr(I) > arr(I+1)) THEN
     temp = arr(I)
     arr(I) = arr(I+1)
     arr(I+1) = temp
     changed = 1
    END IF
   END DO
  END DO

  CALL leave_sr( 'sort' )

 END SUBROUTINE sort

END MODULE readinp_mod
```

READRSP_MOD.F90

```
MODULE readrsp_mod
! Module to read the rsp file

USE error_mod
USE mixdata_mod

IMPLICIT NONE
```

```
CHARACTER(LEN=256) :: rspfile='rsp.dat' ! rsp filename
INTEGER,PARAMETER  :: INRSP=81        ! rsp file unit

CONTAINS


!
!
!
 SUBROUTINE readrspfile
!
! Reads in data from rsp file
!
   INTEGER                  :: istat=0,1
   INTEGER                  :: isotop, neg
   REAL(KIND=4)               :: keff
   CHARACTER(LEN=60)          :: description
   REAL,DIMENSION(:),ALLOCATABLE  :: d,siga,nufis,fis,chi
   REAL,DIMENSION(:,:),ALLOCATABLE :: scat_mat,di_mat
   INTEGER                  :: idum
   REAL                     :: vol,lflux,rflux,lcur,rcur,avflux
!
!
!
   CALL enter_sr( 'readrspfile' )
!
! Open the rsp file
!
   OPEN(INRSP, FILE=rspfile, STATUS='OLD',ACTION='READ',&
      FORM='UNFORMATTED',ACCESS='SEQUENTIAL',IOSTAT=istat)
!
! Check opening
!
   IF (istat /= 0) THEN
     CALL error( 'Error opening rsp file '//rspfile )
   ENDIF
!
! Read heading data
!
   READ(INRSP)
   READ(INRSP)
   READ(INRSP)
!
! Find out the total number of mixtures by reading through the whole file
!
   READ(INRSP,IOSTAT=istat) isotop,neg,keff,description
!write     WRITE(*,*) isotop,neg,keff,description
!
   IMIX=IMIX+1
   DO WHILE (istat == 0)
    READ(INRSP)
```

```
    READ(INRSP)
    READ(INRSP)
    READ(INRSP,IOSTAT=istat) isotop,neg,keff,description
!write     WRITE(*,*) isotop,neg,keff,description
    IMIX=IMIX+1
   END DO
!
   IMIX=IMIX-1
!write    write(*,*) IMIX
!
! Assign mixtures array
!
   CALL allocate_mixtures( imix )
!
! Get back to beginning of file
!
   REWIND INRSP
!
! Read heading data
! !!!!!!!!!!!! Why 3 ????
!
   READ(INRSP)
   READ(INRSP)
   READ(INRSP)
!
! Allocate arrays
!
   ALLOCATE(d(neg),STAT=istat)
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate d memory' )
   END IF

   ALLOCATE(siga(neg),STAT=istat)
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate siga memory' )
   END IF

   ALLOCATE(nufis(neg),STAT=istat)
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate nufis memory' )
   END IF

   ALLOCATE(fis(neg),STAT=istat)
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate fis memory' )
   END IF

   ALLOCATE(chi(neg),STAT=istat)
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate chi memory' )
   END IF
```

```
   ALLOCATE(scat_mat(neg,neg),STAT=istat)
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate scat_mat memory' )
   END IF

   ALLOCATE(di_mat(neg,neg),STAT=istat)
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate di_mat memory' )
   END IF
!
! Read information
!
   DO I=1,IMIX
    READ(INRSP) isotop,neg,keff,description
    CALL add_mixture(a=isotop,b=neg,c=keff,d=description,num=I)
    READ(INRSP) d,siga,nufis,fis,chi,scat_mat
    CALL add_mixture(e=d,f=siga,g=nufis,h=fis,i=chi,j=scat_mat,num=I)
    READ(INRSP) di_mat
    CALL add_mixture(k=di_mat,num=I)
    READ(INRSP) idum,vol,d,siga,nufis,fis,chi
    CALL add_mixture(l=idum,m=vol,n=d,o=siga,p=nufis,&
              q=fis,r=chi,num=I)
!write    write(*,*) isotop, idum
   END DO

!
! Deallocate Arrays
!
   DEALLOCATE(d)
   DEALLOCATE(siga)
   DEALLOCATE(nufis)
   DEALLOCATE(fis)
   DEALLOCATE(scat_mat)
   DEALLOCATE(di_mat)
!
! Close File
!
   CLOSE(INRSP)
!
!
   CALL leave_sr( 'readrspfile' )
!
   RETURN

  END SUBROUTINE readrspfile

END MODULE readrsp_mod


!////////////////////// INPUT RSP FORMAT //////////////////////
```

```
!
!
! RECORD 1: ISOTOP,NEG,KEFF,DESCRIPTION (2I3,F12.6,A60)
!
! RECORD 2: D,SIGA,NUFIS,FIS,CHI,SCATT-MATRIX(COLUMN-WISE :
!        => SCAT(1<-1), SCAT(2<-1), ..., SCAT(NEG<-1), SCAT(1<-2),
!           SCAT(2<-2), SCAT(3<-2), ..., SCAT(NEG<-2), SCAT(1<-3),
!           ....., ....., ....., ....., SCAT(1<-NEG),
!           SCAT(2<-NEG), SCAT(3<-NEG),....., SCAT(NEG<-NEG)
!
! RECORD 3: I/D-MATRIX(COLUMN-WISE :
!        => 1/D(1<-1), 1/D(2<-1), ..., 1/D(NEG<-1), 1/D(1<-2),
!           1/D(2<-2), 1/D(3<-2), ..., 1/D(NEG<-2), 1/D(1<-3),
!           ....., ....., ..., ....., 1/D(1<-NEG),
!           1/D(2<-NEG), 1/D(3<-NEG),....., 1/D(NEG<-NEG)
!
! RECORD 4: IDUM ,VOL,LFLUX,RFLUX,LCUR,RCUR,AVFLUX
!
!
```

MIXDATA_MOD.F90

```
MODULE mixdata_mod
! Module for handling the mixure data

USE error_mod

IMPLICIT NONE

! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! MIX Type - contains all data for a certain mixture
! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

TYPE :: MIX

 INTEGER                :: isotop,number,zone,neg      ! General Data
 REAL                   :: keff                        ! General Data
 CHARACTER(LEN=60)      :: description                 ! General Data
 REAL,DIMENSION(:),POINTER  :: d,siga,nufis,fis,chi,downscat ! xsections
 REAL,DIMENSION(:,:),POINTER :: scat_mat, di_mat       ! xsections
 INTEGER                :: idum                        ! Same as Zone
 REAL                   :: vol                         ! Volume of zone
 REAL,DIMENSION(:),POINTER  :: lflux, rflux,lcur,rcur,avflux ! Fluxes + Cur
 REAL                   :: ndens                       ! Number density

END TYPE MIX

! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
TYPE (mix),DIMENSION(:),POINTER :: mixtures, zones ! Arrays for data
INTEGER                 :: imix=0       ! Number of mixtures
INTEGER                 :: izones       ! Number of zones
TYPE(MIX)                :: rod         ! Data for control rod

CONTAINS


!
!
!
 SUBROUTINE allocate_mixtures( num )
!
! Allocates the mixtures array
! NOTE: Does not allocate each mix in the array
!

