

Aalto University
School of Science
Degree Programme in Engineering Physics and Mathematics

Antti Rintala

Development and validation of Apros multigroup nodal diffusion model

Master's thesis
Espoo, October 12, 2015

Supervisor: Prof. Filip Tuomisto
Instructor: D.Sc. (Tech.) Markus Airila

Author:	Antti Rintala		
Title:	Development and validation of Apros multigroup nodal diffusion model		
Date:	October 12, 2015	Pages:	viii + 94
Major:	Advanced Energy Systems	Code:	Tfy-56
Supervisor:	Prof. Filip Tuomisto		
Instructor:	D.Sc. (Tech.) Markus Airila		
<p>The development of a steady state and transient multigroup nodal diffusion model for process simulation software Apros was continued and the models were validated. The initial implementation of the model was performed in 2009 and it has not been under continuous development afterwards.</p> <p>Some errors in the steady state model were corrected. The transient model was found to be incorrect. The solution method of the transient model was derived, and the program code not common with the steady state model was almost completely rewritten. The temporal solution method of the transient model is based on implicit discretization of the flux time derivative. The spatial solution method is similar to that of the steady state model with small modifications. Some features such as a temporal flux extrapolation transient acceleration method were implemented into the models.</p> <p>After this work, the steady state model is correctly implemented based on comparisons against DIF3D. The multiplication factors calculated with the steady state model and Serpent show relatively high differences, but the power distribution differences are on an acceptable level. The reactor total power time behaviors without simulating thermal hydraulics were similar to those calculated with DYN3D. With a step withdrawal of control rods, the total power time evolution was more different between the codes. However, the relative difference was not higher than in the case of a slow insertion of control rods. The differences of relative axial powers were small between the codes. The differences can be concluded to be caused by the differences between the solution methods of the codes. Therefore, the transient model is also correctly implemented and it produces results similar to DYN3D.</p> <p>The basic features of a steady state and transient multigroup nodal diffusion model have now been implemented in Apros and verified to work. Multiple suggestions for continuing the development of the model are proposed.</p>			
Keywords:	Apros, DIF3D, DYN3D, Generation IV, fast reactors, multi-group, nodal diffusion method, Serpent, SFR		
Language:	English		

Tekijä:	Antti Rintala		
Työn nimi:	Aprosin moniryhmänodaalidiffuusiomallin kehittäminen ja validointi		
Päiväys:	12. lokakuuta 2015	Sivumäärä:	viii + 94
Pääaine:	Energiatieteet	Koodi:	Tfy-56
Valvoja:	Prof. Filip Tuomisto		
Ohjaaja:	TkT Markus Airila		
<p>Prosessisimulointiohjelmisto Aprosin moniryhmänodaalidiffuusiomallin tasapaino- ja transienttilaskennan kehittämistä jatkettiin sekä kyseiset mallit validoitiin. Mallin alkuperäinen toteutus on vuodelta 2009, eikä malli ole ollut jatkuvassa kehityksessä toteutuksen jälkeen.</p> <p>Tasapainomallista korjattiin joitain virheitä. Transienttimalli havaittiin virheeliseksi. Transienttimallin ratkaisumenetelmä johdettiin työn aikana, sekä mallin tasapainomallista erilliset osat ohjelmakoodista kirjoitettiin lähes kokonaan uudelleen. Transienttimallin aikaratkaisumenetelmä perustuu vuon aikaderivaatan implisiittiseen diskretointiin. Noodien sisäinen paikkaratkaisumenetelmä on pienillä muutoksilla yhtenevä tasapainomallin kanssa. Malleihin lisättiin joitain ominaisuuksia, kuten transienttimallin voiden aikaekstrapolaatioon perustuva kiihdytysmenetelmä.</p> <p>Perustuen vertailuihin DIF3D:n kanssa, tasapainomalli on tämän työn jälkeen oikein toteutettu. Kasvukertoimien erot ovat suhteellisen suuria Serpentiin verrattaessa. Tehojakaumien erot ovat hyväksyttävällä tasolla. Reaktorien kokonaistehtojen aikakehitykset simuloimatta termohydrauliikkaa olivat samankaltaisia kuin DYN3D:llä. Askelmaisen säätösauvojen noston simuloinnissa kokonaistehtojen aikakehitysten erot olivat suurempia. Suhteelliset erot eivät kuitenkaan olleet suurempia kuin hitaan säätösauvojen laskun simuloinnissa. Suhteellisten aksiaalitehtojen erot olivat pieniä koodien välillä. Erojen voidaan todeta johtuvan koodien ratkaisumenetelmien eroista. Täten transienttimalli on myös oikein toteutettu ja se tuottaa DYN3D:n kanssa samankaltaisia tuloksia.</p> <p>Aprosiin on nyt toteutettu tasapaino- ja transienttimoniryhmänodaalidiffuusiomallit, ja ne on todettu toimiviksi. Malleille ehdotetaan useita jatkokehityksiä.</p>			
Avainsanat:	Apros, DIF3D, DYN3D, 4. sukupolven reaktorit, nopeat reaktorit, moniryhmä, nodaalidiffuusiomenetelmä, Serpent, SFR		
Kieli:	Englanti		

Acknowledgements

This thesis was written in the Reactor physics team at VTT Technical Research Centre of Finland Ltd in a project partly funded by Fortum Power and Heat Ltd.

First of all, I am most grateful to everybody who contributed towards my opportunity to work with this interesting subject. I would like to thank Markus Airila for acting as my instructor and for his feedback about this work. I also appreciate all the help received from my colleagues while making this thesis.

I wish to thank for all the interesting coffee break conversations about physics, nuclear engineering and everything during my studies, summer jobs and while working towards this thesis. I believe that everyone who have participated in them will recognize themselves.

I also thank my sister Anna for her example which led me to the world of science and physics. The final thanks go to my dear wife Maiju for her constant encouragement and for ensuring it is always nice to come home after a day's work.

Espoo, October 12, 2015

Antti Rintala

Contents

Symbols	vii
1 Introduction	1
1.1 Objectives and scope	3
1.2 Structure of the thesis	3
2 Physical and computational background	4
2.1 Overview of reactor physics	4
2.2 Neutron distribution modeling	6
2.2.1 Neutron transport theory	6
2.2.2 Diffusion theory	8
2.2.3 Monte Carlo method	9
2.3 Calculation path for full core simulations	10
2.4 Computer codes	11
2.4.1 Apros	11
2.4.1.1 Current neutronics models	12
2.4.1.2 Status of the multigroup nodal diffusion model	13
2.4.2 Serpent and Serpent 2	14
2.4.3 DIF3D and DIF3D-K	15
2.4.4 DYN3D	15
3 Computational model description	16
3.1 Starting point of the solution method	16
3.2 Temporal solution method	17
3.3 Spatial solution method	18
3.3.1 Nodal balance equation	20
3.3.2 Spatial boundary conditions	22
3.3.3 Transverse integration	23
3.3.4 Polynomial approximation in hex-plane	26
3.3.5 Polynomial approximation in axial direction	28
3.3.6 Flux moments calculation	30

3.3.7	Approximation of leakage moments	32
3.3.8	Response matrix equation	33
3.4	Computational solution procedure	36
3.4.1	Transient calculation solution algorithm	36
3.4.2	Response matrix solution method	39
3.4.3	Temporal extrapolation	39
3.4.4	Asymptotic extrapolation	40
3.4.5	Steady state calculation solution algorithm	41
3.5	Model implementation into Apros	43
3.5.1	Code structure	43
3.5.2	Reactor model creation	44
3.5.3	Group constant and control assembly descriptions	45
4	Model validation cases	47
4.1	Reactor core models	47
4.2	Group constants	50
4.3	Steady state model implementation	52
4.4	Steady state model accuracy	52
4.5	Transient model	53
5	Results	55
5.1	Steady state model implementation	55
5.2	Steady state model accuracy	58
5.2.1	Medium metallic core	58
5.2.2	Large oxide core	63
5.2.2.1	Different group constant sets	63
5.2.2.2	Additional cases	65
5.2.2.3	Different number of energy groups	74
5.3	Transient model	76
6	Discussion	83
6.1	Validation results	83
6.2	Further research and model development	86
6.3	Summary	89
	Bibliography	94

Symbols

Latin letters

C	precursor density
D	diffusion coefficient
E	energy
G	number of energy groups
J	neutron current, neutron current density
K	number of nodes
L	neutron leakage
M	number of precursor groups
N	angular neutron density
Q	neutron source, source term
S	boundary
T	sum of net currents, temperature
V	volume
f	basis function
h	node flat-to-flat width
k	multiplication factor
n	normal direction
r	position
t	time, total
v	neutron velocity
w	weight function
z	node height

Greek letters

Δ	difference, length
Σ	macroscopic cross section
Ω	solid angle
β	fraction of delayed neutrons

γ	boundary condition coefficient
ε	convergence criterion parameter
λ	precursor decay constant
ν	average number of neutrons released per fission
σ	dominance ratio
ϕ	neutron flux
χ	fission yield
ψ	angular neutron flux, fission source
ω	inverse period

Subscripts

a	absorption
d	delayed
eff	effective
ext	external
f	fission
g	energy group index
i	general index
m	precursor group index
n	expansion order index, time index
p	prompt
r	removal
s	scattering
s	direction
u	u -direction
v	v -direction

Subscripts (cont.)

x	x -direction
xy	hex-plane
y	y -direction
z	axial direction
0	zeroth moment
1	first moment

Superscripts

(i)	outer iteration index
in	incoming
k	node index
l	neighboring node
n	evaluated at time t_n
out	outgoing
+	upper
—	lower

Other

$\bar{\cdot}$	average
$\tilde{\cdot}$	approximate, modified
$\hat{\cdot}$	improved estimate, unit vector in three-dimensional Euclidean space
$\vec{\cdot}$	vector in three-dimensional Euclidean space
$\ \cdot\ _2$	L_2 vector norm
bold	general matrix or vector

Chapter 1

Introduction

Nuclear power can be used to produce electricity with low total lifecycle carbon dioxide emissions [1]. In addition, nuclear power plants usually have high capacity factors and they use only small amount of nuclear fuel compared with the fuel usage of conventional combustion power plants. However, the nuclear power plants of current generations have various issues requiring improvements in the long run. These include the utilization of only a small portion of energy contained in nuclear fuel, limited fuel resources usable with current technology and the production of nuclear waste which stays highly radioactive for thousands of years.

In order to meet the future requirements of sustainable, economical, safe and proliferation resistant nuclear energy production, the Generation IV International Forum have chosen six different next generation reactor types for further research and development. Three of them, namely gas-, lead- and sodium-cooled fast reactors, utilize the fast neutron spectrum unlike the thermal neutron spectrum used by conventional nuclear reactors nowadays. [2]

Fast reactors are of special interest for multiple reasons. Compared with conventional thermal reactors, fast reactors produce more neutrons than are needed to sustain a fission chain reaction. These additional neutrons can be used to transform fertile fuel nuclei into fissile nuclei. If more fissile nuclei are produced than are fissioned in the reactor, this process is called breeding. The excess fissile nuclei can be used in the current generation nuclear power plants after the fuel has been reprocessed. Therefore, fast reactors can generate more usable nuclear fuel than they consume. Alternatively, the excess neutrons can be used to remove minor actinides from spent nuclear fuel reducing the amount of long lived radioactive nuclei in nuclear waste. This process can be used together with fuel reprocessing to dramatically reduce the amount of nuclear waste and the time the waste stays highly radioactive. Due to these alternatives the energy contained in the naturally most abundant isotope of uranium, namely ^{238}U , can be much more thoroughly utilized in fast reactors than in thermal reactors. [3]

The liquid metal coolants used in lead- and sodium-cooled fast reactors provide many interesting features compared with the ordinary water usually utilized as coolant in conventional thermal reactors. One is the operation of the reactors in atmospheric pressures due to the high boiling temperatures of the liquid metal coolants. This eliminates the need for high pressure systems in the reactor cores. In addition, the high thermal conductivities and heat capacities of the coolants allow high power densities in the cores and ensure heat removal from the cores in accident situations. However, there are also drawbacks compared with water as the coolant. For example, both lead and sodium are solids in room temperature. Therefore, they have to be kept heated at all times to avoid equipment damage. In addition, sodium reacts exothermically with air and water, causing it to explode or burn when coming into contact with these substances. [3, 4]

Due to economic and safety reasons, computational analyses of nuclear power plants must be performed instead of experimental studies to analyze their behavior in the normal use and different accident situations. The modeling of neutron transport, or neutronics, in fast reactors can usually not be performed with sufficient accuracy by using computer codes designed for thermal reactors. This is mainly due to the broad neutron energy interval where most of the different interactions occur compared with the narrow thermal energy region which is of the most interest in thermal reactors. In addition, many fast reactor designs have hexagonal fuel assemblies instead of rectangular ones most commonly used in Western thermal nuclear reactors. The geometry has to be taken into account in the derivation of the neutronics solution methodology. Consequently, the analyses of fast reactor cores require computer codes specifically designed for them. Fortunately, due to the typically long neutron mean free paths in the fast reactor cores, a relatively simple and fast neutron diffusion theory based methods, compared with more accurate and computationally demanding transport theory based methods, can be applied to fast reactor simulations with sufficient accuracy.

In addition to the neutronics, also the thermal hydraulics of a nuclear power plant has to be simulated to provide an accurate description of reactor behavior in different transient situations. This is due to the strong coupling between the neutronics and thermal hydraulics. This raises the need for coupled neutronics and thermal hydraulics codes. The differing thermal hydraulic properties of the liquid metal coolants from those of the ordinary water have to be taken into account in the coupled code. If a full scale nuclear power plant is desired to be simulated, all of these neutronics and thermal hydraulics requirements can be fulfilled by utilizing a suitable fast reactor core model in a multipurpose process simulation code.

1.1 Objectives and scope

This thesis continues the development of a multigroup nodal diffusion model for the process simulation software Apros [5–7]. There are two objectives for this thesis. The first is to finalize the steady state and transient simulation capability of the model. The second is to validate the implementation and to determine the applicability and accuracy of both the steady state model and the neutron kinetics part of the transient model for fast reactor simulations. The validation of the coupled neutronics and thermal hydraulics, also known as reactor dynamics, simulation capability of the model is excluded from the scope of this thesis. A wider perspective for both of the objectives is to provide Apros with a fast reactor simulation feature, thus making it possible to simulate liquid metal cooled fast reactor cores with hexagonal fuel assemblies using Apros.

1.2 Structure of the thesis

In Chapter 2, a short introduction to nuclear reactor physics and computational methods is given in addition to descriptions of different computer codes used in this work. In Chapter 3, the temporal and spatial solution methods of the Apros multigroup nodal diffusion model are derived. Moreover, the utilized computational algorithm and methods are presented. In Chapter 4, the different sodium-cooled fast reactor models and simulation cases used to validate the implementation and accuracy the model are specified. In Chapter 5, the results of the validation calculations are presented. Finally, in Chapter 6, the results of the validation and proposed further development and research of the model are discussed, and a summary of this work is presented.

Chapter 2

Physical and computational background

A nuclear power plant consists of various systems, such as the reactor core, where heat is produced and the thermal hydraulic system, which transfers the heat out of the core. In order to design and safely operate these complex systems, sufficiently accurate models of them have to be available. However, these models have to be computationally solvable in a reasonable time, especially considering time-dependent transient problems which are typically computationally much more demanding than time-independent steady state problems.

In this chapter, first a short introduction to reactor physics is given. Second, the most common models for describing the neutron distribution inside a nuclear reactor core are presented. Third, the calculation path from theoretical nuclear data to full core simulations is described. Finally, the computer codes used and developed in this work are presented.

2.1 Overview of reactor physics

Neutrons interact with the medium they are moving in in various ways, such as by absorbing into a nucleus in a capture reaction, causing a fission reaction in a heavy nucleus or scattering from a nucleus. The probabilities for these interactions to happen are characterized with microscopic cross sections, denoted with σ . The microscopic cross sections are heavily dependent on the relative kinetic energy between the incident neutron and the target nucleus and on the type of the nucleus. The cross section dependence on the kinetic energy of the nuclei are typically taken into account with the temperature of the medium. Two examples of the energy dependence are shown in Fig. 2.1, showing the fission cross section of ^{235}U and the radiative capture cross section of ^{238}U . In a classical sense, the microscopic

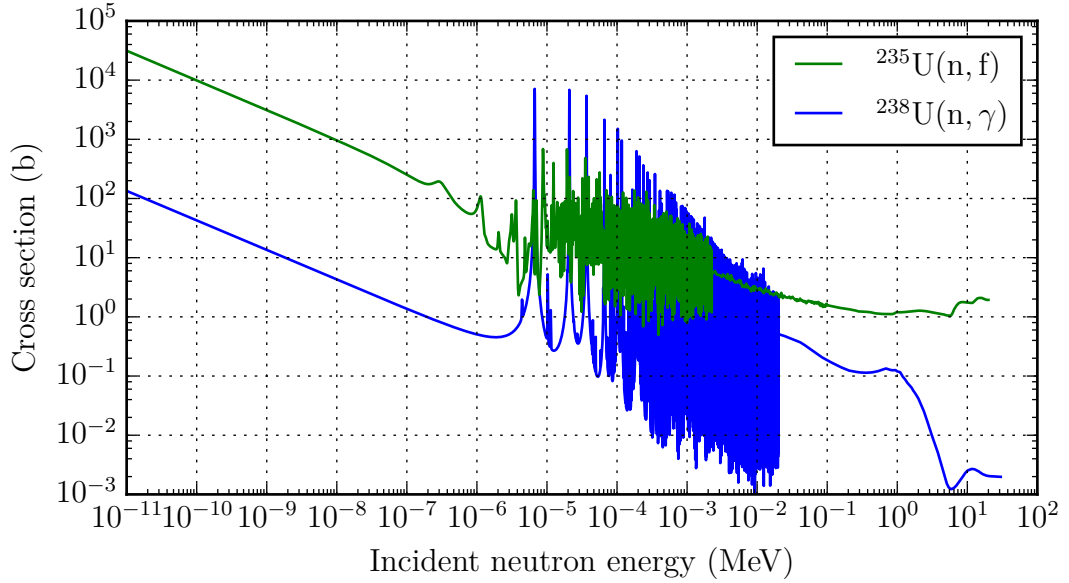


Figure 2.1: Fission cross section of ^{235}U and radiative capture cross section of ^{238}U . [8]

cross section can be interpreted as the effective cross-sectional area of a nucleus. A common unit of the microscopic cross sections is barn, which is equal to 10^{-24} cm^2 . Another common quantity is the macroscopic cross section Σ which is defined as the cross section multiplied by the atomic density of the target nuclei. The macroscopic cross section can be interpreted physically as the interaction probability per unit path length of a neutron.

In a fission reaction, a heavy nuclide splits typically into two lighter daughter nuclei releasing energy and a few neutrons. Subsequently, some of these neutrons may cause new fission reactions, maintaining a fission chain reaction. One of the main parameters characterizing the chain reaction is the multiplication factor k . It is defined as the average number of neutrons born from a fission causing another fissions. The nuclear reactor states $k > 1$, $k = 1$ and $k < 1$ are referred to as supercriticality, criticality and subcriticality, respectively. If a reactor is supercritical, the fission rate inside the reactor is growing and therefore the reactor power is also increasing. If the reactor is critical, the fission rate and reactor power stay constants. If the reactor is subcritical, the fission rate and therefore the reactor power are decreasing.