  INTEGER,INTENT(IN) :: num    ! Size of array to allocate
  INTEGER         :: istat=0

  CALL enter_sr( 'allocate_mixtures' )

! Check if already allocates

  IF (ASSOCIATED(mixtures)) THEN
    CALL warning( 'Mixtures array already allocated - will deallocate ' )
    CALL deallocate_mixtures
  ENDIF

! Allocate mixtures

  ALLOCATE(mixtures(1:num),STAT=istat)

! Check if allocation successful

  IF (istat /= 0) THEN
   CALL error( 'Can not allocate mixture array' )
  ENDIF

! Assign the number of mixtures - imix

  imix = num

  CALL leave_sr( 'allocate_mixtures' )

 END SUBROUTINE allocate_mixtures


!
!
!
```

```
 SUBROUTINE allocate_mix( neg, m1 )
!
! Allocates all arrays in type mix using neg
! NOTE: Also initializes the arrays to 0
!

   INTEGER,INTENT(IN)   :: neg      ! Size of arrays
   TYPE(MIX),INTENT(OUT) :: m1      ! MIX to allocate
   INTEGER              :: istat,I,J

   CALL enter_sr( 'allocate_mix' )

! Set the number of groups in the mix

   m1%neg = neg

! Allocate arrays, with error checking

   ALLOCATE( m1%d(neg), STAT=istat )
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate memory' )
   ENDIF

   ALLOCATE( m1%siga(neg), STAT=istat )
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate memory' )
   ENDIF

   ALLOCATE( m1%nufis(neg), STAT=istat )
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate memory' )
   ENDIF

   ALLOCATE( m1%fis(neg), STAT=istat )
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate memory' )
   ENDIF

   ALLOCATE( m1%chi(neg), STAT=istat )
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate memory' )
   ENDIF

   ALLOCATE( m1%scat_mat(neg,neg), STAT=istat )
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate memory' )
   ENDIF

   ALLOCATE( m1%di_mat(neg,neg), STAT=istat )
   IF (istat /= 0) THEN
    CALL error( 'Can not allocate memory' )
```

```
  ENDIF

  ALLOCATE( m1%lflux(neg), STAT=istat )
  IF (istat /= 0) THEN
   CALL error( 'Can not allocate memory' )
  ENDIF

  ALLOCATE( m1%rflux(neg), STAT=istat )
  IF (istat /= 0) THEN
   CALL error( 'Can not allocate memory' )
  ENDIF

  ALLOCATE( m1%lcur(neg), STAT=istat )
  IF (istat /= 0) THEN
   CALL error( 'Can not allocate memory' )
  ENDIF

  ALLOCATE( m1%rcur(neg), STAT=istat )
  IF (istat /= 0) THEN
   CALL error( 'Can not allocate memory' )
  ENDIF

  ALLOCATE( m1%avflux(neg), STAT=istat )
  IF (istat /= 0) THEN
   CALL error( 'Can not allocate memory' )
  ENDIF

  ALLOCATE( m1%downscat(neg), STAT=istat )
  IF (istat /= 0) THEN
   CALL error( 'Can not allocate memory' )
  ENDIF

! Initialize arrays

  DO I=1,neg
   m1%d(I) = 0D0
   m1%siga(I) = 0D0
   m1%nufis(I) = 0D0
   m1%fis(I) = 0D0
   m1%chi(I) = 0D0
   m1%downscat(I) = 0D0
   m1%lflux(I) = 0D0
   m1%rflux(I) = 0D0
   m1%lcur(I) = 0D0
   m1%rcur(I) = 0D0
   m1%avflux(I) = 0D0

   DO J=1,neg
    m1%scat_mat(I,J) = 0D0
    m1%di_mat(I,J) = 0D0
   END DO
```

```
   END DO

   m1%ndens = 0D0

   CALL leave_sr( 'allocate_mix' )

  END SUBROUTINE allocate_mix


!
!
!
  SUBROUTINE deallocate_mix( p1 )
!
! Deallocates all memory associated with a mix
!

   TYPE(MIX) :: p1 ! MIX to deallocate

   CALL enter_sr( 'deallocate_mix' )

! Deallocate all arrays if allocated, with error checking

   IF (ASSOCIATED(p1%d)) THEN
    DEALLOCATE(p1%d)
   END IF
   IF (ASSOCIATED(p1%siga)) THEN
    DEALLOCATE(p1%siga)
   END IF
   IF (ASSOCIATED(p1%nufis)) THEN
    DEALLOCATE(p1%nufis)
   END IF
   IF (ASSOCIATED(p1%fis)) THEN
    DEALLOCATE(p1%fis)
   END IF
   IF (ASSOCIATED(p1%chi)) THEN
    DEALLOCATE(p1%chi)
   END IF
   IF (ASSOCIATED(p1%scat_mat)) THEN
    DEALLOCATE(p1%scat_mat)
   END IF
   IF (ASSOCIATED(p1%di_mat)) THEN
    DEALLOCATE(p1%di_mat)
   END IF
   IF (ASSOCIATED(p1%lflux)) THEN
    DEALLOCATE(p1%lflux)
   END IF
   IF (ASSOCIATED(p1%rflux)) THEN
    DEALLOCATE(p1%rflux)
   END IF
   IF (ASSOCIATED(p1%lcur)) THEN
```

```
   DEALLOCATE(p1%lcur)
  END IF
  IF (ASSOCIATED(p1%rcur)) THEN
   DEALLOCATE(p1%rcur)
  END IF
  IF (ASSOCIATED(p1%avflux)) THEN
   DEALLOCATE(p1%avflux)
  END IF

  CALL leave_sr( 'deallocate_mix' )

 END SUBROUTINE deallocate_mix


!
!
!
 SUBROUTINE deallocate_mixtures
!
! Deallocates mixtures array including each mix
!

  INTEGER         :: I
  TYPE(MIX),POINTER :: p1

  CALL enter_sr( 'deallocate_mixtures' )

! First check if associated, then deallocate each mix,
! then deallocate mixtures

  IF (ASSOCIATED(mixtures)) THEN
   DO I=1,IMIX
    p1 => mixtures(I)
    CALL deallocate_mix(p1)
   END DO
   DEALLOCATE(mixtures)
  ENDIF

  CALL leave_sr( 'deallocate_mixtures' )

 END SUBROUTINE deallocate_mixtures


!
!
!
 SUBROUTINE deallocate_zones
!
! Deallocates zones array
!
```

```
  INTEGER        :: I
  TYPE(MIX),POINTER :: p1

  CALL enter_sr( 'deallocate_zones' )

! First check if associated, then deallocate each mix,
! then deallocate zones

  IF (ASSOCIATED(zones)) THEN
   DO I=1,IZONES
    p1 => zones(I)
    CALL deallocate_mix(p1)
   END DO
   DEALLOCATE(zones)
  ENDIF

  CALL leave_sr( 'deallocate_zones' )

 END SUBROUTINE deallocate_zones


!
!
!
 SUBROUTINE copy_mix( from_m, to_m, neg )
!
! Copies the data from MIX from_m to MIX to_m
!
  TYPE(MIX),INTENT(IN)  :: from_m ! Source MIX
  TYPE(MIX),INTENT(OUT) :: to_m   ! Destination MIX
  INTEGER,INTENT(IN)    :: neg    ! Size
  INTEGER               :: I,J

  CALL enter_sr( 'copy_mix' )

! Deallocate to_m (just to make sure)

  CALL deallocate_mix( to_m )

! Allocate to_m with correct size

  CALL allocate_mix( neg, to_m )

! Set size

  to_m%neg = neg
  to_m%ndens = from_m%ndens

! Copy everything

  DO I=1,neg
```

```
    to_m%d(I) = from_m%d(I)
    to_m%siga(I) = from_m%siga(I)
    to_m%nufis(I) = from_m%nufis(I)
    to_m%fis(I) = from_m%fis(I)
    to_m%chi(I) = from_m%chi(I)
    to_m%lflux(I) = from_m%rflux(I)
    to_m%rflux(I) = from_m%rflux(I)
    to_m%lcur(I) = from_m%lcur(I)
    to_m%rcur(I) = from_m%rcur(I)
    to_m%avflux(I) = from_m%avflux(I)
    to_m%downscat(I) = from_m%downscat(I)

    DO J=1,neg
      to_m%scat_mat(I,J) = from_m%scat_mat(I,J)
      to_m%di_mat(I,J) = from_m%di_mat(I,J)
    END DO

  END DO

  CALL leave_sr( 'copy_mix' )

 END SUBROUTINE copy_mix


!
!
!
 SUBROUTINE calc_downscat( p1 )
!
! Calculates the down scattering using the scat_mat matrix
!

  TYPE (MIX),INTENT(INOUT) :: p1      ! MIX for which to calculate downscat
  INTEGER                  :: i,j,istat
  REAL                     :: sum=0D0

  CALL enter_sr( 'calc_downscat' )

  DO j=1,p1%neg
    DO i=j+1,p1%neg
      sum=sum+p1%scat_mat(i,j)
    END DO
    p1%downscat(j) = sum
    sum = 0D0
  END DO

  CALL leave_sr( 'calc_downscat' )

 END SUBROUTINE calc_downscat
```

```
!
!
!
 SUBROUTINE add_mixture( a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,num )
!
! Adds mixture data to a mixture in the mixtures array at position I
! MIX is allocated if required
!

  INTEGER,OPTIONAL          :: a,b
  REAL,OPTIONAL             :: c
  CHARACTER(LEN=60),OPTIONAL :: d
  REAL,DIMENSION(:),OPTIONAL :: e,f,g,h,i
  REAL,DIMENSION(:,:),OPTIONAL:: j,k
  INTEGER,OPTIONAL          :: l
  REAL,OPTIONAL             :: m
  REAL,DIMENSION(:),OPTIONAL :: n,o,p,q,r
  INTEGER                   :: num,istat,count,count2
  TYPE (mix),POINTER        :: p1

  CALL enter_sr( 'add_mixture' )

  p1 => mixtures(num)

! Start setting data

  IF (PRESENT(a)) THEN
   p1%isotop = a
   p1%zone   = a/10
   p1%number = MOD(a,10)
  ENDIF

  IF (PRESENT(b)) THEN
   p1%neg = b

! allocate all arrays

    CALL allocate_mix( b, p1 )

  ENDIF

  IF (PRESENT(c)) THEN
   p1%keff = c
  ENDIF

  IF (PRESENT(d)) THEN
   p1%description = d
  ENDIF

  IF (PRESENT(e)) THEN
```

```
 DO count=1,p1%neg
  p1%d(count)=e(count)
 END DO
ENDIF

IF (PRESENT(f)) THEN
 DO count=1,p1%neg
  p1%siga(count)=f(count)
 END DO
ENDIF

IF (PRESENT(g)) THEN
 DO count=1,p1%neg
  p1%nufis(count)=g(count)
 END DO
ENDIF

IF (PRESENT(h)) THEN
 DO count=1,p1%neg
  p1%fis(count)=h(count)
 END DO
ENDIF

IF (PRESENT(i)) THEN
 DO count=1,p1%neg
  p1%chi(count)=i(count)
 END DO
ENDIF

IF (PRESENT(j)) THEN
 DO count=1,p1%neg
  DO count2=1,p1%neg
   p1%scat_mat(count,count2)=j(count,count2)
  END DO
 END DO
 CALL calc_downscat(p1)
ENDIF

IF (PRESENT(k)) THEN
 DO count=1,p1%neg
  DO count2=1,p1%neg
   p1%di_mat(count,count2)=k(count,count2)
  END DO
 END DO
ENDIF

IF (PRESENT(l)) THEN
 p1%idum=l
 p1%zone=l
 IF ((p1%zone < 10) .AND. (p1%isotop > 100)) THEN
  p1%number = MOD(p1%isotop,100)
```