However, not all the fission neutrons are born immediately during the fission reaction. A fraction β of fission neutrons are produced in the neutron emissions of the high-energetic β -decay products of the fission daughter nuclei. These neutrons are called delayed neutrons to differentiate them from the prompt neutrons

born during the fission reaction. The neutron-emitting fission product nuclei are called delayed neutron precursors. The precursors are typically divided into 6 or 8 precursor groups based on their β -decay half lives. Although β is very small as its effective value is typically less than 1 % in fast reactors [3], the effect of the delayed neutrons on the time-dependent behavior of a nuclear reactor is significant. Without them, even a slight increase in the multiplication factor would lead to an unstable exponential growth of the neutron density and thus the reactor power, if the various feedback mechanisms present in the nuclear reactor were not effective enough.

2.2 Neutron distribution modeling

The neutrons interact with the materials inside a nuclear reactor in various ways and the interaction probabilities are highly dependent on the neutron energies. Therefore, the distribution of the neutrons inside space and energy, and in transient problems also in time, inside a nuclear reactor is required to characterize the reactor. As the determination of the neutron distribution is an important and demanding task, various methods have been developed for it.

2.2.1 Neutron transport theory

Neutron transport theory is the most commonly applied model for defining the neutron distribution. It describes the distribution of neutrons inside a phase space defined by three-dimensional spacial coordinates \vec{r} , angle $\hat{\Omega}$ and energy E at a specific time t . This distribution is called the angular neutron density N , which can be physically interpreted as $N(\vec{r}, \hat{\Omega}, E, t) dV d\hat{\Omega} dE$ being the number of neutrons in a differential phase space element $dV d\hat{\Omega} dE$ about $(\vec{r}, \hat{\Omega}, E)$ at time t . [9]

Defining the angular neutron flux ψ with

$$\psi(\vec{r}, \hat{\Omega}, E, t) = v(E)N(\vec{r}, \hat{\Omega}, E, t), \quad (2.1)$$

where $v(E)$ is the neutron velocity, the neutron distribution can be described with the time-dependent neutron transport equation

$$\begin{aligned} \frac{1}{v(E)} \frac{\partial \psi(\vec{r}, \hat{\Omega}, E, t)}{\partial t} + \hat{\Omega} \cdot \nabla \psi(\vec{r}, \hat{\Omega}, E, t) + \Sigma_t(\vec{r}, E) \psi(\vec{r}, \hat{\Omega}, E, t) = \\ \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \Sigma_s(\vec{r}, \hat{\Omega}' \cdot \hat{\Omega}, E' \rightarrow E) \psi(\vec{r}, \hat{\Omega}', E', t) + \\ \frac{\chi_p(\vec{r}, E)}{4\pi} \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' (1 - \beta(\vec{r}, E')) \nu \Sigma_f(\vec{r}, E') \psi(\vec{r}, \hat{\Omega}', E', t) + \\ \frac{1}{4\pi} \sum_{m=1}^M \chi_{d,m}(\vec{r}, E) \lambda_m(\vec{r}) C_m(\vec{r}, t) + \frac{1}{4\pi} Q_{\text{ext}}(\vec{r}, E, t), \end{aligned} \quad (2.2)$$

2.2. NEUTRON DISTRIBUTION MODELING

which can be derived using physical conservation principles in a differential phase space element. Here Σ_t and $\nu\Sigma_f$ are the macroscopic total and neutron production cross sections, respectively, Σ_s is the macroscopic double differential scattering cross section, χ_p and χ_d are the prompt and delayed fission yields, respectively, M is the number of precursor groups, λ_m and C_m are the precursor decay constant and precursor density for the precursor group m and Q_{ext} is the external neutron source. In addition, the total delayed neutron fraction is defined as

$$\beta(\vec{r}, E') = \sum_{m=1}^M \beta_m(\vec{r}, E'). \quad (2.3)$$

The precursor densities are governed by

$$\frac{\partial C_m(\vec{r}, t)}{\partial t} = \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \beta_m(\vec{r}, E') \nu \Sigma_f(\vec{r}, E') \psi(\vec{r}, \hat{\Omega}', E', t) - \lambda_m(\vec{r}) C_m(\vec{r}, t), \quad (2.4)$$

which can also be derived as the decay and production balance of immobile delayed neutron precursors in a differential phase space element. Neutron multiplication reactions, such as (n, 2n) reactions are not taken into account here. In addition, the external source is assumed to be isotropic.

If a time-independent problem is studied, then the time derivative of the angular flux and β_m and C_m are set to zero, and the fission source term is divided by the effective multiplication factor k_{eff} to obtain a neutron balance. The result is the steady state transport equation formulated as a k -eigenvalue problem.

Eqs. (2.2) and (2.4) should be solved together with appropriate initial conditions for the angular flux and precursor densities, and boundary condition for the angular flux. The solution would be an almost exact representation of the neutron distribution, as only small assumptions are made. These assumptions include treating neutrons as pointlike particles traveling straight paths between interactions, without any neutron-neutron interactions. The cross sections are assumed to be independent of the angular flux and constant in time. The scattering cross section is assumed to depend only on the scattering angle and not on the original direction of the neutron movement. Moreover, the fission neutrons are assumed to be emitted isotropically and the fission yields are assumed to be independent of the energy of the neutron causing the fission.

If a time-dependent problem is studied, the requirement of the time-independence of the cross sections might seem to be problematic, as the cross sections change for example due to changes of nuclear fuel temperatures. However, this problem is usually circumvented by treating the cross sections as constants during sufficiently small time steps.

Due to the complex nature of the transport equation, it is practically impossible to be solved analytically in any real world geometries. Therefore, approximations

and assumptions, especially for the angular variable, are needed to solve it. They lead to different deterministic transport methods, such as to the spherical harmonics method or the method of characteristics. In addition, all the possible neutron energies are usually divided into discrete groups, where for example the cross sections are treated as constants. [10]

2.2.2 Diffusion theory

The most common method used to solve the transport equation is to use the diffusion theory. During the derivation of the theory, Eq. (2.2) is integrated over the angular variable. This introduces new quantities, namely the scalar neutron flux defined as

$$\phi(\vec{r}, E, t) = \int_{4\pi} d\hat{\Omega} \psi(\vec{r}, \hat{\Omega}, E, t) \quad (2.5)$$

and neutron current density defined as

$$\vec{J}(\vec{r}, E, t) = \int_{4\pi} d\hat{\Omega} \hat{\Omega} \psi(\vec{r}, \hat{\Omega}, E, t) \quad (2.6)$$

to the resulting equation. In addition, the neutron energy dependence is divided into G energy groups by integrating the resulting equation over energy from E_{g-1} to E_g . The group $g = 1$ is the group with the highest energy, whereas the group $g = G$ is the group with the lowest energy. The neutron current density is approximated using the diffusion approximation, or Fick's law, which states that neutron current density is directly proportional to the gradient of the scalar neutron flux as

$$\vec{J}_g(\vec{r}, t) = -D_g(\vec{r}) \nabla \phi_g(\vec{r}, t), \quad (2.7)$$

where D_g is the diffusion coefficient, ϕ_g is the scalar neutron flux and g denotes the energy group. A more thorough derivation of the diffusion theory can be found for example in Ref. [11].

With this procedure, the time-dependent multigroup neutron diffusion equation can be written as

$$\begin{aligned} \frac{1}{v_g(\vec{r})} \frac{\partial \phi_g(\vec{r}, t)}{\partial t} - \nabla \cdot D_g(\vec{r}) \nabla \phi_g(\vec{r}, t) + \Sigma_{r,g}(\vec{r}) \phi_g(\vec{r}, t) = \\ \sum_{g' \neq g} \Sigma_{s,g' \rightarrow g}(\vec{r}) \phi_{g'}(\vec{r}, t) + (1 - \beta(\vec{r})) \chi_{p,g}(\vec{r}) \sum_{g'=1}^G \nu \Sigma_{f,g'}(\vec{r}) \phi_{g'}(\vec{r}, t) + \\ \chi_{d,g}(\vec{r}) \sum_{m=1}^M \lambda_m(\vec{r}) C_m(\vec{r}, t) + Q_{\text{ext}}(\vec{r}, t), \end{aligned} \quad (2.8)$$

where v_g is the neutron velocity, $\Sigma_{r,g}$ and $\Sigma_{s,g' \rightarrow g}$ are the macroscopic group removal and from group g' to group g transfer cross sections, respectively, and $\chi_{p,g}$ and $\chi_{d,g}$

are the prompt and delayed fission yields, respectively. The precursor densities are governed by

$$\frac{\partial C_m(\vec{r}, t)}{\partial t} = \beta_m(\vec{r}) \sum_{g=1}^G \nu \Sigma_{f,g}(\vec{r}) \phi_g(\vec{r}, t) - \lambda_m(\vec{r}) C_m(\vec{r}, t). \quad (2.9)$$

The Eqs. (2.8) and (2.9) should be solved together with appropriate initial conditions for the scalar flux and precursor densities, and boundary condition for the scalar flux. If a time-independent problem is studied, the time derivative of the scalar flux and β_m and C_m are set to zero, and the fission source term is divided by the effective multiplication factor k_{eff} to obtain a neutron balance. The result is a steady state multigroup diffusion equation formulated as a k -eigenvalue problem.

The validity of the diffusion theory requires a few assumptions. First, neutron scattering has to be isotropic and the dominant interaction type over absorption. Second the spatial neutron distribution has to vary only linearly. Third, all the neutron sources have to be isotropic. Finally, the time derivative of the neutron current density divided by the neutron velocity has to be small compared with the flux gradient. These assumptions do not hold for example near highly absorbing materials, such as control rods, or near vacuum boundaries. Nevertheless, the diffusion theory is widely used in reactor analysis due to its simplicity and computational efficiency compared with the more complicated transport methods. [11, 12]

2.2.3 Monte Carlo method

A completely different approach to describe the neutron distribution inside a nuclear reactor is to perform a Monte Carlo reactor physics simulation. Instead of trying to solve the transport or diffusion equations, individual neutron histories are simulated to calculate statistical estimates for different integral quantities, such as the multiplication factor, neutron flux and reaction rates. A neutron history consists of the interactions the neutron undergoes and the path the neutron travels between the interactions. It begins from the emission of the neutron and ends to the absorption of the neutron or when the neutron leaks permanently out of the problem geometry. [13]

The power of the reactor physics Monte Carlo simulations lies in the fact that the method itself does not require approximations in the neutron interaction physics or the studied problem geometry. However, due to the statistical nature of the method, the accuracy of the results is proportional to the number of neutron histories simulated. Therefore, compared with the deterministic methods, the method is computationally extremely demanding. Obtaining completely accurate results would require the simulation of infinitely many neutron histories.

Due to the computational inefficiency, coupled full core Monte Carlo neutronics and thermal hydraulics simulations are still too demanding to be solved. However,

time-independent full core simulations for example to be used as reference calculations for different deterministic methods are within the limitations of computational resources nowadays. In addition, Monte Carlo simulations can be used to generate homogenized group constants for deterministic full core simulation codes.

2.3 Calculation path for full core simulations

The path to calculate a deterministic full core simulation requires many steps, where the best available knowledge of neutron interactions is transferred into a format usable by the simulation code. This path from the nuclear data into deterministic full core simulation is presented in Fig. 2.2.

The neutron interaction data obtained from experimental measurements and theoretical nuclear models are collected into evaluated nuclear data file libraries.

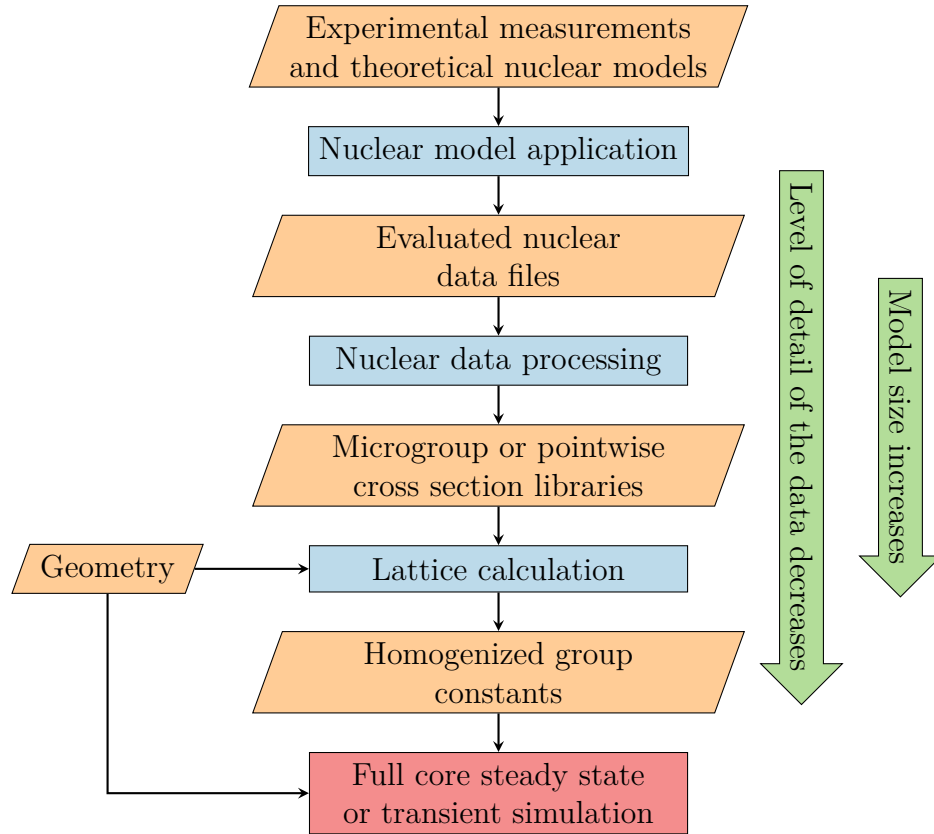


Figure 2.2: Typical nuclear reactor analysis calculation steps for a deterministic full core simulation.

The libraries need to be processed into either discrete energy groupwise form to be used in deterministic lattice calculation codes or continuous energy pointwise form to be used in continuous-energy Monte Carlo lattice calculation codes. Whereas the evaluated nuclear data file library format is designed to preserve the accuracy of the original data, the microgroup condensation causes loss of accuracy and generality, and even the continuous-energy pointwise library is generated with only a certain accuracy.

During a lattice calculation, the accurate fuel assembly geometries are used to produce homogenized group constants. The group constants are calculated so that the reaction rates over the homogenized volume are preserved in the full core simulations utilizing the generated group constants. The homogenization is carried out for each fuel assembly or material type in different local operating conditions, such as with different coolant and fuel temperatures and fuel burnups. The homogenized group constants are generated for a small number of energy groups. For thermal reactors, usually two energy groups are used, but for fast reactors, more groups are required to model the reactors with sufficient accuracy. For time-dependent simulations, in addition to the diffusion coefficients used in diffusion calculation codes and the macroscopic cross sections used in transport and diffusion calculation codes, the group constants include kinetics parameters such as the effective fractions of delayed neutrons and the neutron velocities.

The last step in the path is to perform a full core steady state or transient simulation to determine for example the power distribution inside a reactor. During a full core simulation, the reactor geometry is treated as consisting of homogeneous nodes, where the homogenized group constants produced by the lattice code are utilized.

2.4 Computer codes

2.4.1 Apros

Apros is a process simulation software developed by VTT Technical Research Centre of Finland Ltd and Fortum Power Solutions [7]. The software can be used for full scale modeling and dynamic simulations of different industrial processes, such as paper mills and combustion and nuclear power plants. In addition to the thermal hydraulics, the simulation capabilities of the software extend to complex systems including process automation, electrical systems and nuclear reactor neutronics.

An Apros model is built by connecting suitable process components together and configuring their parameters. The components include thermal hydraulic components such as pipes, pumps and valves, ordinary nuclear reactor components such as pressurizers and steam generators, and automation and electrical system

components. The models can be built directly using a command window, or more typically a graphical user interface. Grades is used as the graphical user interface in Apros version 5 and a new, more flexible graphical user interface is used with Apros version 6.

The thermal hydraulics of a nuclear power plant can be modeled with three different models. The difference between the models is the treatment of the mass, momentum and energy of the liquid and gas phases as a mixture or separately. The most accurate method, namely the six-equation model, is computationally the most demanding. The five- and three-equation models are less accurate and computationally less demanding, in this order. Due to accuracy considerations, the six-equation model is recommended to be used to simulate the thermal hydraulics inside reactor cores of conventional thermal reactors. In the case of a single phase flow, the three-equation homogeneous model can also be used. The sodium present in sodium fast reactors is simulated with the three-equation model. [14, 15]

2.4.1.1 Current neutronics models

The current release version of Apros has four different neutronics models, namely the point kinetics model, the one- and three-dimensional finite difference models and the three-dimensional nodal model [14, 16–18]. The simulation accuracy of the models increases together with the computational cost from the point kinetics model to the nodal model. All the models are capable of performing steady state and transient simulations.

The point kinetics model uses a single one-group flux value to simulate the whole reactor. Therefore its applicability is very limited. The point kinetics model has features common to the more accurate models, such as six delayed neutron precursor groups and the simulation of reactor poisoning and decay heat.

The one-dimensional finite difference model simulates the reactor behavior only in the axial direction. Therefore it is applicable to simulations where the reactor behavior has only a small impact compared with other effects in the simulation, or the reactor transients are symmetrical in the radial direction. In addition, the condensation of the usual group constants into radially homogenized group constants to be used with the one-dimensional model requires an additional calculation step to those described in Sec. 2.3.

The three-dimensional finite difference model can be used to model various transients without limitations in their symmetries. However, the accuracy of this one-point per node model is not high enough for some demanding simulation cases. Therefore, a more accurate but still relatively fast three-dimensional nodal model based on modal decomposition is included in Apros.

All of these models are developed to simulate conventional thermal reactors, as for example they are based on one- or two-group approximations. Therefore, they

are not suitable for simulating fast reactors accurately. To extend the simulation capabilities to fast reactors, a three-dimensional multigroup nodal model has been developed for Apros.

2.4.1.2 Status of the multigroup nodal diffusion model

A steady state multigroup nodal diffusion model has been implemented into Apros in 2009 [5]. The solution method is based on the nodal diffusion option of the DIF3D simulation package, described in Sec. 2.4.3. This method was chosen due to the applicability of diffusion theory in fast reactor problems, its capability of handling hexagonal geometries and its fast calculation speed suitable for transient analyses. Compared to the thermal reactor neutronics models of Apros, some features, such as the simulation of reactor poisoning and the iteration of equilibrium boron concentration are not present in the multigroup model, as they are not important in the case of fast reactors.

Since the implementation, the model has completely been rewritten using Fortran 2003 programming language standard with object-oriented programming like approach using derived data types introduced in Fortran 90 standard. In addition, the development of a transient version, based on the solution method of the DIF3D-K simulation code, described in Sec. 2.4.3, has been started [6]. However, no comprehensive validation results of the steady state model exist, the transient model has not been compared with other simulation codes and the model has not been included in the release version of Apros.

The steady state model has been tested by simulating the Large Core Code Evaluation Working Group (LCCEWG) benchmark problem 1 [5, 19]. The results were compared with the other benchmark participants and Serpent (see Sec. 2.4.2) and found satisfactory. The model has also been tested with the OECD/NEA Sodium-cooled fast reactor (SFR) Benchmark Task Force steady state benchmarks [20] and one Phénix end-of-life test [6, 21]. However, no complete results of these simulations exist, as only the obtained multiplication factors are known from the SFR benchmark problems [22, 23]. The differences of the multiplication factors of the four reactor cores, simulated with all control rods withdrawn from the cores, were from -0.52% to 0.20% compared with Serpent full core calculations. The Apros model calculations were performed with 20 energy groups and the group constants were obtained from the Serpent full core calculations. However, the diffusion coefficients used in these calculations have probably been erroneous due to the incorrect diffusion coefficient calculation method in Serpent at that time [24]. Therefore, these multiplication factors are not comparable to those calculated in this work.