```
   ENDIF
  ENDIF

  IF (PRESENT(m)) THEN
   p1%vol = m
  ENDIF

  IF (PRESENT(n)) THEN
   DO count=1,p1%neg
    p1%lflux(count)=n(count)
   END DO
  ENDIF

  IF (PRESENT(o)) THEN
   DO count=1,p1%neg
    p1%rflux(count)=o(count)
   END DO
  ENDIF

  IF (PRESENT(p)) THEN
   DO count=1,p1%neg
    p1%lcur(count)=p(count)
   END DO
  ENDIF

  IF (PRESENT(q)) THEN
   DO count=1,p1%neg
    p1%rcur(count)=q(count)
   END DO
  ENDIF

  IF (PRESENT(r)) THEN
   DO count=1,p1%neg
    p1%avflux(count)=r(count)
   END DO
  ENDIF

! Just making sure of cleaning up by nullifying p1

  NULLIFY(p1)

  CALL leave_sr( 'add_mixture' )

 END SUBROUTINE add_mixture


!
!
!
 SUBROUTINE show_mixture( p1, num , ioption )
!
```

```
! Shows data on a mixture given by p1
!

  INTEGER   :: i,j,num  ! num - number in mixtures array
  INTEGER   :: ioption  ! 1,0 display/do not display detail data
  TYPE (MIX) :: p1

  CALL enter_sr( 'show_mixture' )

  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'Mixture Number: ', num
  WRITE(OUTFILE,*) '=================='
  WRITE(OUTFILE,104) 'ISOTOP: ', p1%isotop
  WRITE(OUTFILE,104) 'Material Number: ', p1%number
  WRITE(OUTFILE,104) 'Zone Number: ', p1%zone
  WRITE(OUTFILE,105) 'K-Effective: ', p1%keff
  WRITE(OUTFILE,104) 'Number of Energy Groups: ', p1%neg
  WRITE(OUTFILE,105) 'Number Density: ', p1%ndens
  WRITE(OUTFILE,106) 'Description: ', p1%description
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,104) 'IDUM: ', p1%idum
  WRITE(OUTFILE,105) 'Volume: ', p1%vol
  WRITE(OUTFILE,*)

  IF (ioption == 0) THEN
   RETURN
  ENDIF

  WRITE(OUTFILE,100) 'DIF ','ABS ','NU*F','FISS',' CHI','SCAT'
  DO i=1,p1%neg
   WRITE(OUTFILE,101) i,p1%d(i),p1%siga(i),p1%nufis(i), p1%fis(i), p1%chi(i),&
         p1%downscat(i)
  END DO

  WRITE(OUTFILE,102)

  DO i=1,p1%neg
   WRITE(OUTFILE,103) i,(p1%scat_mat(i,j),j=1,p1%neg)
  END DO

  WRITE(OUTFILE,107) 'LFL ','RFL ','LCUR','RCUR','AVFL'
  DO i=1,p1%neg
   WRITE(OUTFILE,108) i,p1%lflux(i),p1%rflux(i),p1%lcur(i), p1%rcur(i),&
         p1%avflux(i)
  END DO

  CALL leave_sr( 'show_mixture' )

  RETURN

100 FORMAT(/1X,' G  ',6(5X,A4,5X)/1X,88('-'))
```

```
101 FORMAT(1X,I2,2X,1P6D14.6)
102 FORMAT(/1X,'SCATTERING MATRIX:(I<-J)')
103 FORMAT(1X,I2,2X,1P9D14.6/(5X,1P9D14.6))
104 FORMAT(T2,A25,T30,I12)
105 FORMAT(T2,A25,T30,F12.5)
106 FORMAT(T2,A25,T30,A60)
107 FORMAT(/1X,' G  ',5(5X,A4,5X)/1X,74('-'))
108 FORMAT(1X,I2,2X,1P5D14.6)

 END SUBROUTINE show_mixture


!
!
!
 SUBROUTINE show_zone( p1, num , ioption )
!
! Show details on a zone p1 with number num in zones array
!

  INTEGER :: i,j,num ! num - number of zone in zone array
  INTEGER :: ioption ! 1,0 display/do not display detail data
  TYPE (MIX) :: p1

  CALL enter_sr( 'show_zone' )

  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'Zone Number: ', num
  WRITE(OUTFILE,*) '=========================='
  WRITE(OUTFILE,104) 'Zone Number: ', p1%zone
  WRITE(OUTFILE,104) 'Number of Energy Groups: ', p1%neg
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,105) 'Volume: ', p1%vol
  WRITE(OUTFILE,*)

  IF (ioption == 0) THEN
   RETURN
  ENDIF

  WRITE(OUTFILE,100) 'DIF ','ABS ','NU*F','FISS',' CHI','SCAT'
  DO i=1,p1%neg
   WRITE(OUTFILE,101) i,p1%d(i),p1%siga(i),p1%nufis(i), p1%fis(i), p1%chi(i),&
         p1%downscat(i)
  END DO

  WRITE(OUTFILE,102)

  DO i=1,p1%neg
   WRITE(OUTFILE,103) i,(p1%scat_mat(i,j),j=1,p1%neg)
  END DO
```

```
  WRITE(OUTFILE,107) 'LFL ','RFL ','LCUR','RCUR','AVFL'
  DO i=1,p1%neg
   WRITE(OUTFILE,108) i,p1%lflux(i),p1%rflux(i),p1%lcur(i), p1%rcur(i),&
         p1%avflux(i)
  END DO

  CALL leave_sr( 'show_zone' )

  RETURN

100 FORMAT(/1X,' G  ',6(5X,A4,5X)/1X,88('-'))
101 FORMAT(1X,I2,2X,1P6D14.6)
102 FORMAT(/1X,'SCATTERING MATRIX:(I<-J)')
103 FORMAT(1X,I2,2X,1P9D14.6/(5X,1P9D14.6))
104 FORMAT(T2,A25,T30,I12)
105 FORMAT(T2,A25,T30,F12.5)
106 FORMAT(T2,A25,T30,A60)
107 FORMAT(/1X,' G  ',5(5X,A4,5X)/1X,74('-'))
108 FORMAT(1X,I2,2X,1P5D14.6)

 END SUBROUTINE show_zone

!
!
!
 SUBROUTINE show_all_mixtures( ioption )
!
! Show all mixtures in the mixtures array
!