Ref. [6] also states that the differences of multiplication factors between DIF3D and the steady state model, being around 10^{-8} , are negligible. This suggests that

the steady state model is correctly implemented. The transient model has not been tested against DIF3D-K as the original code has not been available.

Finally, Ref. [6] states that the model is ready to be used to simulate time-dependent problems. Furthermore, a flux normalization performed after each time step should possibly be revised, and the implementation of the model could be enhanced with for example enabling its parameters to be customized in Apros and utilizing more parameters directly from Serpent group constant generation calculations. However, during this work, several defects and errors, some of which fatal, were found in the implementation of the model.

In the steady state model, there was for example an error in the handling of the reactor boundaries which prevented simulation cases with specific geometries and boundary conditions to converge, even though some such cases were claimed to be simulated [6]. In addition, the flux was not properly normalized to the user given power. In the transient model, there was an error in the subroutine reading the group constant file which prevented all transient simulations with group constant files produced with the supplied processing script. Some terms were missing from the spatial solution equations and the computational algorithm was erroneous, as it included many incorrect parts of the steady state algorithm and many parts essential to the time-dependent solution were flawed. For example, the multiplication factor was updated during transient simulations and the precursor densities were not calculated properly. There were also smaller errors common to both models, such as indexing errors in the connections between the neutronics and thermal hydraulics. Finding the errors preventing the simulations altogether and correcting the defects in the calculation algorithms formed a major part of this work. The parts of the transient solution algorithm not shared with the steady state algorithm were derived and almost completely rewritten during this work. In addition, in this work a transient calculation acceleration method was implemented, the usage of group constants produced by Serpent was improved and the modeling of finite length control rods was added to the model.

2.4.2 Serpent and Serpent 2

Serpent is a three-dimensional continuous-energy Monte Carlo reactor physics burnup calculation code developed at VTT Technical Research Centre of Finland Ltd [25]. In addition to the current release version of Serpent, a completely rewritten Serpent 2 is in a beta-testing phase. The current development of the code is focused on Serpent 2, which has significantly improved functionalities compared with Serpent. Serpent 2 is used in this work, and is referred to as Serpent from now on.

One of the main uses of Serpent is the group constant production for deterministic reactor simulator codes. The group constants include infinite lattice and

B_1 leakage corrected cross sections and diffusion coefficients, kinetics parameters such as effective delayed neutron fractions and precursor decay constants, and assembly discontinuity factors. An automated burnup sequence for creating group constants for all operating conditions of a reactor, parametrized by different branch and history variables, is being developed. [26, 27]

Serpent can also be used for steady state modeling of full reactor cores in three dimensions without the approximations of the deterministic codes. However, due to the high computational costs of Monte Carlo simulations, it is feasible to cover only a small number of different reactor states with the simulations.

2.4.3 DIF3D and DIF3D-K

DIF3D is a code system for solving multigroup steady state neutron diffusion and transport equations. It provides three flux solution methods, namely the finite difference and nodal diffusion solvers and the variational nodal transport solver. The solvers can be used in two- or three-dimensional Cartesian or hexagonal geometries. The code system is primarily designed for fast reactor problems. The hexagonal geometry nodal diffusion solver, used in this work, is described in detail in Ref. [28].

The steady state nodal diffusion solver has been extended to neutron kinetics code DIF3D-K, solving the time-dependent diffusion equation [29]. The time discretization is performed with either the theta method or one of the three space-time factorization methods [30]. The code has also been coupled to SAS thermal hydraulics code family [31, 32].

2.4.4 DYN3D

DYN3D is a three-dimensional reactor simulator code capable of performing steady state and transient diffusion calculations in Cartesian or hexagonal geometry [33, 34]. The code has originally been developed for two-group thermal reactor simulations, but a more recent multigroup version of DYN3D is used in this work.

The code uses transverse integration to solve the neutron diffusion equation inside the nodes. The obtained equations are solved using flux expansions consisting of quadratic polynomials and exponential functions. In the hexagonal plane, either the side averaged fluxes (HEXNEM1 method) or the side averaged and corner point fluxes (HEXNEM2 method) can be used to couple adjacent nodes. The time derivative of the time-dependent diffusion equation is discretized using an implicit difference with an exponential transformation. The code has an internal thermal hydraulics solver and it has also been coupled to thermal hydraulics system codes ATHLET and RELAP5.

Chapter 3

Computational model description

The previous chapter provided an introduction to reactor physics and different neutron distribution modeling methods on a general level. In this chapter, the derivation of the solution algorithm of the Apros multigroup nodal diffusion model is presented with more detail. The temporal solution is based on the fully implicit theta solution method of DIF3D-K [29], whereas the spatial solution method is the same as of the hexagonal geometry nodal diffusion solver of DIF3D [28].

In this chapter, first the temporal solution method of the time-dependent multigroup neutron diffusion equation is derived. Second, the spatial solution method for the hexagonal geometry is derived. Third, the actual computational solution procedure including the different acceleration methods used to speed up the convergence of the solution is presented. In addition, the steady state solution procedure is also briefly introduced. Finally, the implementation of the model into Apros is described.

3.1 Starting point of the solution method

Beginning from Eq. (2.8), the time-dependent multigroup diffusion equation in a homogeneous node k can be written as

$$\begin{aligned} \frac{1}{v_g^k} \frac{\partial \phi_g^k(\vec{r}, t)}{\partial t} - \nabla \cdot D_g^k \nabla \phi_g^k(\vec{r}, t) + \Sigma_{r,g}^k \phi_g^k(\vec{r}, t) &= \sum_{g'=1}^{g-1} \Sigma_{s,g' \rightarrow g}^k \phi_{g'}^k(\vec{r}, t) \\ &+ (1 - \beta^k) \chi_{p,g}^k \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^k(\vec{r}, t) + \chi_{d,g}^k \sum_{m=1}^M \lambda_m^k C_m^k(\vec{r}, t), \end{aligned} \quad (3.1)$$

where $\vec{r} \in V^k$, where V^k is the volume of the node. A few assumptions have been made here. First, the node is assumed to be homogeneous and therefore all the group constants are assumed to be constants inside the node. Second, only down-scattering is considered. Finally, external sources are not considered here. With

similar assumptions, the time dependence of the precursor density for precursor group m inside the node k is governed by

$$\frac{\partial C_m^k(\vec{r}, t)}{\partial t} = \beta_m^k \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^k(\vec{r}, t) - \lambda_m^k C_m^k(\vec{r}, t), \quad (3.2)$$

obtained from Eq. (2.9).

3.2 Temporal solution method

In the nodal diffusion solver of DIF3D-K, either the theta time-discretization method or one of the three different space-time factorization methods can be used as the temporal solution method. The space-time factorization methods assume that the shape of the neutron flux changes much slower than the amplitude of the flux, whereas the theta methods are based on the discretization of the flux time derivative term. The most accurate of the space-time factorization methods, namely the improved quasistatic method has been found to be computationally faster than the fully implicit theta method with comparable accuracy [30]. However, in transients with significant change of the flux shape, the improved quasistatic method might be computationally slower than the implicit theta method [31]. Nevertheless, the fully implicit time differentiation has been chosen to be the temporal solution method of the Apros model, as it is simple to implement, versatile in different transient situations and inherently stable with different time step lengths.

In order to solve Eq. (3.1), the time domain is discretized into discrete times $\{t_n\}$ and the time derivative of the neutron flux is approximated using backward difference at time t_{n+1} as

$$\left. \frac{\partial \phi_g^k(\vec{r}, t)}{\partial t} \right|_{t_{n+1}} = \frac{\phi_g^k(\vec{r}, t_{n+1}) - \phi_g^k(\vec{r}, t_n)}{\Delta t_n}, \quad (3.3)$$

where $\Delta t_n = t_{n+1} - t_n$ is the time step length. Consistently with the backward difference, Eq. (3.2) can be solved assuming the fission source to vary linearly during the time step as

$$\nu \Sigma_{f,g} \phi_g^k(\vec{r}, t) = \nu \Sigma_{f,g} \phi_g^k(\vec{r}, t_n) + \frac{\nu \Sigma_{f,g} \phi_g^k(\vec{r}, t_{n+1}) - \nu \Sigma_{f,g} \phi_g^k(\vec{r}, t_n)}{\Delta t_n} (t - t_n). \quad (3.4)$$

The resulting equation can be written as

$$C_m^k(\vec{r}, t_{n+1}) = C_m^k(\vec{r}, t_n) e^{-\lambda_m^k \Delta t_n} + F_{0,m}^{k,n} \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^k(\vec{r}, t_n) + F_{1,m}^{k,n} \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^k(\vec{r}, t_{n+1}), \quad (3.5)$$

where the coefficients are defined as

$$F_{0,m}^{k,n} = \frac{\beta_m^k}{\lambda_m^k} \left(\frac{1}{\lambda_m^k \Delta t_n} (1 - e^{-\lambda_m^k \Delta t_n}) - e^{-\lambda_m^k \Delta t_n} \right) \quad (3.6a)$$

$$F_{1,m}^{k,n} = \frac{\beta_m^k}{\lambda_m^k} \left(1 - \frac{1}{\lambda_m^k \Delta t_n} (1 - e^{-\lambda_m^k \Delta t_n}) \right). \quad (3.6b)$$

As Δt_n approaches zero, both of these coefficients approach zero as $\beta_m^k \Delta t_n$. Therefore, with short time steps they depend almost linearly on the time step length.

Substituting Eqs. (3.3) and (3.5) into Eq. (3.1), the following equation for the neutron flux at time t_{n+1} in node k and energy group g is obtained

$$\begin{aligned} -\nabla \cdot D_g^k \nabla \phi_g^k(\vec{r}, t_{n+1}) + \left(\frac{1}{v_g^k \Delta t_n} + \Sigma_{r,g}^k \right) \phi_g^k(\vec{r}, t_{n+1}) = \\ \sum_{g'=1}^{g-1} \Sigma_{s,g' \rightarrow g}^k \phi_{g'}^k(\vec{r}, t_{n+1}) + \\ \left((1 - \beta^k) \chi_{p,g}^k + \chi_{d,g}^k \sum_{m=1}^M \lambda_m^k F_{1,m}^{k,n} \right) \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^k(\vec{r}, t_{n+1}) + \\ \frac{1}{v_g^k \Delta t_n} \phi_g^k(\vec{r}, t_n) + \\ \chi_{d,g}^k \sum_{m=1}^M \left(\lambda_m^k C_m^k(\vec{r}, t_n) e^{-\lambda_m^k \Delta t_n} + \lambda_m^k F_{0,m}^{k,n} \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^k(\vec{r}, t_n) \right). \end{aligned} \quad (3.7)$$

The resulting equation can be identified to be the same form as a fixed source problem equation. Part of the source term depends only on the fission source at the end of the time step and part of it depends on the fission source and precursor densities at the beginning of the time step. Therefore, the flux at time t_{n+1} can be solved using any solver capable of solving fixed source problems if the neutron flux and precursor densities are known at the beginning of the time step.

3.3 Spatial solution method

The volume of a hexagonal prism node k with width h and height Δz^k can be defined as

$$V^k : (x, y, z), \quad x \in \left[-\frac{h}{2}, \frac{h}{2} \right], \quad y \in [-y_s(x), y_s(x)], \quad z \in \left[-\frac{\Delta z^k}{2}, \frac{\Delta z^k}{2} \right], \quad (3.8)$$

where

$$y_s(x) = \frac{1}{\sqrt{3}}(h - |x|). \quad (3.9)$$

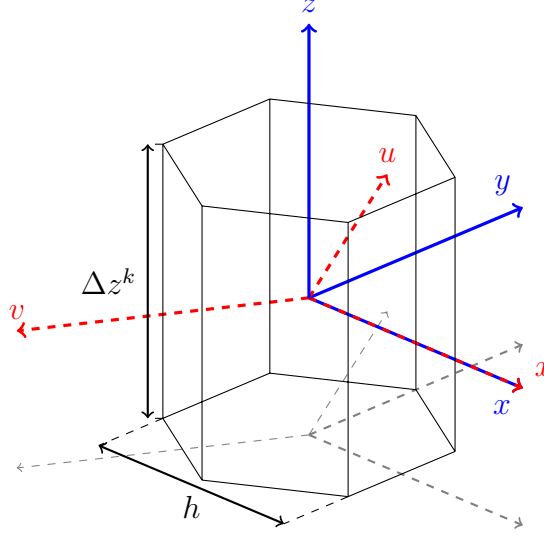


Figure 3.1: Cartesian (x, y, z) and symmetric hex-plane (x, u, v) coordinate systems in node k with width h and height Δz^k .

The node along with the Cartesian (x, y, z) and symmetric hex-plane (x, u, v) coordinate systems used in the derivation is presented in Fig. 3.1. The total volume of the node can be calculated by integrating over the node as

$$V^k = \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy = \frac{\sqrt{3}}{2} h^2 \Delta z^k. \quad (3.10)$$

Denoting quantities defined at time t_n with superscript n , Eq. (3.7) can be written as

$$-\nabla \cdot D_g^k \nabla \phi_g^{k,n+1}(\vec{r}) + \tilde{\Sigma}_{r,g}^{k,n+1} \phi_g^{k,n+1}(\vec{r}) = Q_g^{k,n+1}(\vec{r}), \quad (3.11)$$

where

$$\tilde{\Sigma}_{r,g}^{k,n+1} = \frac{1}{v_g^k \Delta t_n} + \Sigma_{r,g}^k \quad (3.12)$$

is the modified group removal cross section and

$$\begin{aligned} Q_g^{k,n+1}(\vec{r}) = & \left((1 - \beta^k) \chi_{p,g}^k + \chi_{d,g}^k \sum_{m=1}^M \lambda_m^k F_{1,m}^{k,n} \right) \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^{k,n+1}(\vec{r}) + \\ & \sum_{g'=1}^{g-1} \Sigma_{s,g' \rightarrow g}^k \phi_{g'}^{k,n+1}(\vec{r}) + \frac{1}{v_g^k \Delta t_n} \phi_g^{k,n}(\vec{r}) + \\ & \chi_{d,g}^k \sum_{m=1}^M \left(\lambda_m^k C_m^{k,n}(\vec{r}) e^{-\lambda_m^k \Delta t_n} + \lambda_m^k F_{0,m}^n \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \phi_{g'}^{k,n}(\vec{r}) \right) \end{aligned} \quad (3.13)$$

is the source term of the time step n . From now on, the explicit time dependence will be suppressed, as all quantities in Eq. (3.11) are determined at the same time.

The spatial solution method is derived in detail in Refs. [5, 28]. The differences between the derivations here and in Refs. [5, 28] are the replacement of the group removal cross section with the modified group removal cross section and the source term, which now includes the contribution of delayed neutrons and the part of the flux time derivative depending on the flux at the beginning of the time step. In addition, the prompt fission source is not normalized by the multiplication factor, as now a time dependent problem is studied.

3.3.1 Nodal balance equation

Integrating Eq. (3.11) over the node volume, the nodal balance equation

$$\frac{2}{3h}(\bar{L}_{gx}^k + \bar{L}_{gu}^k + \bar{L}_{gv}^k) + \frac{1}{\Delta z^k} \bar{L}_{gz}^k + \tilde{\Sigma}_{r,g}^k \bar{\phi}_g^k = \bar{Q}_g^k \quad (3.14)$$

is obtained. Here the average flux is defined as

$$\bar{\phi}_g^k = \frac{1}{V^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \phi_g^k(x, y, z) \quad (3.15)$$

and the average source as

$$\bar{Q}_g^k = \frac{1}{V^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy Q_g^k(x, y, z). \quad (3.16)$$

The leakage term in the x -direction in the hex-plane is

$$\bar{L}_{gx}^k = \bar{J}_{gx}^k \left(\frac{h}{2} \right) - \bar{J}_{gx}^k \left(-\frac{h}{2} \right), \quad (3.17)$$

where the face-averaged surface-normal components of the net currents are defined as

$$\bar{J}_{gx}^k \left(\pm \frac{h}{2} \right) = \frac{1}{\Delta z^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \frac{1}{2y_s(x)} \int_{-y_s(x)}^{y_s(x)} dy \left(-D_g^k \frac{\partial}{\partial x} \phi_g^k(x, y, z) \right) \Bigg|_{x=\pm \frac{h}{2}}. \quad (3.18)$$

The integration of the first term in Eq. (3.11) over the node volume is transformed into integration over the node boundaries using Gauss' theorem. Similar definitions for \bar{L}_{gu}^k and \bar{L}_{gv}^k can be derived. [28]

Consistently with the hex-plane directions, the leakage term in the axial direction is defined as

$$\bar{L}_{gz}^k = \bar{J}_{gz}^k \left(\frac{\Delta z^k}{2} \right) - \bar{J}_{gz}^k \left(-\frac{\Delta z^k}{2} \right), \quad (3.19)$$

where the face-averaged surface-normal components of the net current are defined as

$$\bar{J}_{gz}^k\left(\pm\frac{\Delta z^k}{2}\right) = \frac{\Delta z^k}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \left(-D_g^k \frac{\partial}{\partial z} \phi_g^k(x, y, z) \right) \Big|_{z=\pm\frac{\Delta z^k}{2}}. \quad (3.20)$$

The face-averaged surface-normal net currents can still be divided into outgoing and incoming partial currents across the node boundaries. For x -direction the result is

$$\bar{J}_{gx}^k\left(\pm\frac{h}{2}\right) = \pm \left(\bar{J}_{gx}^{\text{out},k}\left(\pm\frac{h}{2}\right) - \bar{J}_{gx}^{\text{in},k}\left(\pm\frac{h}{2}\right) \right), \quad (3.21)$$

where the partial currents are defined as

$$\begin{aligned} \bar{J}_{gx}^{\text{out},k}\left(\pm\frac{h}{2}\right) &= \frac{1}{\Delta z^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \frac{1}{2y_s(x)} \int_{-y_s(x)}^{y_s(x)} dy \\ &\quad \left(\frac{1}{4} \phi_g^k(x, y, z) \mp \frac{1}{2} D_g^k \frac{\partial}{\partial x} \phi_g^k(x, y, z) \right) \Big|_{x=\pm\frac{h}{2}} \end{aligned} \quad (3.22)$$

and

$$\begin{aligned} \bar{J}_{gx}^{\text{in},k}\left(\pm\frac{h}{2}\right) &= \frac{1}{\Delta z^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \frac{1}{2y_s(x)} \int_{-y_s(x)}^{y_s(x)} dy \\ &\quad \left(\frac{1}{4} \phi_g^k(x, y, z) \pm \frac{1}{2} D_g^k \frac{\partial}{\partial x} \phi_g^k(x, y, z) \right) \Big|_{x=\pm\frac{h}{2}}. \end{aligned} \quad (3.23)$$

For the axial direction, the division is

$$\bar{J}_{gz}^k\left(\pm\frac{\Delta z^k}{2}\right) = \pm \left(\bar{J}_{gz}^{\text{out},k}\left(\pm\frac{\Delta z^k}{2}\right) - \bar{J}_{gz}^{\text{in},k}\left(\pm\frac{\Delta z^k}{2}\right) \right), \quad (3.24)$$

where the partial currents are defined as

$$\begin{aligned} \bar{J}_{gz}^{\text{out},k}\left(\pm\frac{\Delta z^k}{2}\right) &= \frac{\Delta z^k}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \\ &\quad \left(\frac{1}{4} \phi_g^k(x, y, z) \mp \frac{1}{2} D_g^k \frac{\partial}{\partial z} \phi_g^k(x, y, z) \right) \Big|_{x=\pm\frac{\Delta z^k}{2}} \end{aligned} \quad (3.25)$$

and

$$\begin{aligned} \bar{J}_{gz}^{\text{in},k}\left(\pm\frac{\Delta z^k}{2}\right) &= \frac{\Delta z^k}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \\ &\quad \left(\frac{1}{4} \phi_g^k(x, y, z) \pm \frac{1}{2} D_g^k \frac{\partial}{\partial z} \phi_g^k(x, y, z) \right) \Big|_{x=\pm\frac{\Delta z^k}{2}}. \end{aligned} \quad (3.26)$$

3.3.2 Spatial boundary conditions

In addition to the initial conditions provided by neutron flux and precursor densities at time t_n , the solution of Eq. (3.7) is subject to spatial boundary conditions defining the couplings between adjacent nodes and the reactor boundaries.

Boundary conditions between adjacent nodes k and l can be written in terms of surface averaged incoming and outgoing partial currents by requiring them to be continuous across the nodal interfaces [28]. This results in the hex-plane directions in

$$\bar{J}_{gs}^{\text{in},k}\left(\frac{h}{2}\right) = \bar{J}_{gs}^{\text{out},l}\left(-\frac{h}{2}\right) \quad (3.27a)$$

$$\bar{J}_{gs}^{\text{in},l}\left(-\frac{h}{2}\right) = \bar{J}_{gs}^{\text{out},k}\left(\frac{h}{2}\right), \quad (3.27b)$$

where $s \in \{x, u, v\}$, and in the axial direction in

$$\bar{J}_{gz}^{\text{in},k}\left(\frac{\Delta z^k}{2}\right) = \bar{J}_{gz}^{\text{out},l}\left(-\frac{\Delta z^k}{2}\right) \quad (3.28a)$$

$$\bar{J}_{gz}^{\text{in},l}\left(-\frac{\Delta z^k}{2}\right) = \bar{J}_{gz}^{\text{out},k}\left(\frac{\Delta z^k}{2}\right), \quad (3.28b)$$

where the coordinates are local node coordinates for each node k and l . The Eqs. (3.27) and (3.28) are equivalent in an integral sense to the physical requirement of the continuity of the neutron flux and surface-normal net current across an interface.

At the reactor outer boundary S , a general boundary condition

$$a_g \phi_g^k(\vec{r}) + 2b_g \hat{n} \cdot D_g^k \nabla \phi_g^k(\vec{r}) = 0, \quad (3.29)$$

can be applied [28]. Here $\vec{r} \in S$, \hat{n} is the unit vector pointing outwards from the reactor boundary and a_g and b_g are coefficients specifying the boundary condition type. For example, a zero flux boundary condition can be set with $a_g = 1$ and $b_g = 0$. With the partial current notation, this boundary condition can be presented as

$$\bar{J}_{gs}^{\text{in},k}(\vec{r}) = \gamma_g \bar{J}_{gs}^{\text{out},k}(\vec{r}), \quad (3.30)$$

where $\vec{r} \in S$ and $s \in \{x, u, v, z\}$ depending on the direction of the boundary. The coefficient

$$\gamma_g = \frac{b_g - a_g}{b_g + a_g}, \quad (3.31)$$

when $b_g \neq -a_g$ specifies the boundary condition type. Some typical boundary conditions are defined as

$$\gamma_g = \begin{cases} -1 & \text{zero flux boundary condition,} \\ 0 & \text{zero incoming partial current boundary condition,} \\ 1 & \text{zero net current or reflective boundary condition.} \end{cases} \quad (3.32)$$

3.3.3 Transverse integration

The partial currents required to solve Eq. (3.14) can be solved using different methods. The transverse integration method, as described in detail in Ref. [28], is utilized in this work. In the method, the three-dimensional diffusion equation is reduced to three one-dimensional equations in the hex-plane and one one-dimensional equation in the axial direction.

Using the partially integrated flux

$$\phi_{gx}^k(x) = \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \int_{-y_s(x)}^{y_s(x)} dy \phi_g^k(x, y, z), \quad (3.33)$$

current

$$J_{gx}^k(x) = \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \int_{-y_s(x)}^{y_s(x)} dy \left(-D_g^k \frac{\partial}{\partial x} \phi_g^k(x, y, z) \right) \quad (3.34)$$

and source

$$Q_{gx}^k(x) = \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \int_{-y_s(x)}^{y_s(x)} dy Q_g^k(x, y, z), \quad (3.35)$$

and performing a neutron balance on a differential slice perpendicular to the x -direction defined as

$$\delta V^k : (x, y, z), x \in [x, x + dx], y \in [-y_s(x), y_s(x)], z \in \left[-\frac{\Delta z^k}{2}, \frac{\Delta z^k}{2} \right] \quad (3.36)$$

the balance equation

$$\begin{aligned} \frac{d}{dx} J_{gx}^k(x) + \tilde{\Sigma}_{r,g}^k \phi_{gx}^k(x) &= Q_{gx}^k(x) \\ &- \frac{2}{\sqrt{3}} (J_{gxy}^k(x, y_s(x)) - J_{gxy}^k(x, -y_s(x))) - \int_{-y_s(x)}^{y_s(x)} dy L_{gz}^k(x, y) \end{aligned} \quad (3.37)$$

in the x -direction is obtained. Here $J_{gxy}^k(x, \pm y_s(x))$ are z -integrated surface-normal components of the net current across the u - and v -directed surfaces, defined as

$$J_{gxy}^k(x, \pm y_s(x)) = \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \left(\mp D_g^k \hat{n}_{\pm} \cdot \nabla \phi_g^k(x, y, z) \right) \Big|_{y=\pm y_s(x)}, \quad (3.38)$$

where the unit normal vectors are defined as

$$\hat{n}_+ = \begin{cases} \hat{n}_{v+}, & -\frac{h}{2} < x < 0 \\ \hat{n}_{u+}, & 0 < x < \frac{h}{2} \end{cases} \quad (3.39a)$$

$$\hat{n}_- = \begin{cases} \hat{n}_{u-}, & -\frac{h}{2} < x < 0 \\ \hat{n}_{v-}, & 0 < x < \frac{h}{2}, \end{cases} \quad (3.39b)$$

each pointing outwards from the node surface.

The axial leakage term in Eq. (3.37) is defined as

$$L_{gz}^k(x, y) = \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \left(-D_g^k \frac{\partial^2}{\partial z^2} \phi_g^k(x, y, z) \right) = -D_g^k \frac{\partial}{\partial z} \phi_g^k(x, y, z) \Big|_{z=-\frac{\Delta z^k}{2}}^{z=\frac{\Delta z^k}{2}}. \quad (3.40)$$

The average axial leakage is

$$\bar{L}_{gz}^k = \frac{\Delta z^k}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy L_{gz}^k(x, y), \quad (3.41)$$

consistently with Eqs. (3.19) and (3.20).

Differentiating Eq. (3.33) as

$$\begin{aligned} -D_g^k \frac{d}{dx} \phi_{gx}^k(x) &= -D_g^k \frac{d}{dx} \int_{-y_s(x)}^{y_s(x)} dy \phi_{gxy}^k(x, y) \\ &= -D_g^k \int_{-y_s(x)}^{y_s(x)} dy \frac{\partial}{\partial x} \phi_{gxy}^k(x, y) \\ &\quad - D_g^k \frac{dy_s(x)}{dx} (\phi_{gxy}^k(x, y_s(x)) + \phi_g^k(x, -y_s(x))), \end{aligned} \quad (3.42)$$

where

$$\phi_{gxy}^k(x, y) = \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \phi_g^k(x, y, z) \quad (3.43)$$

and

$$\frac{dy_s(x)}{dx} = -\frac{1}{\sqrt{3}} \operatorname{sgn}(x), \quad (3.44)$$

which is obtained by differentiating Eq. (3.9), and substituting Eq. (3.34) into Eq. (3.42), the relationship between $\phi_{gx}^k(x)$ and $J_{gx}^k(x)$ is obtained as

$$J_{gx}^k = -D_g^k \frac{d}{dx} \phi_{gx}^k(x) + D_g^k \frac{dy_s(x)}{dx} (\phi_{gxy}^k(x, y_s(x)) + \phi_{gxy}^k(x, -y_s(x))). \quad (3.45)$$

3.3. SPATIAL SOLUTION METHOD

Similar relationship between the average values $\bar{\phi}_{gx}^k(x)$ and $\bar{J}_{gx}^k(x)$, defined as

$$\bar{\phi}_{gx}^k(x) = \frac{1}{\Delta z^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \frac{1}{2y_s(x)} \int_{-y_s(x)}^{y_s(x)} dy \phi_g^k(x, y, z) = \frac{1}{\Delta z^k} \frac{1}{2y_s(x)} \phi_{gx}^k(x) \quad (3.46)$$

and

$$\begin{aligned} \bar{J}_{gx}^k(x) &= \frac{1}{\Delta z^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \frac{1}{2y_s(x)} \int_{-y_s(x)}^{y_s(x)} dy \left(-D_g^k \frac{\partial}{\partial x} \phi_g^k(x, y, z) \right) \\ &= \frac{1}{\Delta z^k} \frac{1}{2y_s(x)} J_{gx}^k(x) \end{aligned} \quad (3.47)$$

can be obtained as

$$\bar{J}_{gx}^k(x) = -D_g^k \frac{d}{dx} \bar{\phi}_{gx}^k(x) + D_g^k \frac{1}{2y_s(x)} \frac{dy_s(x)}{dx} E_{gx}^k(x), \quad (3.48)$$

where

$$E_{gx}^k(x) = \phi_{gxy}^k(x, y_s(x)) + \phi_{gxy}^k(x, -y_s(x)) - 2\bar{\phi}_{gx}^k(x). \quad (3.49)$$

In a similar way, equations corresponding to Eqs. (3.37), (3.45) and (3.48) in the u - and v -directions can be derived.

As in the x -direction, using the hex-plane integrated flux

$$\phi_{gz}^k(z) = \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \phi_g^k(x, y, z), \quad (3.50)$$

current

$$J_{gz}^k(z) = \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \left(-D_g^k \frac{\partial}{\partial z} \phi_g^k(x, y, z) \right) \quad (3.51)$$

and source

$$Q_{gz}^k(z) = \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy Q_g^k(x, y, z), \quad (3.52)$$

and performing a neutron balance in a differential slice perpendicular to z -direction, defined as

$$\delta V^k : (x, y, z), \quad x \in \left[-\frac{h}{2}, \frac{h}{2} \right], \quad y \in [-y_s(x), y_s(x)], \quad z \in [z, z + dz] \quad (3.53)$$

the axial balance equation

$$\frac{d}{dz} J_{gz}^k(z) + \tilde{\Sigma}_{r,g}^k \phi_{gz}^k = Q_{gz}^k(z) - L_{gxy}^k(z) \quad (3.54)$$

is obtained. Here the hex-plane leakage is defined as

$$L_{gxy}^k(z) = \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \left(-D_g^k \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \phi_g^k(x, y, z) \right) \quad (3.55)$$

and the corresponding sum of average leakages in the hex-plane as

$$\bar{L}_{gxy}^k = \frac{3h}{2} \frac{1}{V^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz L_{gxy}^k(z), \quad (3.56)$$

which can be separated to average leakages in the three hex-plane directions as

$$\bar{L}_{gxy}^k = \bar{L}_{gx}^k + \bar{L}_{gy}^k + \bar{L}_{gz}^k. \quad (3.57)$$

The hex-plane averaged flux is defined as

$$\bar{\phi}_{gz}^k(z) = \frac{\Delta z^k}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \phi_g^k(x, y, z) = \frac{\Delta z^k}{V^k} \phi_{gz}^k(z) \quad (3.58)$$

and the hex-plane averaged current as

$$\bar{J}_{gz}^k(z) = \frac{\Delta z^k}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \int_{-y_s(x)}^{y_s(x)} dy \left(-D_g^k \frac{\partial}{\partial x} \phi_g^k(x, y, z) \right) = \frac{\Delta z^k}{V^k} J_{gz}^k(z). \quad (3.59)$$

The relationship between $J_{gz}^k(z)$ and $\phi_{gz}^k(z)$ is given simply by Eqs. (3.50) and (3.51), as the integration limits of the latter equation do not depend on x . The result is

$$J_{gz}^k(z) = -D_g^k \frac{d}{dz} \phi_{gz}^k(z). \quad (3.60)$$

Analogously, the relationship between the average values is given by Eqs. (3.58) and (3.59) as

$$\bar{J}_{gz}^k(z) = -D_g^k \frac{d}{dz} \bar{\phi}_{gz}^k(z). \quad (3.61)$$

The results of the transverse integration are the coupled differential equations relating the partially integrated currents and fluxes. These one-dimensional fluxes can then be approximated with suitable functions. Additional approximations are required for the transverse integrated leakage terms.

3.3.4 Polynomial approximation in hex-plane

The one-dimensional fluxes are solved by applying polynomial approximations to the partially integrated fluxes. In the hex-plane directions, the approximation

3.3. SPATIAL SOLUTION METHOD

is given by four basis functions specifying the spatial accuracy of the model, as presented here for the x -direction as

$$\phi_{gx}^k(x) \approx \tilde{\phi}_{gx}^k(x) = \Delta z^k \cdot 2y_s(x) \left(\bar{\phi}_g^k + \sum_{n=1}^4 a_{gxn}^k f_n(x) \right), \quad (3.62)$$

where the expansion coefficients are

$$a_{gx1}^k = \bar{\phi}_{gx}^k \left(\frac{h}{2} \right) - \bar{\phi}_{gx}^k \left(-\frac{h}{2} \right) \quad (3.63a)$$

$$a_{gx2}^k = \bar{\phi}_{gx}^k \left(\frac{h}{2} \right) + \bar{\phi}_{gx}^k \left(-\frac{h}{2} \right) - 2\bar{\phi}_g^k \quad (3.63b)$$

$$a_{gx3}^k = -\frac{26}{189} \frac{h}{D_g^k} (\bar{L}_{gu}^k + \bar{L}_{gv}^k) - \frac{8}{21} \left(\bar{\phi}_{gx}^k \left(\frac{h}{2} \right) + \bar{\phi}_{gx}^k \left(-\frac{h}{2} \right) - 2\bar{\phi}_g^k \right) \quad (3.63c)$$

$$a_{gx4}^k = -24\bar{\phi}_{gx1}^k + \frac{16}{3} \left(\bar{\phi}_{gx}^k \left(\frac{h}{2} \right) - \bar{\phi}_{gx}^k \left(-\frac{h}{2} \right) \right), \quad (3.63d)$$

and the basis functions are

$$f_1(x) = \frac{x}{h} \quad (3.64a)$$

$$f_2(x) = \frac{36}{13} \left(\frac{x}{h} \right)^2 - \frac{5}{26} \quad (3.64b)$$

$$f_3(x) = \frac{10}{13} \left(\frac{x}{h} \right)^2 - \frac{1}{2} \left| \frac{x}{h} \right| + \frac{3}{52} \quad (3.64c)$$

$$f_4(x) = \frac{x}{h} \left(\left| \frac{x}{h} \right| - \frac{1}{2} \right). \quad (3.64d)$$

The derivation of the expansion coefficients and the motivations for the chosen basis functions are described in detail in Ref. [28]. Briefly, the polynomial approximation is constructed in a such way that that it preserves the node average flux

$$\frac{1}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \tilde{\phi}_{gx}^k(x) = \bar{\phi}_g^k \quad (3.65)$$

and the surface-averaged fluxes

$$\left(\frac{1}{\Delta z^k} \frac{1}{2y_s(x)} \tilde{\phi}_{gx}^k(x) \right) \Big|_{x=\pm \frac{h}{2}} = \bar{\phi}_{gx}^k \left(\pm \frac{h}{2} \right). \quad (3.66)$$

In addition, the basis functions satisfy the constraints

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} dx \, 2y_s(x) f_n(x) = 0, \quad n = 1 \dots 4 \quad (3.67)$$

and

$$f_n\left(\pm \frac{h}{2}\right) = 0, \quad n = 3, 4. \quad (3.68)$$

The first and second terms of the polynomial approximation are obtained by applying a quadratic polynomial approximation to the partially integrated flux. The third basis function is added to provide a first derivative discontinuity at $x = 0$. Its expansion coefficient is obtained by requiring the partially integrated net current $J_{gx}^k(x)$ to be continuous at $x = 0$. The fourth basis function provides quadratic approximations for both half-intervals $-\frac{h}{2} < x < 0$ and $0 < x < \frac{h}{2}$ of the node. Its expansion coefficient is solved by applying the inner product

$$\langle w(x), f(x) \rangle = \frac{1}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \, w(x) f(x) \quad (3.69)$$

with a weight function $w(x)$ to the one-dimensional neutron balance equation (3.37), using the half-node weight function

$$w(x) = w_1(x) = \text{sgn}(x). \quad (3.70)$$

The chosen weight function preserves the neutron balance in both of the half-nodes in the x -direction. Similarly, the spatial flux moment ϕ_{gx1} present in Eq. (3.63d) is defined as

$$\phi_{gx1}^k = \langle w_1(x), \phi_{gx}^k(x) \rangle. \quad (3.71)$$

The solution of the flux moment is described in Sec. 3.3.6.

3.3.5 Polynomial approximation in axial direction

For the axial flux, the one-dimensional polynomial approximation is applied as

$$\phi_{gz}^k(z) \approx \tilde{\phi}_{gz}^k(z) = \frac{V^k}{\Delta z^k} \left(\bar{\phi}_g^k + \sum_{n=1}^3 a_{gzn}^k f_{zn}(z) \right), \quad (3.72)$$

where the coefficients are defined as

$$a_{gz1}^k = \bar{\phi}_{gz}^k \left(\frac{\Delta z^k}{2} \right) - \bar{\phi}_{gz}^k \left(-\frac{\Delta z^k}{2} \right) \quad (3.73a)$$

$$a_{gz2}^k = \bar{\phi}_{gz}^k \left(\frac{\Delta z^k}{2} \right) + \bar{\phi}_{gz}^k \left(-\frac{\Delta z^k}{2} \right) - 2\bar{\phi}_g^k \quad (3.73b)$$

$$a_{gz3}^k = -120\phi_{gz1}^k + 10 \left(\bar{\phi}_{gz}^k \left(\frac{\Delta z^k}{2} \right) - \bar{\phi}_{gz}^k \left(-\frac{\Delta z^k}{2} \right) \right) \quad (3.73c)$$

and the basis functions as

$$f_{z1}(z) = \frac{z}{\Delta z^k} \quad (3.74a)$$

$$f_{z2}(z) = 3 \left(\left(\frac{z}{\Delta z^k} \right)^2 - \frac{1}{4} \right) \quad (3.74b)$$

$$f_{z3}(z) = \frac{z}{\Delta z^k} \left(\frac{z}{\Delta z^k} - \frac{1}{2} \right) \left(\frac{z}{\Delta z^k} + \frac{1}{2} \right). \quad (3.74c)$$

The derivation of the expansion coefficients and the motivations for the chosen basis functions are described in detail in Ref. [28]. Briefly, the axial flux approximation is constructed so that the node averaged flux

$$\frac{1}{V^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \tilde{\phi}_{gz}^k(z) = \bar{\phi}_g^k \quad (3.75)$$

and the surface-averaged fluxes

$$\frac{\Delta z^k}{V^k} \tilde{\phi}_{gz}^k(z) \Big|_{z=\pm \frac{\Delta z^k}{2}} = \bar{\phi}_{gz}^k \left(\pm \frac{\Delta z^k}{2} \right) \quad (3.76)$$

are preserved. In addition, the basis functions satisfy the constraints

$$\int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz f_{zn}(z) = 0, \quad n = 1 \dots 3 \quad (3.77)$$

and

$$f_{z3} \left(\pm \frac{\Delta z^k}{2} \right) = 0. \quad (3.78)$$

As in the hex-plane directions, the first and second terms of the polynomial approximation are obtained by applying a quadratic polynomial approximation to the partially integrated flux. The third order basis function provides a third order term in the polynomial approximation. Its expansion coefficient is obtained by applying the inner product

$$\langle w(z), f(z) \rangle = \frac{1}{V^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz w(z) f(z) \quad (3.79)$$

with a weight function $w(z)$ to the one-dimensional neutron balance equation (3.54), using the moments weighting function

$$w(z) = w_{z1}(z) = f_{z1}(z), \quad (3.80)$$

defined in Eq. (3.74a). The spatial flux moment $\phi_{gz1}^k(z)$ present in Eq. (3.73c) is defined as

$$\phi_{gz1}^k = \langle w_{z1}(z), \phi_{gz}^k(z) \rangle. \quad (3.81)$$

The solution of the flux moment is described in Sec. 3.3.6.