  INTEGER,INTENT(IN) :: ioption
  INTEGER :: I

  CALL enter_sr( 'show_all_mixtures' )

  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*)' ************************************************************'
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'MIXTURE INFORMATION'
  WRITE(OUTFILE,*) '-------------------'
  WRITE(OUTFILE,*)

  DO I=1,imix
   CALL show_mixture( mixtures(I), I, ioption )
   WRITE(OUTFILE,*)
   WRITE(OUTFILE,*)
  END DO

  WRITE(OUTFILE,*) ' ************************************************************'

  CALL leave_sr( 'show_all_mixtures' )
```

```
 END SUBROUTINE show_all_mixtures

!
!
!
 SUBROUTINE show_all_zones( ioption )
!
! Show all zones in the zones array
!

  INTEGER,INTENT(IN) :: ioption
  INTEGER :: I

  CALL enter_sr( 'show_all_zones' )

  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) ' ************************************************************'
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'ZONE INFORMATION'
  WRITE(OUTFILE,*) '----------------'
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'NOTE: Macroscopic Cross-Sections!'
  WRITE(OUTFILE,*)

  DO I=1,izones
   CALL show_zone( zones(I), I, ioption )
   WRITE(OUTFILE,*)
   WRITE(OUTFILE,*)
  END DO

  WRITE(OUTFILE,*) ' ************************************************************'

  CALL leave_sr( 'show_all_zones' )

 END SUBROUTINE show_all_zones


!
!
!
 SUBROUTINE create_zone_data
!
! Create zone data from mixture data
!

  INTEGER :: I,J,K,L

  CALL enter_sr( 'create_zone_data' )

! Find the number of zones and set izones
```

```
  CALL find_num_zones( izones )

!write   WRITE(*,*) izones

! Allocate the zones array

  ALLOCATE(zones(izones),STAT=I)

  IF (I /= 0) THEN
   CALL error( 'Can not allocate memory' )
  END IF

! Set data

  DO I=1,izones

   CALL allocate_mix( mixtures(1)%neg, zones(I))

   DO J=1,imix

    IF (mixtures(J)%zone == I) THEN

     zones(I)%neg = mixtures(J)%neg
     zones(I)%zone = I
     zones(I)%vol = mixtures(J)%vol

     DO K=1,mixtures(J)%neg
      zones(I)%d(K)    = zones(I)%d(K) + mixtures(J)%d(K)
      zones(I)%siga(K) = zones(I)%siga(K) + mixtures(J)%siga(K)*mixtures(j)%ndens
      zones(I)%nufis(K)= zones(I)%nufis(K) + mixtures(J)%nufis(K)*mixtures(j)%ndens
      zones(I)%fis(K)  = zones(I)%fis(K) + mixtures(J)%fis(K)*mixtures(j)%ndens
      zones(I)%chi(K)  = zones(I)%chi(K) + mixtures(J)%chi(K)*mixtures(j)%ndens
      zones(I)%downscat(K) = zones(I)%downscat(K) &
                  + mixtures(J)%downscat(K)*mixtures(j)%ndens
      zones(I)%lflux(K) = mixtures(J)%lflux(K)
      zones(I)%rflux(K) = mixtures(J)%rflux(K)
      zones(I)%lcur(K) = mixtures(J)%lcur(K)
      zones(I)%rcur(K) = mixtures(J)%rcur(K)
      zones(I)%avflux(K) = mixtures(J)%avflux(K)

      DO L=1,mixtures(J)%neg
       zones(I)%scat_mat(K,L) = zones(I)%scat_mat(K,L) + &
                  mixtures(J)%scat_mat(K,L)*mixtures(j)%ndens
       zones(I)%di_mat(K,L) = zones(I)%di_mat(K,L) + &
                  mixtures(J)%di_mat(K,L)
      END DO

     END DO

    END IF
```

```
    END DO

   END DO

   CALL leave_sr( 'create_zone_data' )

  END SUBROUTINE create_zone_data

!
!
!
  SUBROUTINE find_num_zones( izones )
!
! Find the number of zones from the mixture data
!

   INTEGER,INTENT(OUT) :: izones
   INTEGER :: I

   CALL enter_sr( 'find_num_zones' )

   izones = 1
   DO I=1,imix
    IF (mixtures(I)%zone > izones) THEN
      izones = mixtures(I)%zone
    END IF
   END DO


   CALL leave_sr( 'find_num_zones' )

  END SUBROUTINE find_num_zones


END MODULE mixdata_mod

!////////////////////// INPUT RSP FORMAT //////////////////////
!
!
! RECORD 1: ISOTOP,NEG,KEFF,DESCRIPTION (2I3,F12.6,A60)
!
! RECORD 2: D,SIGA,NUFIS,FIS,CHI,SCATT-MATRIX(COLUMN-WISE :
!         => SCAT(1<-1), SCAT(2<-1), ..., SCAT(NEG<-1), SCAT(1<-2),
!           SCAT(2<-2), SCAT(3<-2), ..., SCAT(NEG<-2), SCAT(1<-3),
!           ....., ....., ....., ....., SCAT(1<-NEG),
!           SCAT(2<-NEG), SCAT(3<-NEG),....., SCAT(NEG<-NEG)
!
! RECORD 3: I/D-MATRIX(COLUMN-WISE :
!         => 1/D(1<-1), 1/D(2<-1), ..., 1/D(NEG<-1), 1/D(1<-2),
!           1/D(2<-2), 1/D(3<-2), ..., 1/D(NEG<-2), 1/D(1<-3),
```

```
!          ....., ....., ..., ....., 1/D(1<-NEG),
!          1/D(2<-NEG), 1/D(3<-NEG),....., 1/D(NEG<-NEG)
!
! RECORD 4: IDUM ,VOL,LFLUX,RFLUX,LCUR,RCUR,AVFLUX
!
!
```

XSEC_MOD.F90

```
MODULE xsec_mod
! Module for calculating the xsections in the control rod region

USE error_mod
USE readrsp_mod
USE readinp_mod
USE mixdata_mod
USE diff_mod

IMPLICIT NONE

CONTAINS

!
!
!
 SUBROUTINE rod_xsec( cnumz, czones )
!
! Determines xsection data for control rod
!