3.3.6 Flux moments calculation

The node average, or the zeroth moment flux is obtained simply from Eq. (3.14) or applying unit weighting $w(x) = w_0(x) = 1$ with the inner product of Eq. (3.69) on Eq. (3.37) or unit weighting $w(z) = w_0(z) = 1$ with the inner product of Eq. (3.79) on Eq. (3.54). The result is

$$\bar{\phi}_g^k = \frac{1}{\tilde{\Sigma}_{r,g}^k} \bar{Q}_g^k - \frac{2}{3h\tilde{\Sigma}_{r,g}^k} (\bar{L}_{gx}^k + \bar{L}_{gu}^k + \bar{L}_{gv}^k) - \frac{1}{\Delta z^k \tilde{\Sigma}_{r,g}^k} \bar{L}_{gz}^k. \quad (3.82)$$

The first spatial flux moment in x -direction is obtained by applying the half-node weighting function defined in Eq. (3.70) with the inner product of Eq. (3.69) on Eq. (3.37). The result is

$$\left(\tilde{\Sigma}_{r,g}^k + \frac{32}{h} \frac{D_g^k}{h} \right) \phi_{gx1}^k = Q_{gx1}^k - \frac{1}{\Delta z^k} L_{gzx1}^k - \frac{2}{3h} (\bar{T}_{gx}^k + \bar{T}_{gu}^k - \bar{T}_{gv}^k) + \frac{40}{9h} \frac{D_g^k}{h} a_{gx1}^k, \quad (3.83)$$

where

$$L_{gzx1}^k = \frac{\Delta z^k}{V^k} \int_{-\frac{h}{2}}^{\frac{h}{2}} dx \operatorname{sgn}(x) \int_{-y_s(x)}^{y_s(x)} dy L_{gz}^k(x, y) \quad (3.84)$$

and

$$\bar{T}_{gx}^k = \bar{J}_{gx}^k \left(\frac{h}{2} \right) + \bar{J}_{gx}^k \left(-\frac{h}{2} \right). \quad (3.85)$$

The two remaining hex-plane spatial flux moments are obtained similarly by substituting

$$x \rightarrow u, u \rightarrow v, v \rightarrow -x \quad (3.86)$$

in Eq. (3.83) to obtain the u -direction equation and

$$x \rightarrow v, u \rightarrow -x, v \rightarrow -u \quad (3.87)$$

to obtain the v -direction equation.

3.3. SPATIAL SOLUTION METHOD

Applying the moments weight function defined in Eq. (3.80) with the inner product of Eq. (3.79) on Eq. (3.54), the equation for the first spatial flux moment in the axial direction is obtained as

$$\tilde{\Sigma}_{r,g}^k \phi_{gz1}^k = Q_{gz1}^k - \frac{2}{3h} L_{gxyz1}^k - \frac{1}{2} \frac{1}{\Delta z^k} \bar{T}_{gz}^k - \frac{D_g^k}{(\Delta z^k)^2} a_{gz1}^k, \quad (3.88)$$

where

$$L_{gxyz1}^k = \frac{3h}{2} \frac{1}{V^k} \int_{-\frac{\Delta z^k}{2}}^{\frac{\Delta z^k}{2}} dz \frac{z}{\Delta z^k} L_{gxy}^k(z) \quad (3.89)$$

and

$$\bar{T}_{gz}^k = \bar{J}_{gz}^k \left(\frac{\Delta z^k}{2} \right) + \bar{J}_{gz}^k \left(-\frac{\Delta z^k}{2} \right). \quad (3.90)$$

Gathering Eq. (3.83) with its u - and v -direction counterparts together with Eq. (3.88), the first moments equations can be presented as

$$\begin{aligned} \begin{pmatrix} \phi_{gx1}^k \\ \phi_{gu1}^k \\ \phi_{gv1}^k \\ \phi_{gz1}^k \end{pmatrix} &= \frac{h}{\alpha_{g1}^k} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \alpha_{g2}^k \end{pmatrix} \begin{pmatrix} Q_{gx1}^k - \frac{1}{\Delta z^k} L_{gzx1}^k \\ Q_{gu1}^k - \frac{1}{\Delta z^k} L_{gzu1}^k \\ Q_{gv1}^k - \frac{1}{\Delta z^k} L_{gzv1}^k \\ Q_{gz1}^k - \frac{2}{3h} L_{gxyz1}^k \end{pmatrix} \\ &\quad - \frac{2}{3\alpha_{g1}^k} \begin{pmatrix} 1 & 1 & -1 & 0 \\ 1 & 1 & 1 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & \alpha_{g3}^k \end{pmatrix} \begin{pmatrix} \bar{T}_{gx}^k \\ \bar{T}_{gu}^k \\ \bar{T}_{gv}^k \\ \bar{T}_{gz}^k \end{pmatrix} \\ &\quad + \frac{40}{9\alpha_{g1}^k} \frac{D_g^k}{h} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\alpha_{g4}^k \end{pmatrix} \begin{pmatrix} a_{gx1}^k \\ a_{gu1}^k \\ a_{gv1}^k \\ a_{gz1}^k \end{pmatrix}, \end{aligned} \quad (3.91)$$

where the coefficients are defined as

$$\alpha_{g1}^k = h \tilde{\Sigma}_{r,g}^k + 32 \frac{D_g^k}{h} \quad (3.92a)$$

$$\alpha_{g2}^k = \frac{\alpha_{g1}^k}{h\tilde{\Sigma}_{r,g}^k} \quad (3.92b)$$

$$\alpha_{g3}^k = \frac{3\alpha_{g1}^k}{4\Delta z^k \tilde{\Sigma}_{r,g}^k} \quad (3.92c)$$

$$\alpha_{g4}^k = \frac{9\alpha_{g1}^k}{40} \frac{h}{(\Delta z^k)^2 \tilde{\Sigma}_{r,g}^k}. \quad (3.92d)$$

The full derivation of the equation is presented in Ref. [28].

3.3.7 Approximation of leakage moments

Eq. (3.91) includes leakage moments, which need to be approximated. A detailed discussion and the derivations of the leakage moment approximations are presented in Ref. [28].

The axial leakage defined in Eq. (3.40) is approximated by its average value over the z -directed surfaces, defined in Eq. (3.41), as

$$L_{gz}^k(x, y) \approx \bar{L}_{gz}^k. \quad (3.93)$$

Consequently from Eq. (3.84),

$$L_{gzx1}^k = L_{gzu1}^k = L_{gzv1}^k = 0. \quad (3.94)$$

The hex-plane leakage defined in Eq. (3.55) is approximated within the node with a quadratic polynomial

$$L_{gxy}^k(z) \approx \rho_{gxy}^k(z) = \frac{2}{3h} V^k (\bar{L}_{gxy} + \rho_{gxy1}^k f_{z1}(z) + \rho_{gxy2}^k f_{z2}(z)) \quad (3.95)$$

defined between the bottom

$$z_1 = -\frac{\Delta z^k}{2} - \Delta z^{k-} \quad (3.96)$$

and the top

$$z_2 = \frac{\Delta z^k}{2} + \Delta z^{k+}, \quad (3.97)$$

of the neighboring nodes. Here k^- and k^+ denote the nodes below and above the node k , respectively.

The expansion coefficients in Eq. (3.95) are defined in a such way that the average hex-plane leakages in nodes k^- and k^+ are preserved as

$$\bar{L}_{gxy}^{k-} = \frac{3h}{2} \frac{1}{V^{k-}} \int_{z_1}^{-\frac{\Delta z^k}{2}} dz \rho_{gxy}^k(z) \quad (3.98a)$$

$$\bar{L}_{gxy}^{k+} = \frac{3h}{2} \frac{1}{V^{k+}} \int_{\frac{\Delta z^k}{2}}^{z_2} dz \rho_{gxy}^k(z). \quad (3.98b)$$

This results in

$$\begin{aligned} \rho_{gxy1}^k(z) = \frac{2}{3h} \frac{V^k}{d} & ((2\Delta z^{k-} + \Delta z^k)(\Delta z^{k-} + \Delta z^k)(\bar{L}_{gxy}^{k+} - \bar{L}_{gxy}^k) \\ & + (\Delta z^k + \Delta z^{k+})(\Delta z^k + 2\Delta z^{k+})(\bar{L}_{gxy}^k - \bar{L}_{gxy}^{k+})) \end{aligned} \quad (3.99a)$$

$$\begin{aligned} \rho_{gxy2}^k(z) = \frac{2}{3h} \frac{V^k}{d} \Delta z^k & ((\Delta z^{k-} + \Delta z^k)(\bar{L}_{gxy}^{k+} - \bar{L}_{gxy}^k) \\ & - (\Delta z^k + \Delta z^{k+})(\bar{L}_{gxy}^k - \bar{L}_{gxy}^{k-})). \end{aligned} \quad (3.99b)$$

Substituting the approximation of Eq. (3.95) into Eq. (3.89) leads to

$$L_{gxyz1}^k = \frac{3h}{2} \frac{\Delta z^k}{V^k} \frac{1}{12} \rho_{gxy1}^k, \quad (3.100)$$

which can be reformulated as

$$L_{gxyz1}^k = \mu_-^k (\bar{L}_{gxy}^k - \bar{L}_{gxy}^{k-}) + \mu_+^k (\bar{L}_{gxy}^{k+} - \bar{L}_{gxy}^k), \quad (3.101)$$

where

$$\mu_-^k = \frac{\Delta z^k}{12d} (2\Delta z^{k+} + \Delta z^k)(\Delta z^{k+} + \Delta z^k) \quad (3.102a)$$

$$\mu_+^k = \frac{\Delta z^k}{12d} (2\Delta z^{k-} + \Delta z^k)(\Delta z^{k-} + \Delta z^k) \quad (3.102b)$$

and

$$d = (\Delta z^{k-} + \Delta z^k)(\Delta z^{k-} + \Delta z^k + \Delta z^{k+})(\Delta z^k + \Delta z^{k+}). \quad (3.103)$$

For calculating the hex-plane leakage moment of Eq. (3.101) for a node adjacent to the bottom or top boundary of the reactor, the average hex-plane leakage and the height of the adjacent node lying outside the reactor boundary are set to zero. This is consistent with an approximation that the hex-plane leakage is zero at the reactor-sided boundary of the adjacent node lying outside the reactor. For a zero-incoming partial current boundary condition, this is an additional approximation. [28]

3.3.8 Response matrix equation

A major part of the solution method is the response matrix equation, which relates the outgoing face-averaged partial currents of a node to the incoming face-averaged partial currents and source and leakage moments. [28]

In order to solve the outgoing partial currents, all the surface-averaged fluxes and net currents of the approximations described in Secs. 3.3.4 and 3.3.5 are eliminated

in favor of the face-averaged partial currents. The face-averaged net currents are defined in Eqs. (3.18) and (3.20) for x - and axial directions, respectively, and can be eliminated with Eqs. (3.21) and (3.24). The required partial currents are defined in Eqs. (3.22), (3.23), (3.25) and (3.26). The surface averaged fluxes, defined in Eqs. (3.46) and (3.58), can be written in terms of the partial currents as

$$\bar{\phi}_{gx}^k\left(\pm\frac{h}{2}\right) = 2\left(\bar{J}_{gx}^{\text{out},k}\left(\pm\frac{h}{2}\right) + \bar{J}_{gx}^{\text{in},k}\left(\pm\frac{h}{2}\right)\right) \quad (3.104)$$

and

$$\bar{\phi}_{gz}^k\left(\pm\frac{\Delta z^k}{2}\right) = 2\left(\bar{J}_{gz}^{\text{out},k}\left(\pm\frac{\Delta z^k}{2}\right) + \bar{J}_{gz}^{\text{in},k}\left(\pm\frac{\Delta z^k}{2}\right)\right) \quad (3.105)$$

for the x - and axial directions, respectively. The relationships are straightforwardly obtained from the definitions of the face-averaged fluxes and partial currents.

Using the x -direction polynomial approximation of Eq. (3.62) in Eq. (3.48) and the axial direction polynomial approximation of Eq. (3.72) in Eq. (3.61), the equations

$$\bar{J}_{gx}^{\text{out},k}\left(\frac{h}{2}\right) = -\frac{D_g^k}{h}\left(a_{gx1}^k + \frac{36}{13}a_{gx2}^k + \frac{7}{26}a_{gx3}^k + \frac{1}{2}a_{gx4}^k + E_{gx}^k\left(\frac{h}{2}\right)\right) + \bar{J}_{gx}^{\text{in},k}\left(\frac{h}{2}\right) \quad (3.106)$$

and

$$\bar{J}_{gz}^{\text{out},k}\left(\frac{\Delta z^k}{2}\right) = -\frac{D_g^k}{\Delta z^k}\left(a_{gz1}^k + 3a_{gz2}^k + \frac{1}{2}a_{gz3}^k\right) + \bar{J}_{gz}^{\text{in},k}\left(\frac{\Delta z^k}{2}\right) \quad (3.107)$$

are obtained. Finally, by eliminating the expansion coefficients using Eqs. (3.63) and (3.73), the term E_{gx}^k presented in Eq. (3.49), flux moments in favor of the source and leakage moments using Eqs. (3.82) and (3.91) and the net currents in favor of the partial currents, the equation

$$\begin{aligned} (a_1 \ a_2 \ a_3 \ a_4 \ a_3 \ a_2 \ a_5 \ a_5) \mathbf{J}_g^{\text{out},k} &= (b_1 \ b_2) \begin{pmatrix} \bar{Q}_g^k \\ Q_{gx1}^k \end{pmatrix} \\ &+ (c_1 \ c_2 \ c_3 \ c_4 \ c_3 \ c_2 \ c_5 \ c_5) \mathbf{J}_g^{\text{in},k} \end{aligned} \quad (3.108)$$

is obtained for the x -direction and the equation

$$\begin{aligned} (a_6 \ a_6 \ a_6 \ a_6 \ a_6 \ a_6 \ a_7 \ a_8) \mathbf{J}_g^{\text{out},k} &= (b_3 \ b_4) \begin{pmatrix} \bar{Q}_g^k \\ Q_{gz1}^k - \frac{2}{3h}L_{gxyz1}^k \end{pmatrix} \\ &+ (c_6 \ c_6 \ c_6 \ c_6 \ c_6 \ c_6 \ c_7 \ c_8) \mathbf{J}_g^{\text{in},k} \end{aligned} \quad (3.109)$$

for the axial direction. In the equations, the outgoing and incoming partial current vectors are defined as

$$\begin{aligned} \mathbf{J}_g^{\text{out},k} &= \begin{pmatrix} \bar{J}_{gx}^{\text{out},k}\left(\frac{h}{2}\right) & \bar{J}_{gu}^{\text{out},k}\left(\frac{h}{2}\right) & \bar{J}_{gv}^{\text{out},k}\left(\frac{h}{2}\right) & \bar{J}_{gx}^{\text{out},k}\left(-\frac{h}{2}\right) & \bar{J}_{gu}^{\text{out},k}\left(-\frac{h}{2}\right) \\ & \bar{J}_{gv}^{\text{out},k}\left(-\frac{h}{2}\right) & \bar{J}_{gz}^{\text{out},k}\left(\frac{\Delta z^k}{2}\right) & \bar{J}_{gz}^{\text{out},k}\left(-\frac{\Delta z^k}{2}\right) \end{pmatrix}^T \\ &= \begin{pmatrix} \mathbf{J}_{gxy}^{\text{out},k} \\ \mathbf{J}_{gz}^{\text{out},k} \end{pmatrix}. \end{aligned} \quad (3.110)$$

Together with the u - and v -direction versions of Eq. (3.108) and Eq. (3.109), the resulting matrix equation can be written in the following form

$$\mathbf{A}_g^k \mathbf{J}_g^{\text{out},k} = \mathbf{B}_g^k (\mathbf{Q}_g^k - \mathbf{L}_g^k) + \mathbf{C}_g^k \mathbf{J}_g^{\text{in},k}, \quad (3.111)$$

where the source vector is defined as

$$\mathbf{Q}_g^k = \begin{pmatrix} \bar{Q}_g^k & Q_{gx1}^k & Q_{gu1}^k & Q_{gv1}^k & Q_{gz1}^k \end{pmatrix}^T, \quad (3.112)$$

and the leakage vector as

$$\mathbf{L}_g^k = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{2}{3h} L_{gxyz1}^k \end{pmatrix}^T, \quad (3.113)$$

where the leakage moment approximations presented in Sec. 3.3.7 are taken into account. The coefficients a_i , b_i and c_i present in Eqs. (3.108) and (3.109) form the matrices \mathbf{A} , \mathbf{B} and \mathbf{C} , respectively. Finally, the outgoing partial currents can be solved from the response matrix equation obtained from Eq. (3.111) by inverting \mathbf{A}_g^k as

$$\mathbf{J}_g^{\text{out},k} = \mathbf{P}_g^k (\mathbf{Q}_g^k - \mathbf{L}_g^k) + \mathbf{R}_g^k \mathbf{J}_g^{\text{in},k}, \quad (3.114)$$

where the coefficient matrices are defined as

$$\mathbf{P}_g^k = (\mathbf{A}_g^k)^{-1} \mathbf{B}_g^k = \begin{pmatrix} \mathbf{P}_{gxy}^k \\ \mathbf{P}_{gz}^k \end{pmatrix} \quad (3.115)$$

and

$$\mathbf{R}_g^k = (\mathbf{A}_g^k)^{-1} \mathbf{C}_g^k = \begin{pmatrix} \mathbf{R}_{gxy}^k & \mathbf{R}_{gxyz}^k \\ \mathbf{R}_{gzxy}^k & \mathbf{R}_{gz}^k \end{pmatrix}. \quad (3.116)$$

The full derivation of the response matrix equation including the expressions for the coefficients a_i , b_i and c_i is presented in Ref. [28]. The coefficients of the matrices \mathbf{P}_g^k and \mathbf{R}_g^k depend only on nodewise group constants and the width and the height of the respective node. Therefore, they can be calculated at the beginning of each time step without needing to be updated during the solution process of one time step.