  INTEGER,INTENT(IN)          :: cnumz      ! Number of zones in rod
  INTEGER,DIMENSION(:),INTENT(IN) :: czones     ! Zone numbers
  INTEGER                     :: I,J,K,L,istat
  TYPE(MIX),POINTER           :: p1
  REAL,POINTER,DIMENSION(:)   :: avftvol

  CALL enter_sr( 'rod_xsec' )

! Allocate and initialize the rod

  CALL allocate_mix( zones(1)%neg,rod )
  rod%vol = 0D0
  rod%neg = zones(1)%neg
  ALLOCATE(avftvol(rod%neg),STAT=I)
  IF (I /= 0) THEN
   CALL error( 'Can not allocate avftvol' )
  END IF
  DO I=1,rod%neg
   avftvol(I)=0D0
```

```
  END DO

! Start combinig zone data

  DO I=1,cnumz

   DO J=1,izones

    IF (J == czones(I)) THEN ! In correct Zone

     p1 => zones(J)

! Set rod volume

      rod%vol = rod%vol + p1%vol

      DO K=1,rod%neg
       rod%d(K) = rod%d(K) + p1%siga(K)*p1%vol*p1%avflux(K)
       rod%siga(K) = rod%siga(K) + p1%siga(K)*p1%vol*p1%avflux(K)
       rod%nufis(K) = rod%nufis(K) + p1%nufis(K)*p1%vol*p1%avflux(K)
       rod%fis(K) = rod%fis(K) + p1%fis(K)*p1%vol*p1%avflux(K)
       rod%chi(K) = rod%chi(K) + p1%chi(K)*p1%vol*p1%avflux(K)
       rod%downscat(K) = rod%downscat(K) + &
                  p1%downscat(K)*p1%vol*p1%avflux(K)
       rod%avflux(K) = rod%avflux(K) + p1%avflux(K)*p1%vol
       avftvol(K) = avftvol(K) + p1%avflux(K)*p1%vol

       DO L=1,rod%neg
        rod%scat_mat(K,L) = rod%scat_mat(K,L) + &
                    p1%scat_mat(K,L)*p1%vol*p1%avflux(K)
        rod%di_mat(K,L) = rod%di_mat(K,L) + &
                  p1%di_mat(K,L)*p1%vol*p1%avflux(K)
       END DO

      END DO

     END IF

    END DO

   END DO

   DO K=1,rod%neg
    rod%avflux(K) = rod%avflux(K) / rod%vol
    rod%d(K) = rod%d(K) / avftvol(K)
    rod%siga(K) = rod%siga(K) / avftvol(K)
    rod%nufis(K) = rod%nufis(K) / avftvol(K)
    rod%fis(K) = rod%fis(K) / avftvol(K)
    rod%chi(K) = rod%chi(K) / avftvol(K)
    rod%downscat(K) = rod%downscat(K) / avftvol(K)
```

```
    DO L=1,rod%neg
     rod%scat_mat(K,L) = rod%scat_mat(K,L) / avftvol(K)
     rod%di_mat(K,L) = rod%di_mat(K,L) / avftvol(K)
    END DO
   END DO

   CALL calc_downscat( rod )

! Get the left current => lcur of the first zone

   DO K=1,rod%neg
    rod%lcur(K) = zones(czones(1))%lcur(K)
    rod%lflux(K) = zones(czones(1))%lflux(K)
   END DO

! Get the right current => rcur of the last zone

   DO K=1,zones(czones(cnumz))%neg
    rod%rcur(K) = zones(czones(cnumz))%rcur(K)
    rod%rflux(K) = zones(czones(cnumz))%rflux(K)
   END DO

   CALL leave_sr( 'rod_xsec' )

  END SUBROUTINE rod_xsec


!
!
!
  SUBROUTINE show_xsecs
!
! Show cross-section data
!

   INTEGER :: I,J
   REAL    :: d0,leakage,xflux,yflux,radius,xrad,yrad
   CHARACTER(LEN=256) :: str

   CALL enter_sr( 'show_xsecs' )

   WRITE(OUTFILE,*)
   WRITE(OUTFILE,*)
   WRITE(OUTFILE,*) '-----------------------------------'
   WRITE(OUTFILE,*) 'CROSS-SECTIONS FOR CONTROL ROD REGION'
   WRITE(OUTFILE,*) '-----------------------------------'
   WRITE(OUTFILE,*)
   WRITE(OUTFILE,*) 'Control Rod Data:'
   WRITE(OUTFILE,*) '================='
   WRITE(OUTFILE,*)
   WRITE(OUTFILE,*) 'Number of Zones in Control rod:', cnumz
```

```
WRITE(OUTFILE,*) 'Control Rod Zones:', (czones(I),I=1,cnumz)
WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'Width of control rod:', a
WRITE(OUTFILE,*) 'Height of control rod:', pa
WRITE(OUTFILE,*) 'Volume of control rod:', rod%vol
WRITE(OUTFILE,*)
radius = radii(czones(cnumz))
IF (totx) THEN
 xrad = (((radii(rx)-radii(rx-1))/2)+radii(rx-1)-radius)
ELSE
 xrad = radii(rx) - radius
END IF
IF (toty) THEN
 yrad = (((radii(ry)-radii(ry-1))/2)+radii(ry-1)-radius)
ELSE
 yrad = radii(ry) - radius
END IF
WRITE(OUTFILE,*) 'Distance from CR to center of x neighbour:', xrad
WRITE(OUTFILE,*) 'Distance from CR to center of y neighbour:', yrad
WRITE(OUTFILE,*) 'Width of x neighbour:', pxa
WRITE(OUTFILE,*) 'Height of y neighbour:', ppya
WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'a:', a
WRITE(OUTFILE,*) 'p:', pa/a
WRITE(OUTFILE,*) 'px:', pxa/a
WRITE(OUTFILE,*) 'py:', ppya/((pa/a)*a)
WRITE(OUTFILE,*)

WRITE(OUTFILE,100) 'DIF ','ABS ','NU*F','FISS',' CHI','SCAT'
DO i=1,rod%neg
 WRITE(OUTFILE,101) i,rod%d(i),rod%siga(i),rod%nufis(i), rod%fis(i),&
        rod%chi(i),&
        rod%downscat(i)
END DO

WRITE(OUTFILE,102)

DO i=1,rod%neg
 WRITE(OUTFILE,103) i,(rod%scat_mat(i,j),j=1,rod%neg)
END DO

WRITE(OUTFILE,107) 'LFL ','RFL ','LCUR','RCUR','AVFL'
DO i=1,rod%neg
 WRITE(OUTFILE,108) i,rod%lflux(i),rod%rflux(i),rod%lcur(i), rod%rcur(i),&
        rod%avflux(i)
END DO

WRITE(OUTFILE,111) 'FLX','FLY','LEAKAGE','D0'
DO i=1,rod%neg
 CALL calc_d0(i,d0)
 CALL get_leakage(i,leakage)
```