3.4 Computational solution procedure

Until now, the discussion has been concentrating on the theoretical derivation of the solution method of the time-dependent neutron flux inside a single homogeneous node. However, as the actual objective is to solve the fluxes computationally inside a full reactor core, the solution algorithm is discussed next. As it is desirable that the solution is obtained in as short time as possible, different acceleration methods used to speed up the convergence of the solution are also presented here.

3.4.1 Transient calculation solution algorithm

The transient solution algorithm of a single time step, based on a fission source iteration process, is presented in Fig. 3.2. At the beginning of the time step, control rod positions are updated, the nodewise group constants are calculated, and the coefficients of the response matrix equation (3.114) matrices are calculated. In addition, the part of the source term depending on the flux at the beginning of the time step, called the fixed source, and the part of the delayed neutron source depending on the information at the beginning of the time step, called the fixed part of delayed neutron source, are calculated. Afterwards, the temporal extrapolation acceleration method, described in Sec. 3.4.3, is applied for the fluxes and partial currents.

During the outer iterations, the asymptotic source extrapolation acceleration method, described in Sec. 3.4.4, is applied first. Then, during the inner or energy group iterations, for all the energy groups and nodes starting from the group with the highest energy the group source term moment vectors for iteration i are calculated using

$$\begin{aligned} \mathbf{Q}_g^{k,n+1,(i)} = & \left((1 - \beta^k) \chi_{p,g}^k + \chi_{d,g}^k \sum_{m=1}^M \lambda_m^k F_{1,m}^{k,n} \right) \boldsymbol{\psi}^{k,n+1,(i)} + \\ & \sum_{g'=1}^{g-1} \Sigma_{s,g' \rightarrow g}^k \boldsymbol{\phi}_{g'}^{k,n+1,(i)} + \frac{1}{v_g^k \Delta t_n} \boldsymbol{\phi}_g^{k,n} + \\ & \chi_{d,g}^k \sum_{m=1}^M \left(\lambda_m^k \mathbf{C}_m^{k,n} e^{-\lambda_m^k \Delta t_n} + \lambda_m^k F_{0,m}^n \boldsymbol{\psi}^{k,n} \right), \end{aligned} \quad (3.117)$$

which is obtained from Eq. (3.13) by applying the different moments weightings to it. Here, the fission source moment vector is defined as

$$\boldsymbol{\psi}^{k,n,(i)} = \sum_{g'=1}^G \nu \Sigma_{f,g'}^k \boldsymbol{\phi}_{g'}^{k,n,(i)}, \quad (3.118)$$

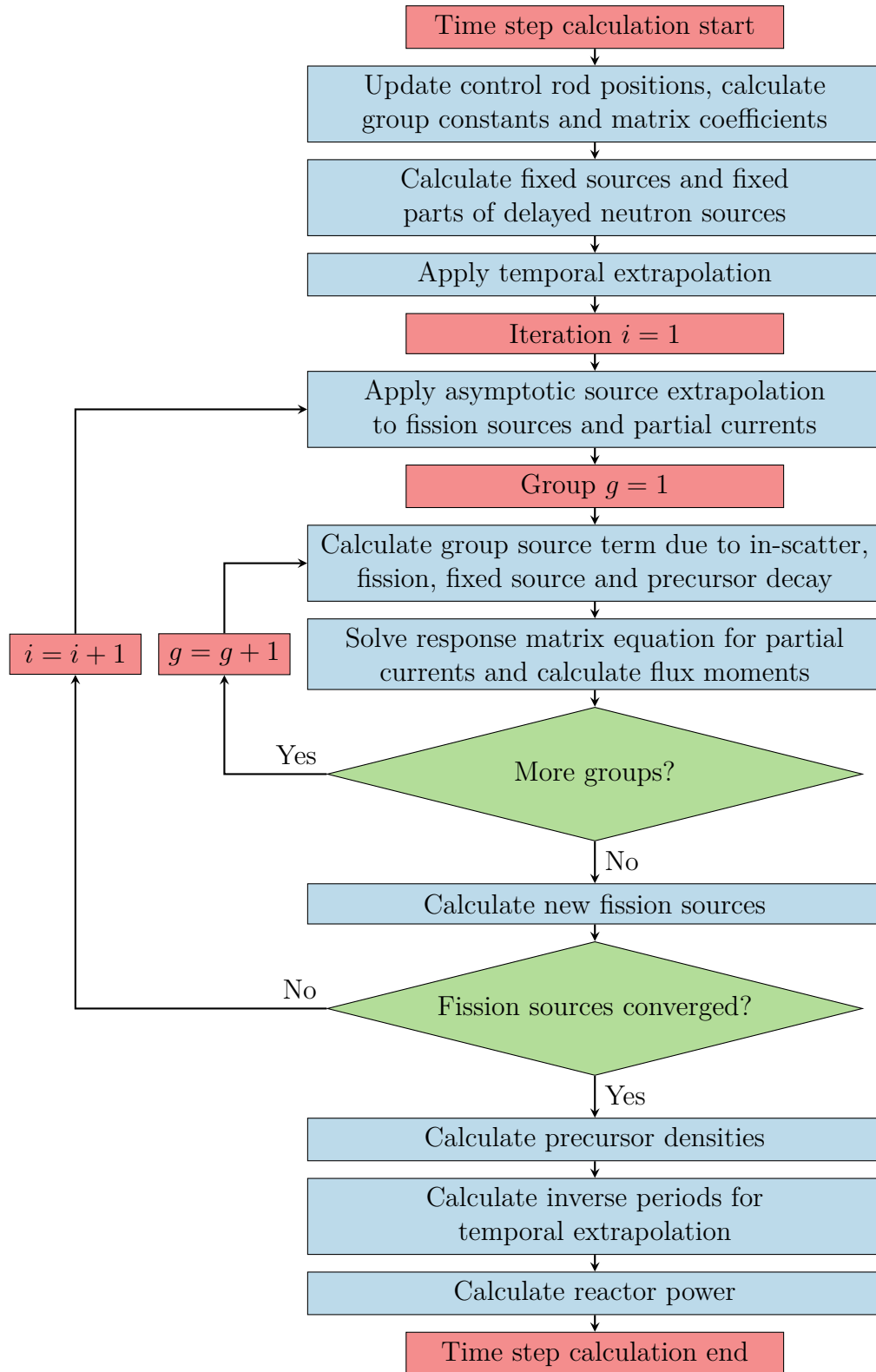


Figure 3.2: Solution algorithm of one time step during a transient calculation.

3.4. COMPUTATIONAL SOLUTION PROCEDURE

the flux moment vector as

$$\boldsymbol{\phi}_g^{k,n,(i)} = \left(\phi_g^{k,n,(i)} \quad \phi_{gx1}^{k,n,(i)} \quad \phi_{gu1}^{k,n,(i)} \quad \phi_{gv1}^{k,n,(i)} \quad \phi_{gz1}^{k,n,(i)} \right)^T, \quad (3.119)$$

and the precursor density moment vector as

$$\mathbf{C}_m^{k,n} = \left(C_m^{k,n} \quad C_{mx1}^{k,n} \quad C_{mu1}^{k,n} \quad C_{mv1}^{k,n} \quad C_{mz1}^{k,n} \right)^T. \quad (3.120)$$

Using the group source terms, the outgoing partial currents are solved from the response matrix equation (3.114) with the procedure described in Sec. 3.4.2, and the flux moments are calculated using Eqs. (3.82) and (3.91). After the energy group loop, new fission sources are calculated and their convergence is checked for the pointwise maximum relative change between successive iterations

$$\max_k \left| \frac{\psi^{k,n+1,(i)} - \psi^{k,n+1,(i-1)}}{\psi^{k,n+1,(i)}} \right| < \varepsilon_2 \quad (3.121)$$

and for the average fission source change inside the reactor

$$\frac{1}{K} \sqrt{\sum_{k=1}^K \left(\frac{\psi^{k,n+1,(i)} - \psi^{k,n+1,(i-1)}}{\psi^{k,n+1,(i)}} \right)^2} < \varepsilon_3, \quad (3.122)$$

where ψ refers to the node average, or zeroth moment fission source, ε_2 and ε_3 are convergence criteria parameters set by the user and K is the number of nodes inside the reactor. The convergence criterion parameter ε_1 of the multiplication factor is only defined for the steady state problems, and is described in Sec. 3.4.5.

If the criteria of Eqs. (3.121) and (3.122) were not met, the solution was not converged, and a new outer iteration is performed. If the criteria were met, the solution was converged. Then new precursor densities at the end of the time step are calculated with

$$\mathbf{C}_m^{k,n+1} = \mathbf{C}_m^{k,n} e^{-\lambda_m^k \Delta t_n} + F_{0,m}^{k,n} \boldsymbol{\psi}^{k,n} + F_{1,m}^{k,n} \boldsymbol{\psi}^{k,n+1}, \quad (3.123)$$

which is obtained by applying the different moments weightings to Eq. (3.5). Finally, the inverse periods for the temporal extrapolation at the beginning of the next time step are calculated and the reactor power is updated using the new fission sources.

As the transient solution method for one time step requires the neutron flux and precursor densities at the beginning of the time step, an initial state is needed. At the very beginning of a transient simulation, such a state can be produced by calculating a steady state k -eigenvalue calculation, described in Sec. 3.4.5. As the eigenvalue calculation can be performed in a truly non-steady state by scaling the

fission source with the effective multiplication factor, a neutron balance might not exist at the beginning of a transient calculation when using Eqs. (3.117) and (3.118) to calculate the group source terms. Therefore, when a steady state calculation is used to initialize a transient calculation, during the transient calculation all the neutron production cross sections in Eqs. (3.117), (3.118) and (3.123) are divided by the effective multiplication factor obtained from the steady state calculation. In addition, the precursor density moments at the beginning of the transient calculation are initialized by setting the time derivative of Eq. (3.2) to zero, resulting in

$$\mathbf{C}_m^{k,0} = \frac{\beta_m^k}{\lambda_m^k} \psi^{k,0} \quad (3.124)$$

written in terms of the different spatial moments. This procedure ensures the solution to be in a steady state at the beginning of a transient calculation.

3.4.2 Response matrix solution method

To solve the response matrix equation (3.114) inside a node, it is divided into hex-plane and axial direction equations as

$$\mathbf{J}_{gxy}^{\text{out},k} = \mathbf{P}_{gxy}^k \mathbf{Q}_g^k + \mathbf{R}_{gxy}^k \mathbf{J}_{gxy}^{\text{in},k} + \mathbf{R}_{gxyz}^k \mathbf{J}_{gz}^{\text{in},k}, \quad (3.125a)$$

$$\mathbf{J}_{gz}^{\text{out},k} = \mathbf{P}_{gz}^k (\mathbf{Q}_g^k - \mathbf{I}_g^k) + \mathbf{R}_{gz}^k \mathbf{J}_{gz}^{\text{in},k} + \mathbf{R}_{gzxy}^k \mathbf{J}_{gxy}^{\text{in},k}. \quad (3.125b)$$

The equations are solved by sweeping the nodes in the hex-plane and axial directions in the following way. First, the different axial planes are swept in odd-even ordering. For each hex-plane, the hex-plane partial currents are solved in a four-color checkerboard ordering, shown in Fig. 3.3. First the radial partial currents are solved twice using Eq. (3.125a) successively for each color in the odd-even and four-color ordering. After the radial partial currents are solved, the axial partial currents are calculated once using Eq. (3.125b) in the odd-even ordering for each plane. This partial current iteration is performed twice. [28]

The node colors are determined in a such way that two neighboring nodes will never have the same color and a possible third-core symmetry of a problem is preserved. The reason for the odd-even and four-color ordering is that the most recent outgoing partial currents of the neighboring nodes are used as the incoming partial currents of the current node. This Gauss-Seidel-like procedure is used to improve the convergence of the solution.

3.4.3 Temporal extrapolation

The flux at the end of the previous time step can be used as a starting guess for the fission source iteration of the current time step. However, in many cases

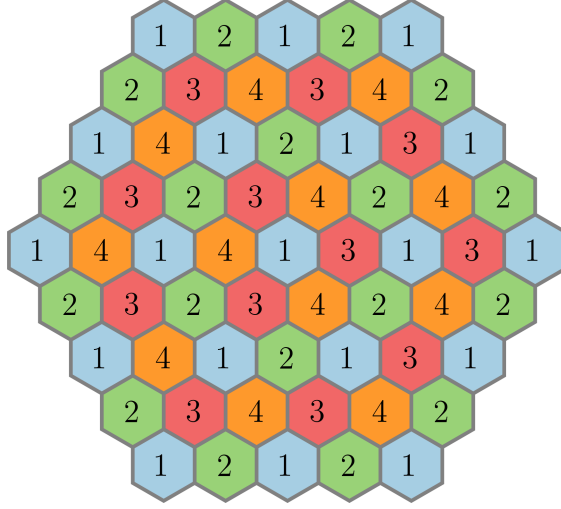


Figure 3.3: Four color checkerboard ordering of a hexagonal lattice.

this starting guess can be improved by assuming an exponential time behavior, extrapolated from the previous two time steps, for the flux during the current time step.

At the end of a time step $n - 1$, the instantaneous inverse flux periods are calculated for each node and energy group as

$$\omega_g^{k,n} = \frac{1}{\Delta t_{n-1}} \ln \left(\frac{\phi_g^{k,n}}{\phi_g^{k,n-1}} \right). \quad (3.126)$$

At the beginning of the current time step n , after the calculation of the fixed part of the fission and precursor source moments, the temporal extrapolation is applied for each node, energy group and flux moment as

$$\phi_g^{k,n+1,(0)} = \phi_g^{k,n} \exp(\omega_g^{k,n} \Delta t_n) \quad (3.127)$$

to obtain a first guess for the flux resulting from the fission source iteration of the current time step. The extrapolation is also applied to the outgoing partial currents of the node in a similar way. Ref. [29] mentions a similar methodology to be used in DIF3D-K, and for example an equivalent method in the case of a discrete ordinates transport code is more thoroughly described in Ref. [35].

3.4.4 Asymptotic extrapolation

The outer iterations are accelerated with an asymptotic source extrapolation method [28]. It is based on the assumption that the fission source solution $\psi^{(i)}$ of

iteration i converges to the exact solution $\boldsymbol{\psi}^{(\infty)}$ with an asymptotic behavior as

$$\boldsymbol{\psi}^{(\infty)} = \boldsymbol{\psi}^{(i)} + \mathbf{R}\sigma^i. \quad (3.128)$$

Here the global solution vector $\boldsymbol{\psi}$ contains all the node average fission sources of the problem, \mathbf{R} is an unknown vector, and σ is the dominance ratio. The dominance ratio of iteration i can be estimated as

$$\tilde{\sigma}^{(i)} = \frac{\|\boldsymbol{\psi}^{(i)} - \boldsymbol{\psi}^{(i-1)}\|_2}{\|\boldsymbol{\psi}^{(i-1)} - \boldsymbol{\psi}^{(i-2)}\|_2}, \quad (3.129)$$

where $\|\cdot\|_2$ denotes the L_2 vector norm. If an asymptotic behavior is observed, an improved estimate for the fission source moment vector of each node is given by

$$\hat{\boldsymbol{\psi}}^{k,(i)} = \boldsymbol{\psi}^{k,(i)} + \omega^{(i)}(\boldsymbol{\psi}^{k,(i)} - \boldsymbol{\psi}^{k,(i-1)}) \quad (3.130)$$

and for the outgoing partial currents of each node by

$$\hat{\mathbf{J}}_g^{\text{out},k,(i)} = \mathbf{J}_g^{\text{out},k,(i)} + \omega^{(i)}(\mathbf{J}_g^{\text{out},k,(i)} - \mathbf{J}_g^{\text{out},k,(i-1)}), \quad (3.131)$$

where

$$\omega^{(i)} = \frac{\tilde{\sigma}^{(i)}}{1 - \tilde{\sigma}^{(i-1)}}. \quad (3.132)$$

The asymptotic source extrapolation is only applied if an asymptotic behavior is observed. The observation is determined by the satisfaction of the following criterion

$$\min(\varepsilon_\omega^{(i)}, \varepsilon_\omega^{(i-1)}) < 0.1, \quad (3.133)$$

where

$$\varepsilon_\omega^{(i)} = \left| \frac{\omega^{(i)} - \omega^{(i-1)}}{\omega^{(i)}} \right|. \quad (3.134)$$

In addition, at least five outer iterations have to be performed between two successive asymptotic source extrapolations.

3.4.5 Steady state calculation solution algorithm

The steady state k -eigenvalue problem solution algorithm, based on a fission source iteration process, is presented in Fig. 3.4. The solution algorithm is based on the work done in Refs. [5, 6] with some clarifications and corrections. It is similar to the transient calculation algorithm presented in Fig. 3.2, but there are of course several fundamental differences between the algorithms. In the steady state algorithm, the problem is forced into a steady state by scaling the fission sources, and the group source terms include only the prompt fission source and in-scattering. Therefore,

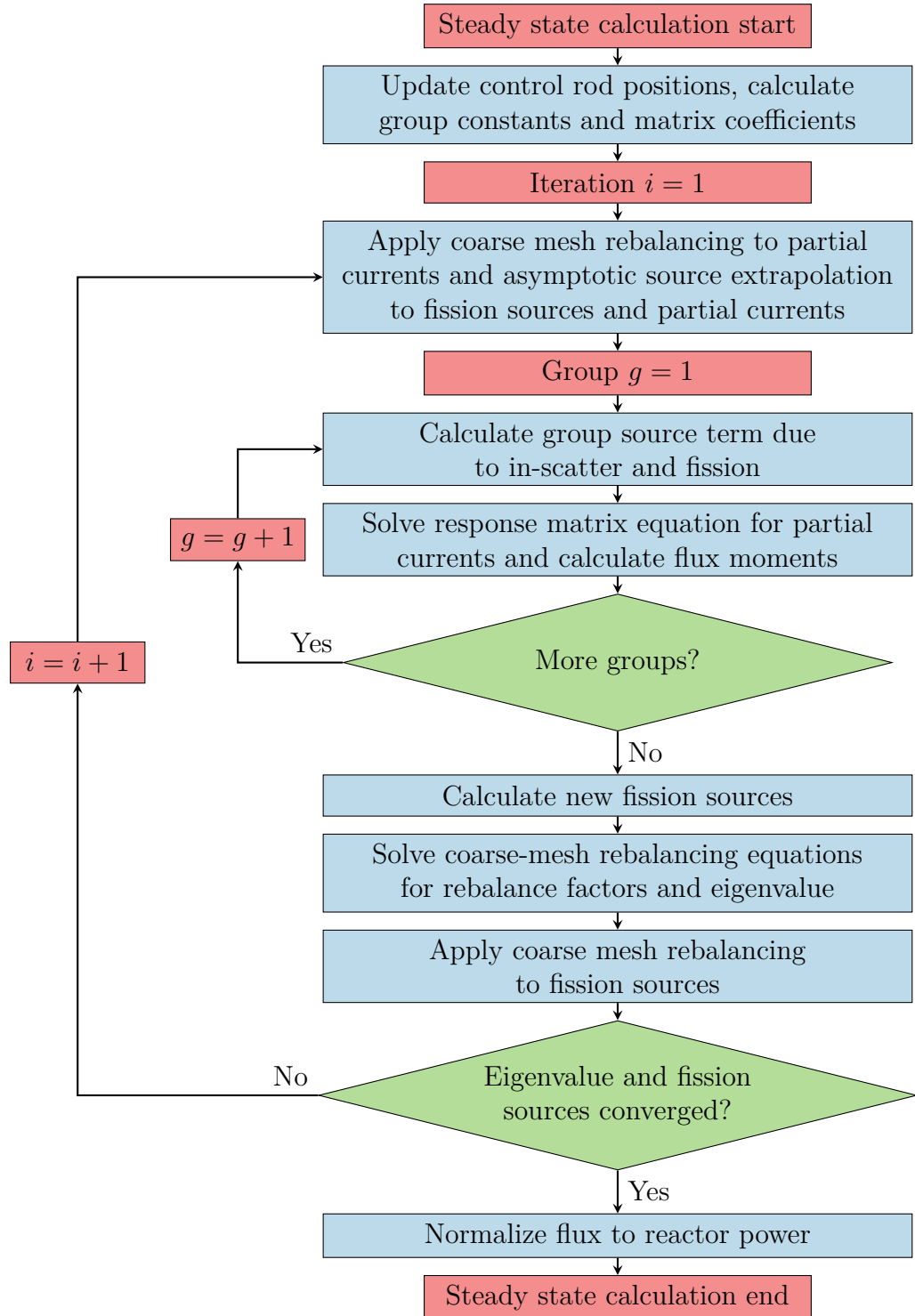


Figure 3.4: Solution algorithm of a steady state calculation.

all the time derivatives are set to zero along with β_m and C_m in all the equations, and the prompt fission source is divided by the effective multiplication factor k_{eff} , which needs to be solved together with the fluxes. The temporal extrapolation method is not used. Instead, the steady state method utilizes a coarse mesh rebalancing acceleration method. Together with the coarse mesh equations, the effective multiplication factor k_{eff} is solved using Wielandt method.

An additional convergence criterion to those defined in Eqs. (3.121) and (3.122) is used for the eigenvalue, defined as

$$|k_{\text{eff}}^{(i)} - k_{\text{eff}}^{(i-1)}| < \varepsilon_1, \quad (3.135)$$

where ε_1 is a user given convergence criterion parameter. In addition, at the end of a steady state calculation, the neutron flux is normalized to a user given power level of the reactor. The steady state solution and coarse mesh rebalancing methods are described in detail in Refs. [5, 28].

3.5 Model implementation into Apros

The Apros multigroup nodal diffusion model is mostly written in Fortran 2003 using an object-oriented programming like approach with derived types separating the model to reactor, assembly and node objects. Some state and configuration variables are stored in the Apros real time database to allow for example the saving of the simulation state and the continuation of the simulation later. Consequently, as Apros is mainly programmed in FORTRAN 77, an explicit interface between the model and rest of Apros is needed. In addition to the interface, some parts of the multigroup model program code related to geometry definitions and connections between the neutronics and thermal hydraulics are also written in FORTRAN 77. These parts are partly based on the program code of the other three-dimensional neutronics models of Apros.

3.5.1 Code structure

As usual with Apros components, the model program code is divided into initialization, known as preparation in Apros, and simulation phases. A model preparation subroutine is called during the preparation of the Apros model, if a multigroup model neutronics component is present in the Apros database. During the model preparation, the reactor geometry and connections between the neutronics and thermal hydraulics solutions are created, and the model is initialized. In the initialization, for example the group constants are read from a separate file, all the required memory allocations are performed and the variables present in the Apros database are linked to the model.

During the simulation phase of Apros, the thermal hydraulic solution of a time step is calculated before calling the neutronics simulation subroutine. If a multigroup model neutronics component is present in the Apros database, a model simulation subroutine is called. In this simulation phase of the model, the thermal hydraulic state of the reactor, including the coolant and fuel temperatures of the nodes, is first updated to be used in the neutronics solution. Then a time step is simulated either using the transient or the steady state algorithm presented in Figs. 3.2 and 3.4, respectively, depending on the user input. After the model is simulated for a time step, the nodewise powers of the reactor are updated to the thermal hydraulics solver.

In addition to the division of the code to preparation and simulation phases, the simulation part of the model program code is divided hierarchically into Fortran modules. The reactor module handles the simulation algorithm, the assembly module is mostly present to obtain a clear structure for the program code and most of the actual calculations are performed in the node module. The code also has other modules, such as the material module, which handles the interpolation of group constants between different burnups and the burnup module, which handles the storage of group constants.

3.5.2 Reactor model creation

The creation of the multigroup reactor model in Apros consists of adding suitable modules to Apros database and defining appropriate parameters for them to obtain the desired problem description. Usually, the thermal hydraulics solution is added first and simulated until the solution is converged to an equilibrium state. In the equilibrium, for example the temperatures of thermal hydraulic nodes are constants. Afterwards, the neutronics solution is added.

The thermal hydraulics solution is created by adding `REACTOR_CHANNEL` modules for each fuel and control assembly to simulate their flow channels. The `REACTOR_CHANNEL` should then be connected to other suitable thermal hydraulics modules to obtain proper boundary conditions or to simulate a larger part of a nuclear power plant.

The fuel and control assemblies are created by adding `NUCLEAR_ELEMENT_3DM` and `NUCLEAR_CONTROL_ELEMENTM` modules, respectively. The names of the corresponding `REACTOR_CHANNEL` modules are given to the assembly modules to connect the thermal hydraulics and neutronics solutions. The heights of the neutronics nodes are obtained from the thermal hydraulic node heights of a `REACTOR_CHANNEL` module. The main parameters of the assembly modules are the materials and burnups of the neutronics nodes and the location of the assembly in the reactor coordinates. The difference between the fuel and control assembly

modules is that the control assembly contains a movable absorber part, defined with one material and burnup.

The actual neutronics solution capability is created by adding a NUCLEAR__ REACTOR__3M module, in which most of the calculation parameters are defined. The group constants for the different materials are defined in a separate file. Currently the boundary conditions for the reactor can be set by specifying the boundary condition coefficient defined in Eq. (3.31) separately for the axial and radial directions. Same coefficients are used for each energy group.

3.5.3 Group constant and control assembly descriptions

The group constants used in the calculations are read from a group constant file where they are stored materialwise. A processing script can be used to produce group constant files from Serpent group constant calculation outputs. The current group constant model of the model is parametrized using independent quadratic fits for changes caused by coolant and fuel temperatures around their reference values. The fit can be written for a group constant Σ as

$$\begin{aligned} \Sigma = \Sigma^{\text{ref}} &+ c_{1,\text{cool}}(T_{\text{cool}}^{\text{ref}} - T_{\text{cool}}) + c_{2,\text{cool}}(T_{\text{cool}}^{\text{ref}} - T_{\text{cool}})^2 \\ &+ c_{1,\text{fuel}}(T_{\text{fuel}}^{\text{ref}} - T_{\text{fuel}}) + c_{2,\text{fuel}}(T_{\text{fuel}}^{\text{ref}} - T_{\text{fuel}})^2, \end{aligned} \quad (3.136)$$

where c_1 and c_2 are the linear and quadratic fit coefficients, respectively. In addition to the temperature fits, the group constants can be defined at multiple burnup values for each material. Linear interpolation is used between the burnup values and the group constants are treated as constants without extrapolation outside the minimum and maximum burnup values.

The following group constants are used for each energy group $g = 1 \dots G$:

- Absorption cross section $\Sigma_{a,g}$
- Diffusion coefficient D_g
- Energy release cross section $\kappa\Sigma_{f,g}$
- Fission yield χ_g
- Neutron production cross section $\nu\Sigma_{f,g}$
- Neutron velocity v_g ,

the following for each energy group $g' = 1 \dots G$ and $g = 1 \dots G$:

- Scattering cross section $\Sigma_{s,g' \rightarrow g}$

and the following for each precursor group $m = 1 \dots M$:

- Effective delayed neutron fraction β_m
- Precursor decay constant λ_m .

A single effective fission yield is currently used instead of the separate prompt and delayed neutron fission yields. The group removal cross section is calculated with

$$\Sigma_{r,g} = \Sigma_{a,g} + \sum_{g'=g+1}^G \Sigma_{s,g \rightarrow g'}. \quad (3.137)$$

The scattering cross section entry contains all the G^2 combinations of groups g' and g . The absorption and scattering cross sections can be replaced with the reduced absorption and scattering production cross sections to take into account the neutron multiplication reactions, such as (n, 2n) reactions [26]. The energy release cross section gives the power density inside a node when multiplied by the average neutron flux inside the node. Depending on the user input, the power can include only the fission power or the sum of the fission power and the power produced in capture reactions taken into account by multiplying the fission power with a correction factor. The unit of the energy release cross section is J/cm. All length and time units of the group constants are in cm and s, respectively.

The control assembly absorbers parts, referred to as control rods from now on, are modeled as cluster control rods consisting of a single material. If a control rod is fully inside a node, the node material is replaced by the control rod material. If the control rod is only partially inside a node, the group constants of the node are volume weighted as

$$\Sigma = (1 - f)\Sigma^{\text{noCR}} + f\Sigma^{\text{CR}}, \quad (3.138)$$

where f is the fraction of the node containing the control rod, and noCR and CR refer to the material of the control assembly node and the material of the control rod, respectively. If the control assembly node or control rod contain material with nonzero fission production cross section, the fission yields, effective delayed neutron fractions and precursor decay constants are weighted with the volume weighted fission production cross sections of the control assembly node and control rod materials.

Chapter 4

Model validation cases

In order to validate the implementation and accuracy of a simulation computer code, the simulation results of the code have to be compared either with measured values of real world benchmark cases or with simulation results of other computer codes. The latter needs to be especially considered if suitable real world benchmarks are not available.

In this chapter, first the reactor core models used in the validation of the Apros multigroup nodal diffusion model are presented. Second, the group constant generation process for the different nodal diffusion codes used in the validation is described. Finally, the different steady state and transient simulation cases used in the validation are presented.

4.1 Reactor core models

The reactor core models used for the Apros steady state model validation were chosen from the OECD/NEA Sodium-cooled fast reactor (SFR) Benchmark Task Force benchmark definitions [20]. The benchmark is one step of a stepwise analysis of feedback and transient behavior of the generation IV sodium-cooled fast reactors. The chosen core models are the large size oxide fuel core, referred to as large oxide core from now on, and the medium size metallic fuel core, referred to as medium metallic core from now on.

By varying different parameters between the simulations, different sources of errors and differences can be found, especially if the parameters are varied independently. Therefore, the two different sized cores can be used to study the accuracy of the model with different sized cores, and the different outer boundary geometries of the two cores can be used in the validation of the model implementation. The cores are also simulated with varying control rod positions to determine their effect on the simulation accuracy.

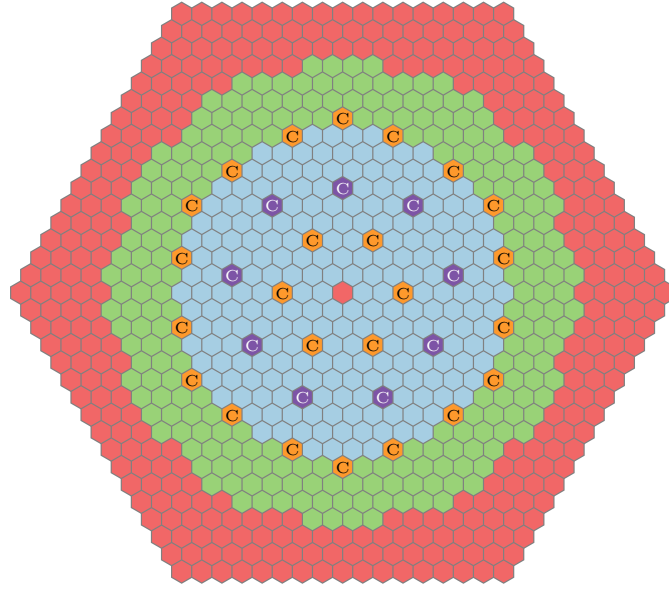


Figure 4.1: Radial layout of the large oxide sodium fast reactor core. Blue and green represent the inner and outer core zone fuel assemblies, respectively, orange and purple represent the primary and secondary control assemblies, respectively, and red represents the radial reflector and the middle assembly.

The benchmark specifications include accurate descriptions of the core geometries and nuclide compositions of the different materials for steady state neutronics studies. For example, the outer core zone fuel has a higher plutonium content than the inner core zone fuel in both cores. No thermal hydraulic data is included, and all the temperatures of the nuclear fuel, and coolant and structure materials are assumed to be constants, corresponding to full power values. The specifications of transient analysis, being one step of the benchmark, were not yet available. As no other liquid metal cooled fast reactor transient benchmark specifications were available, the same core models are also used in the Apros transient model validation.

The radial layout of the large oxide core is presented in Fig. 4.1. It consists of 453 fuel, 330 radial reflector, 33 control and 1 middle assemblies. The inner and outer core zones consist of 225 and 228 fuel assemblies, respectively. The control assemblies are divided into two independent groups. The primary control system consists of 6 and 18 control assemblies in the inner core zone and at the interface between the inner and outer core zones, respectively. The secondary control system consists of 9 control assemblies in the inner core zone. The core has a 120° periodic symmetry.

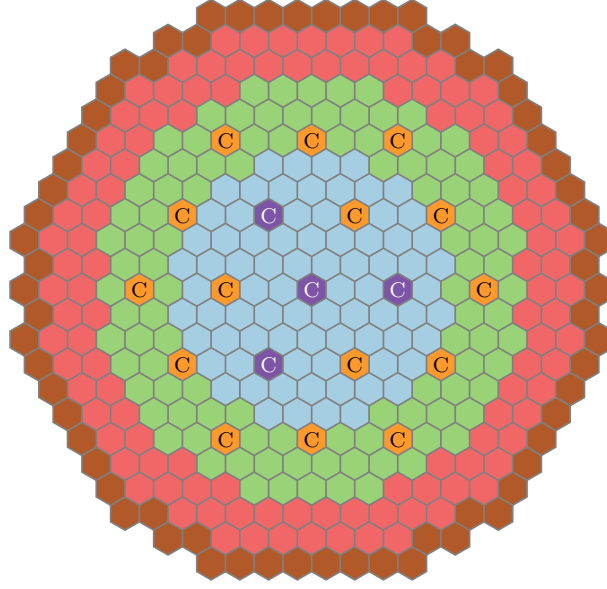


Figure 4.2: Radial layout of the medium metallic fuel sodium fast reactor core. Blue and green represent the inner and outer core zone fuel assemblies, respectively, orange and purple represent the primary and secondary control assemblies, respectively, red represents the radial reflector, and brown represents the radial shield.

The radial layout of the medium metallic core is presented in Fig. 4.2. It consists of 180 fuel, 114 radial reflector, 66 radial shield and 19 control assemblies. The inner and outer core zones consist of 78 and 102 fuel assemblies, respectively. The control assemblies are divided into two independent groups. The primary control system consists of 3 and 12 control assemblies in the inner and outer core zones, respectively, and the secondary control system consists of 4 control assemblies in the inner core zone. The core has a 120° periodic symmetry.

The fuel assemblies of the large oxide core consist from bottom to top of the lower gas plenum, lower axial reflector, active core consisting of five different axial fuel regions, upper gas plenum and upper axial reflector. The control rod materials of the primary and secondary control assemblies consist of natural and ^{10}B -enriched B_4C , respectively. The fuel assemblies of the medium metallic core consist from bottom to top of the lower structure, lower reflector, active core consisting of five different axial fuel regions, replace sodium, gas plenum and upper structure. The control rod materials of the primary and secondary control assemblies consist of ^{10}B -enriched B_4C . In both cores, the parts of the control assemblies without control rods are filled with sodium.

4.2. GROUP CONSTANTS

Table 4.1: Nominal operating conditions of the large oxide and medium metallic sodium fast reactor cores. [20]

Property	Large oxide	Medium metallic
Reactor power (MW)	3600.0	1000.0
Average control rod, coolant and structure temperature (°C)	470.0	432.5
Average fuel temperature (°C)	1227.0	534.0

Table 4.2: Nominal operating dimensions of the large oxide and medium metallic sodium fast reactor core assemblies. [20]

Property	Large oxide (cm)	Medium metallic (cm)
Length	311.1624	480.2000
Active core length	100.5660	85.8200
Control rod length	100.5660	86.7500
Pitch	21.2205	16.2471
Outer duct flat-to-flat distance	20.7468	15.8123

The nominal operating conditions of the cores are presented in Tab. 4.1 and the main nominal operating dimensions of their assemblies in Tab. 4.2. According to the benchmark specifications, both cores are modeled using vacuum boundary conditions.

For the nodal diffusion calculations, the different axial regions of the assemblies are divided into nodes as follows. In the case of the large oxide core, the lower gas plenum, lower axial reflector, each different axial fuel region, upper gas plenum and upper axial reflector are divided into 4, 2, 2, 2 and 4 axial nodes. In the case of the medium metallic core, the lower structure, lower reflector, each different axial fuel region, replace sodium, gas plenum and upper structure are divided into 2, 6, 2, 2, 4 and 6 axial nodes. The nodes inside each axial region are of equal height with each other.

4.2 Group constants

The group constants utilized in the different nodal diffusion codes are generated with Serpent 2 using JEFF-3.1.1 evaluated data library [8]. The basis of the used energy group structure is the ECCO-33 energy group structure designed for fast reactor calculations. However, as the statistical uncertainties of the lowest energy groups tend to be high, the energy groups 25–33 are merged into group 24. The

4.2. GROUP CONSTANTS

Table 4.3: Applied 24-group structure obtained from the ECCO-33 group structure by merging the lowest energy groups into one group. The last energy without a group number is the lower energy limit of group 24.

Group number	Upper energy limit (MeV)	Group number	Upper energy limit (MeV)
1	1.96403×10^1	14	2.47875×10^{-2}
2	1.00000×10^1	15	1.50344×10^{-2}
3	6.06531	16	9.11882×10^{-3}
4	3.67879	17	5.53084×10^{-3}
5	2.23130	18	3.35463×10^{-3}
6	1.35335	19	2.03468×10^{-3}
7	8.20850×10^{-1}	20	1.23410×10^{-3}
8	4.97871×10^{-1}	21	7.48518×10^{-4}
9	3.01974×10^{-1}	22	4.53999×10^{-4}
10	1.83156×10^{-1}	23	3.04325×10^{-4}
11	1.11090×10^{-1}	24	1.48625×10^{-4}
12	6.73795×10^{-2}		1.00001×10^{-11}
13	4.08677×10^{-2}		

high uncertainties are due to the small probability of a fission neutron born at a high energy to slow down to thermal energies inside a fast reactor. [36, 37] The resulting energy group structure is presented in Tab. 4.3. In addition to the few-group structure, Serpent uses a finer microgroup structure to calculate the homogenized diffusion constants. In Ref. [38] it was shown that the accuracy of a one-group diffusion coefficient is dependent on the number of microgroups used during the calculations. Therefore, the ECCO-1968 group structure is used as the microgroup structure during the group constant generation calculations.

The group constants for the multiplying fuel regions are generated in an infinite two-dimensional lattice consisting of the single fuel assembly with periodic boundary conditions. For non-multiplying regions, such as the gas plenums and control rods, the group constants are generated in an infinite two-dimensional lattice consisting of the non-multiplying region surrounded by six half-assemblies of the closest fuel region and with periodic boundary conditions. The homogenization is performed only over the corresponding non-multiplying region. The closest fuel region is determined by the radial and axial location of the non-multiplying region inside the core. As a control rod in the Apros model consists of only one material type, the group constants of all control rods are generated by surrounding them with only the axially middlemost fuel regions.