```
    IF (totx) THEN
      xflux = zones(rx)%avflux(i)
    ELSE
      xflux = zones(rx)%rflux(i)
    END IF

    IF (toty) THEN
      yflux = zones(ry)%avflux(i)
    ELSE
      yflux = zones(ry)%rflux(i)
    END IF
    WRITE(OUTFILE,112) i,xflux,yflux,leakage,d0
  END DO

  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'FINAL RESULTS FOR CITATION'
  WRITE(OUTFILE,*) '--------------------------'
  WRITE(OUTFILE,*)
  WRITE(OUTFILE,109) 'D   ','ABS ','REM'
  DO I=1,rod%neg
    WRITE(OUTFILE,110) I,rod%d(I),rod%siga(I),rod%downscat(I)
  END DO

  WRITE(OUTFILE,*)

  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) 'INPUT TO VSOP'
  WRITE(OUTFILE,*) '-------------'
  WRITE(OUTFILE,*)
  WRITE(str,*) '(1P',rod%neg,'E12.5)'
  WRITE(OUTFILE,str) rod%siga
  WRITE(OUTFILE,str) rod%downscat
  WRITE(OUTFILE,114) 1D0,1D0,1D0,1
  WRITE(OUTFILE,str) rod%d


  CALL leave_sr( 'show_xsecs' )

  RETURN

100 FORMAT(/1X,' G  ',6(5X,A4,5X)/1X,88('-'))
101 FORMAT(1X,I2,2X,1P6D14.6)
102 FORMAT(/1X,'SCATTERING MATRIX:(I<-J)')
103 FORMAT(1X,I2,2X,1P9D14.6/(5X,1P9D14.6))
104 FORMAT(T2,A25,T30,I12)
105 FORMAT(T2,A25,T30,F12.5)
106 FORMAT(T2,A25,T30,A60)
107 FORMAT(/1X,' G  ',5(5X,A4,5X)/1X,74('-'))
108 FORMAT(1X,I2,2X,1P5D14.6)
109 FORMAT(/1X,' G  ',3(5X,A4,5X)/1X,46('-'))
```

```
110 FORMAT(1X,I2,2X,1P3D14.6)
111 FORMAT(/1X,' G  ',4(5X,A4,5X)/1X,74('-'))
112 FORMAT(1X,I2,2X,1P4D14.6)
! 113 FORMAT(1P6E12.5,4X,'C7-',I1)
114 FORMAT(1P3E12.5,I3)

 END SUBROUTINE show_xsecs

END MODULE xsec_mod
```

DIFF_MOD.F90

```
MODULE diff_mod
! Module for calculating diffusion constants

USE error_mod
USE mixdata_mod
USE readinp_mod

IMPLICIT NONE

REAL,ALLOCATABLE,DIMENSION(:) :: radii          ! radii of each zone
REAL                  :: radius,irad    ! radius of control rod region
REAL,PARAMETER            :: PI = 3.14159   ! PI
REAL,PARAMETER            :: accuracy = 1D-3 ! Accuracy for volumes/radii

CONTAINS


!
!
!
 SUBROUTINE calc_radii
!
! Calculates the radii of each zone
! NOTE: assumes zones increase radially outward
!

   INTEGER :: I,J
   REAL   :: volume

   CALL enter_sr( 'calc_radii' )

   ALLOCATE(radii(izones), STAT=I)

   IF (I /= 0) THEN
    CALL error( 'Can not allocate momory' )
```

```
  END IF

  DO I=1,izones

   IF (I == 1) THEN
    radii(I) = SQRT(zones(I)%vol/PI)
   ELSE
    radii(I) = SQRT((zones(I)%vol+PI*radii(I-1)*radii(I-1))/PI)
   END IF

!write    WRITE(*,*) radii(I)

  END DO

  CALL leave_sr( 'calc_radii' )

 END SUBROUTINE calc_radii


!
!
!
 SUBROUTINE check_input

  INTEGER :: I,J
  REAL :: volume,v=0D0,t

  CALL enter_sr( 'check_input' )

! Check if volume of control rod region is correct

  volume = a * pa

  DO I=1,cnumz
   J = 1
   DO WHILE(mixtures(J)%zone /= czones(I))
    J=J+1
   END DO
   v = v + mixtures(J)%vol
  END DO

  WRITE(OUTFILE,*)
  WRITE(OUTFILE,*) '------------------'
  WRITE(OUTFILE,*) 'CHECKING INPUT DATA'
  WRITE(OUTFILE,*) '------------------'
  WRITE(OUTFILE,*)


  WRITE(OUTFILE,*) 'Control Rod Volume'
  WRITE(OUTFILE,*) '------------------'
  WRITE(OUTFILE,'(1X,A37,F12.5)') 'xsdrnpm (Total Volume of cr zones): ', rod%vol
```

```
WRITE(OUTFILE,'(1X,A37,F12.5)') 'TOTNEW Input (a*pa): ', volume

IF ( ABS(volume - rod%vol) > accuracy) THEN
 CALL warning( 'Control rod volumes do not match' )
ENDIF

IF (czones(1) == 1) THEN
 radius = radii(czones(cnumz))
ELSE
 radius = radii(czones(cnumz))
 irad = radii(czones(1))
END IF

WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'Radius (xsdrnpm): ', radius
WRITE(OUTFILE,*)


! Check positions of rx and ry

WRITE(OUTFILE,*) 'Size of Neighbouring Regions'
WRITE(OUTFILE,*) '----------------------------'
WRITE(OUTFILE,*)
WRITE(OUTFILE,*) 'X Neighbour Width'
IF (totx) THEN
 WRITE(OUTFILE,'(1X,A6,F12.5)') 'rx  : ', &
       (2*(((radii(rx)-radii(rx-1))/2)+radii(rx-1)-radius))
 t = ABS(pxa - (2*(((radii(rx)-radii(rx-1))/2)+radii(rx-1)-radius)))
ELSE
 WRITE(OUTFILE,'(1X,A6,F12.5)') 'rx  : ', (2*(radii(rx)-radius))
 t = ABS(pxa - (2*(radii(rx)-radius)))
END IF
WRITE(OUTFILE,'(1X,A6,F12.5)') 'pxa : ', pxa
WRITE(OUTFILE,*)

IF (t > accuracy) THEN
 CALL warning('x radii do not match')
END IF

WRITE(OUTFILE,*) 'Y Neighbour Height'
IF (toty) THEN
 WRITE(OUTFILE,'(1X,A6,F12.5)') 'ry  : ', &
       (2*(((radii(ry)-radii(ry-1))/2)+radii(ry-1)-radius))
 t = ABS(ppya - (2*(((radii(ry)-radii(ry-1))/2)+radii(ry-1)-radius)))
ELSE
 WRITE(OUTFILE,'(1X,A6,F12.5)') 'ry  : ', (2*(radii(ry)-radius))
 t = ABS(ppya - (2*(radii(ry)-radius)))
END IF
WRITE(OUTFILE,'(1X,A6,F12.5)') 'ppya : ', ppya
WRITE(OUTFILE,*)
```

```
  IF ( t > accuracy) THEN
   CALL warning('y radii do not match')
  END IF

  CALL leave_sr( 'check_input' )

 END SUBROUTINE check_input


!
!
!
 SUBROUTINE get_leakage( num, leakage )

  INTEGER,INTENT(IN) :: num
  REAL,INTENT(OUT)   :: leakage

  CALL enter_sr( 'get_leakage' )

!write   write(*,*) 'HIER1',rod%lcur(num),rod%rcur(num)
  leakage = rod%lcur(num) - rod%rcur(num)

  CALL leave_sr( 'get_leakage' )

 END SUBROUTINE get_leakage

!
!
!
 SUBROUTINE calc_d0( num, d0 )

  INTEGER,INTENT(IN) :: num
  REAL,INTENT(OUT)   :: d0
  REAL               :: d1, d2

  CALL enter_sr( 'calc_d0' )

  IF (ownd) THEN
   d0 = ds(num)
   RETURN
  END IF

  d1 = (zones(rx)%d(num)*zones(rx)%vol + &
      zones(rx+1)%d(num)*zones(rx+1)%vol) / &
      (zones(rx)%vol + zones(rx+1)%vol)
!write   write(*,*) "d1=",d1,zones(rx)%vol + zones(rx+1)%vol, ry
!write   write(*,*) zones(ry)%vol + zones(ry+1)%vol, &
!write              zones(ry)%d(num)*zones(ry)%vol,  &
!write              zones(ry+1)%d(num)*zones(ry+1)%vol
  d2 = (zones(ry)%d(num)*zones(ry)%vol + &
      zones(ry+1)%d(num)*zones(ry+1)%vol) / &
```

```
        (zones(ry)%vol + zones(ry+1)%vol)
!write    write(*,*) "d2=",d2
    d0 = (d1+d2)/2
!    d0 = (zones(rx)%d(num)*zones(rx)%vol + &
!          zones(ry)%d(num)*zones(ry)%vol) / &
!         (zones(rx)%vol + zones(ry)%vol)


    CALL leave_sr( 'calc_d0' )

  END SUBROUTINE calc_d0


!
!
!
  SUBROUTINE calc_r( num, r )

    INTEGER,INTENT(IN) :: num
    REAL,INTENT(OUT)   :: r
    REAL               :: leakage
    REAL               :: p,px,py
    REAL               :: xflux, yflux, aflux, d0

    CALL enter_sr( 'calc_r' )

    p = pa / a

    px = pxa / a

    py = ppya / (p*a)

!write    write(*,*) rx, ry,num
    IF (totx) THEN
      xflux = zones(rx)%avflux(num)
    ELSE
      xflux = zones(rx)%rflux(num)
    END IF

    IF (toty) THEN
      yflux = zones(ry)%avflux(num)
    ELSE
      yflux = zones(ry)%rflux(num)
    END IF
    aflux = rod%avflux(num)

    CALL calc_d0(num,d0)

    CALL get_leakage( num, leakage )

!write    WRITE(*,*) a,p,px,py
```

```
!write   WRITE(*,*) xflux, yflux, aflux
!write   WRITE(*,*) 'LEAKAGE= ',leakage, 'D0= ', d0
   r = ((2*p)*((xflux - aflux)/leakage)) + &
      ((2/p)*((yflux - aflux)/leakage)) + &
      ((px - py)/(2*d0))

   r = r*r

!write   WRITE(*,*) r

   r = r + ((4*p)*(py-px)*((xflux - aflux)/leakage)*(1/d0))

!write   WRITE(*,*) r

!write   WRITE(*,*) 'Group: ', num, 'r: ', r

   CALL leave_sr( 'calc_r' )

 END SUBROUTINE calc_r


!
!
!
 SUBROUTINE calc_id( num, id )

   INTEGER,INTENT(IN) :: num
   REAL,INTENT(OUT)   :: id
   REAL               :: r
   REAL               :: leakage
   REAL               :: p,px,py
   REAL               :: xflux, yflux, aflux, d0

   CALL enter_sr( 'calc_id' )

   CALL calc_r( num, r )

   p = pa / a

   px = pxa / a

   py = ppya / (p*a)

   IF (totx) THEN
    xflux = zones(rx)%avflux(num)
   ELSE
    xflux = zones(rx)%rflux(num)
   END IF

   IF (toty) THEN
    yflux = zones(ry)%avflux(num)
```

```
   ELSE
    yflux = zones(ry)%rflux(num)
   END IF
   aflux = rod%avflux(num)

   CALL calc_d0(num,d0)

   CALL get_leakage( num, leakage )

   id = ((2*p)*((xflux-aflux)/leakage)) +    &
      ((2/p)*((yflux - aflux)/leakage)) - &
      ((px + py)/(2*d0)) +              &
      SQRT(R)

!write   WRITE(*,*) 'Group: ', num, 'id: ', id, ((2*p)*((xflux-aflux)/leakage)), &
!write                              ((2/p)*((yflux - aflux)/leakage)), &
!write                              ((px + py)/(2*d0)), &
!write                              SQRT(R), xflux, yflux, aflux, &
!write                              leakage,d0

   CALL leave_sr( 'calc_id' )

  END SUBROUTINE calc_id


!
!
!
  SUBROUTINE rod_d
!
! Calculates the diffusion coefficient for the citation rod
!

   INTEGER :: I,J
   REAL :: id

   CALL enter_sr( 'rod_d' )

! Get the radii for each zone

   CALL calc_radii

! Check dimensions in input are correct

   CALL check_input

! Calculate d

   DO I=1,rod%neg
    CALL calc_id( I, id )
    rod%d(I) = 1/id
```

```
  END DO

  CALL leave_sr( 'rod_d' )

 END SUBROUTINE rod_d


END MODULE diff_mod
```