Figure 4.3: The two-dimensional geometry for the group constant generation of the outermost fuel assemblies of the large oxide core taking into account the softening of the neutron energy spectrum in the radial reflector. Brown represents the outer core zone fuel assemblies for which the group constants are generated for, green represents the other outer core zone fuel assemblies and red represents the radial reflector. The heterogeneous fine structures of the assemblies are not shown here.

For the large oxide core, the effect of neutron energy spectrum softening in the radial reflector is also taken into account using a two-dimensional radial reflector model to generate the group constants for the fuel regions adjacent to the radial reflector. This is performed using the geometry presented in Fig. 4.3, as described in Ref. [39]. The homogenization is performed only over the outermost fuel assemblies. Reflective boundary conditions are used on the top and bottom sides and vacuum boundary conditions are used on the left and right sides of the geometry presented in the figure.

4.3 Steady state model implementation

The implementation of the Apros steady state model is validated against the nodal diffusion solver of DIF3D code, on which the Apros model is based. The simulated cases are the large oxide and medium metallic cores with all control rods withdrawn to the top of the active core, as defined in the benchmark specifications.

As the Apros model and DIF3D solver are based on the same solution method, an extensive proof of the correctness of the implementation can be obtained by comparing the average fluxes of each energy group in all the nodes of the modeled cores instead of comparing only the node- or assemblywise power distributions. Therefore, the compared simulation results of the codes are the effective multiplication factors and node- and groupwise average fluxes.

4.4 Steady state model accuracy

Serpent is a continuous-energy Monte Carlo code modeling the neutron interactions without major approximations. It is also used to generate the group constants used in the Apros simulations. Therefore, the results of Serpent full core calculations can be treated as accurate reference solutions for the Apros model. The validation of the Apros steady state model accuracy is performed by comparing the effective

multiplication factors, assembly power distributions obtained by integrating the nodewise powers axially and axial power distributions obtained by integrating the nodewise powers radially. As the geometries used in Serpent do not consist of homogeneous nodes in the way the Apros model geometries do, the equivalent node powers are obtained by tallying the respective powers in the three-dimensional geometry regions corresponding to the Apros model nodes.

For both the large oxide and medium metallic cores the comparisons are performed with control rods fully withdrawn and fully inserted as specified in the benchmark specifications. However, typical transient simulation cases begin at critical states. Therefore, an additional comparison is made with the large oxide core. In this case, the lower ends of the control rods, referred to as control rod positions from now on, are at the interface between the axial fuel levels bringing the core closest to criticality with Apros. The control rods are not at the exact critical position of either code due to additional error caused by the rod cusping effect¹. Additionally, the effect of asymmetrically positioned control rods are studied briefly with the large oxide core by simulating cases where all but one control rod are fully withdrawn and the one control rod is fully inserted, and vice versa.

Furthermore, the effect of the number of energy groups is shortly studied with the large oxide core by combining adjacent energy groups of the 24-group structure to define 12- and 6-group structures. The comparison is only approximate, as the energy group division is not based on the importance of the different energy intervals in sodium fast reactors. The motivation for studying the required number of energy groups needed to obtain suitable simulation accuracy is the dependence of the computational costs of the simulations on the number of used energy groups.

4.5 Transient model

The validation of the Apros transient model is performed against the DYN3D reactor dynamics code using the HEXNEM1 method which is to some extent similar to the method used in the Apros model. A comparison against the DIF3D-K kinetics code, on which the Apros model is based on, was not performed, as the code was not available. The validation of the implementation and the similarity of the simulation results is performed by comparing the total core power time evolution and the power distributions at the beginning and end of the simulations. As the thermal hydraulics is not simulated, the validation is purely a neutron kinetics validation without any feedback effects present.

¹Control rod cusping is the overestimation of the control rod worth caused by the volume weighting instead of the correct flux weighting of the group constants inside a node with a partially inserted control rod.

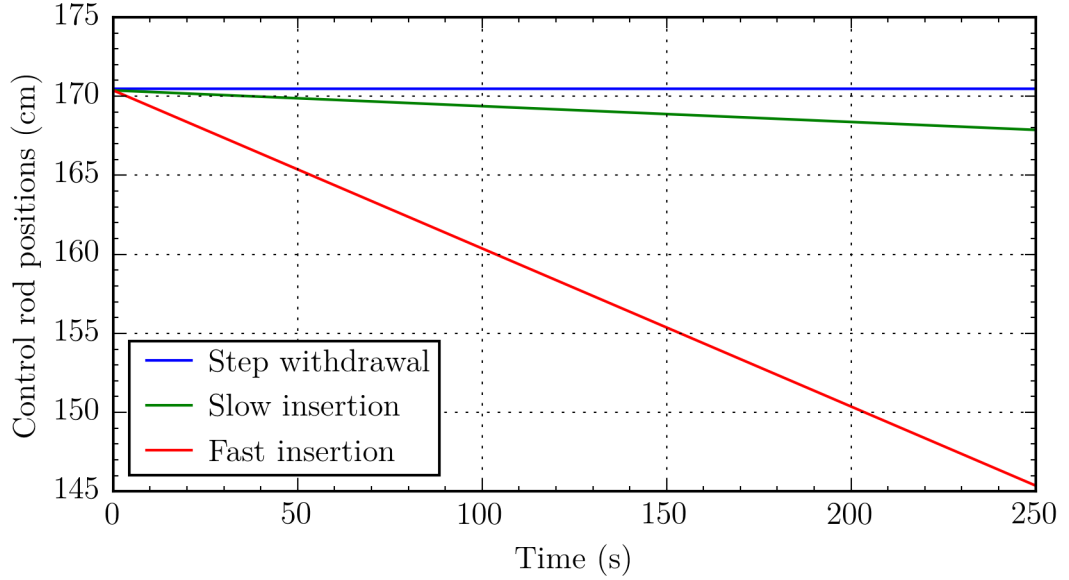


Figure 4.4: Control rod positions defined from the bottom of the core during the different transient simulation cases. The interface between the fifth and sixth axial fuel levels of the active core is at the height of 170.3703 cm from the bottom of the core.

Three transient comparisons are made with the large oxide core. Unlike the core description presented in Sec. 4.1, the control rods are now modeled as half-infinite, beginning from the bottom of the control rod and extending to the top of the reactor core. The half-infinite rods are used instead of finite length rods due to the control rod modeling limitations of DYN3D. Each case is initialized with a steady state calculation where the control rods are at the interface between the fifth and sixth axial fuel levels of the active core. Even though both codes use similar group constant volume weighting for the nodes with partially inserted control rods, the control rod initial positions were not the exact critical positions of either code. This was chosen due to the fact that the critical positions would not be the same for both codes, and additionally to minimize the possible differences at the initial states caused by the different axial flux descriptions of the codes.

The first simulation case is a slow insertion of 0.1 mm/s of all control rods. The second case is a fast insertion of 1 mm/s of all control rods. The final case is a very fast step 1 mm withdrawal of all control rods during 0.05 s. The control rod movements of all three cases are shown in Fig. 4.4. The end of time step control rod positions are used during the simulations.

Chapter 5

Results

In this chapter, the results of the validation calculations described in the previous chapter are presented. First, the steady state simulation results of the Apros steady state model and DIF3D are compared in order to determine whether the model is correctly implemented. Second, the simulation results of the Apros steady state model and Serpent are compared to define the accuracy of the model. Finally, the transient simulation results between the Apros transient model and DYN3D are compared to determine the correctness of the model implementation and the similarity of the simulation results compared with DYN3D.

5.1 Steady state model implementation

The large oxide and medium metallic cores were simulated with Apros and DIF3D with all control rods withdrawn from the cores. The group constants for both cores were calculated with Serpent using 100 inactive and 1000 active cycles with 10^5 neutrons per cycle. For the large oxide core, the group constants taking into account the neutron energy spectrum softening in the radial reflector were used.

The group constant input of DIF3D is different from that of Apros, as for example the average number of neutrons released per fission $\bar{\nu}_g$, fission cross section $\Sigma_{f,g}$ and transport cross section $\Sigma_{tr,g}$ are used instead of the fission production cross section $\nu\Sigma_{f,g}$ and diffusion coefficient D_g used by Apros. Therefore, a special version of the Apros steady state model with group constant definitions equivalent to DIF3D was used to avoid differences caused by a different input accuracy of the group constants. Very tight convergence criteria $\varepsilon_1 = 10^{-12}$ for the multiplication factor and $\varepsilon_2 = 10^{-10}$ for the pointwise fission source were used to determine the true differences between the codes. The default convergence criterion $\varepsilon_3 = 10^{-5}$ for the average fission source was used.

5.1. STEADY STATE MODEL IMPLEMENTATION

Table 5.1: Effective multiplication factors of the large oxide and medium metallic cores calculated with Apros and DIF3D with all control rods withdrawn.

Core	Apros	DIF3D	Difference (10^{-8} %)
Large oxide	1.0299370373	1.0299370412	-37.72
Medium metallic	1.0343758429	1.0343758431	-1.48

The resulting multiplication factors calculated with both codes are presented in Tab. 5.1. Histograms of the relative differences of the node- and groupwise average fluxes between the codes are shown in Figs. 5.1 and 5.2 for the large oxide and medium metallic cores, respectively. As described in Sec. 4.1, the large oxide and medium metallic core models consist of 17974 and 11370 nodes, respectively. Therefore, as 24 energy groups are used, the total numbers of the compared fluxes are 431376 and 272880. The average flux output of DIF3D uses only six significant figures. Therefore, some very small errors are present in the differences.

In the case of the large oxide core, 99.04 % of the flux differences are within 0.00025 % between the codes and 99.99 % of the differences are within 0.00125 % between the codes. All the fluxes with the absolute values of differences higher than 0.005 % are negative. The absolute values of the negative fluxes are small or very small compared with the mean fluxes at their respective axial levels and energy groups. The negative fluxes are usually present in groups with high energies and never in nodes with nonzero fission sources. As the negative average fluxes are present with both codes, they are not an implementation error in the Apros model. Although the physical neutron flux can never be negative, non-negativity is not computationally enforced as there is no particular reason for it. The negative fluxes are probably caused by the nodal formulation of the diffusion theory, and they do not cause any computational problems.

In the case of the medium metallic core, 99.21 % of the flux differences are within 0.00025 % between the codes and 99.91 % are within 0.00125 % between the codes. Almost all the fluxes with the absolute values of differences higher than 0.01 % are negative, and all the absolute values of the fluxes are small or very small compared with the mean fluxes at their respective axial levels and energy groups. All the fluxes with the absolute values of differences between 0.005 % and 0.01 % are small compared with the mean flux at their respective axial level and energy group. In addition, the respective mean flux is very small.

To conclude, the multiplication factor differences of both simulated cases are extremely small between the codes. In addition, almost all the flux differences are within 0.00125 % between the codes, and all the differences are within 0.07 % between the codes. This can be considered an excellent agreement and for exam-

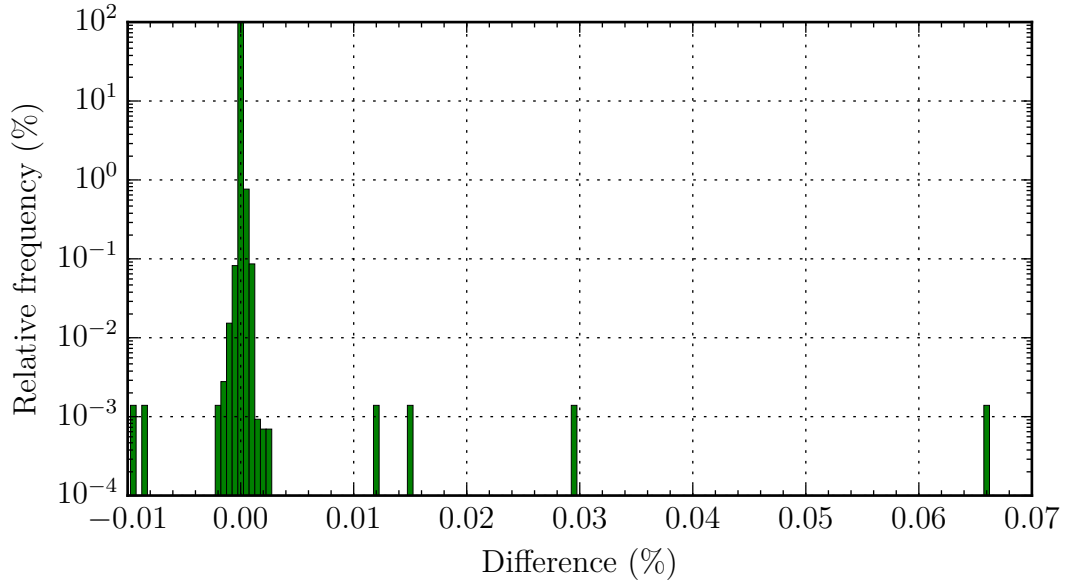


Figure 5.1: The relative frequencies of the differences between Apros and DIF3D average fluxes in each energy group and node of the large oxide core. The histogram bin width is 0.0005 percentage points.

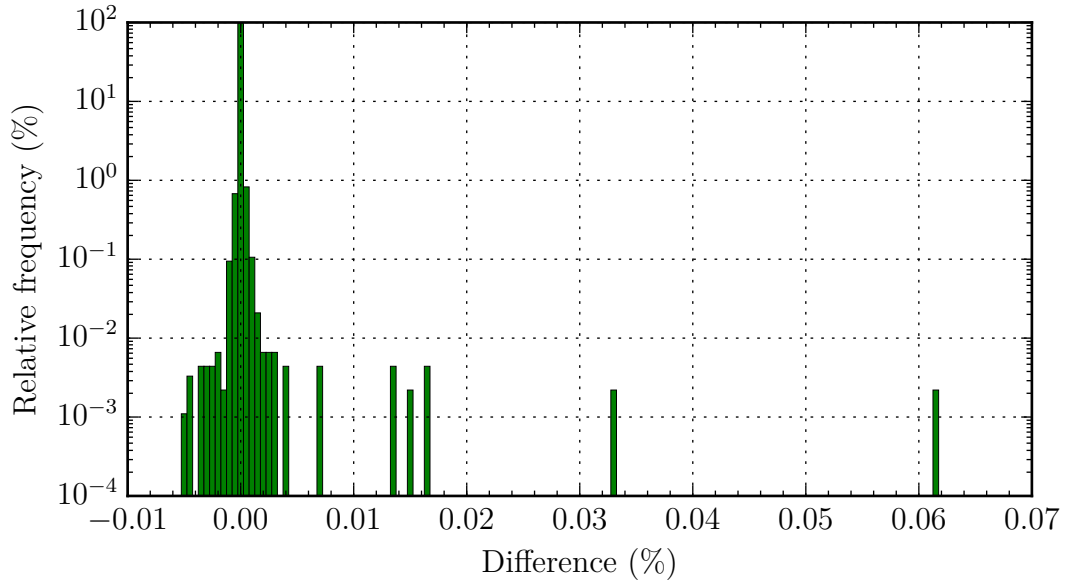


Figure 5.2: The relative frequencies of the differences between Apros and DIF3D average fluxes in each energy group and node of the medium metallic core. The histogram bin width is 0.0005 percentage points.

ple the differences in the program structures of the codes can be accounted for the differences. Therefore, the steady state model of the Apros multigroup nodal diffusion model has been implemented correctly and it is reasonable to proceed to validating its accuracy.

5.2 Steady state model accuracy

The group constants for both the large oxide and medium metallic cores were generated with Serpent using 100 inactive and 1000 active cycles with 10^5 neutrons per cycle. The Serpent full core calculations were mostly performed with 500 inactive and 20000 active cycles with 10^6 neutrons per cycle. For the 120° periodic symmetric cases of the large oxide core the number of inactive cycles was increased to 1000 in order to improve the periodic symmetries of the fission source distributions.

Before comparing the Apros and Serpent results, the Serpent radial power distributions were averaged over the 120° periodic symmetry. This was done to reduce the effect of the statistical variation of the fission source on the comparisons between the codes. Although this procedure reduces the maximum differences between the codes, it is justifiable, as the fission source distribution should be symmetric in Serpent and the asymmetry is only caused by statistical variation. The effect of the symmetrization is not huge, as the number of simulated neutron histories is relatively high. Another way to reduce the asymmetry of the fission source distribution would be to calculate multiple Serpent full core calculations and average the distribution over the calculations. This method could also be utilized to improve the fission source distributions of the asymmetric cases. However, with the used neutron population sizes the full core calculations already took a considerable amount of time and therefore this procedure was not performed.

All Apros calculations were performed with the default convergence criteria $\varepsilon_1 = 10^{-7}$ and $\varepsilon_2 = \varepsilon_3 = 10^{-5}$. As the geometry in Serpent does not consist of homogeneous nodes, the nodewise power distributions were obtained by tallying the powers in three-dimensional geometry regions corresponding to the Apros nodes.

In this section, uncertainty refers to one standard deviation relative statistical error of Serpent, min refers to the negative difference with the highest absolute value, only assemblies with nonzero powers are included in the calculation of the power difference means, and the means of the differences are calculated from the absolute values of the differences.

5.2.1 Medium metallic core

The medium metallic core was simulated with all control rods withdrawn from the core, denoted as case 1, and all control rods inserted in the core, denoted as case 2.

5.2. STEADY STATE MODEL ACCURACY

Table 5.2: (a) Effective multiplication factors and (b) assembly and axial power differences of the medium metallic core calculated with Apros and Serpent. All control rods are either withdrawn from the core (case 1) or inserted in the core (case 2).

(a)						
Case	Apros	Serpent	Uncertainty (10^{-4} %)	Difference (%)		
1	1.03439	1.04520	5.2	−1.03		
2	0.85670	0.88001	5.9	−2.65		

(b)						
Case	Assembly power difference			Axial power difference		
	Max (%)	Min (%)	Mean (%)	Max (%)	Min (%)	Mean (%)
1	2.40	−2.10	1.60	0.53	−2.93	0.66
2	2.66	−2.99	1.53	0.68	−2.02	0.66

The resulting multiplication factors and assembly and axial power distribution differences are shown in Tab. 5.2. The assembly power distributions calculated with Serpent and their differences between Apros and Serpent are presented in Figs. 5.3 and 5.4 for the cases 1 and 2, respectively. The axial power distributions calculated with Apros and Serpent and their differences are presented in Fig. 5.5 for both cases.

Compared with the differences of the multiplication factors, the Serpent uncertainties are insignificant. The multiplication factors are underestimated in both cases and the differences are relatively high, especially in the case 2.

All the Serpent relative statistical errors of the node- and assemblywise powers are smaller than 3.1×10^{-2} % and 1.2×10^{-2} %, respectively, in the case 1, and smaller than 3.3×10^{-2} % and 1.4×10^{-2} %, respectively, in the case 2. Therefore, they are insignificant compared with the differences between the codes.

In both cases, the assembly powers are overestimated in the inner core zone and underestimated in the outer core zone. The highest assembly powers are underestimated, as they are located in the outer core zone. The highest assembly power differences are at the outermost fuel assemblies which have the lowest powers. When comparing the assembly power distributions between the cases, the insertion of the control rods somewhat increases the magnitude of the highest differences, whereas the mean difference slightly decreases. In addition, in the case 1, the assembly power differences are mostly dependent on the assembly distance from the center of the core, whereas in the case 2 the insertion of the control rods clearly

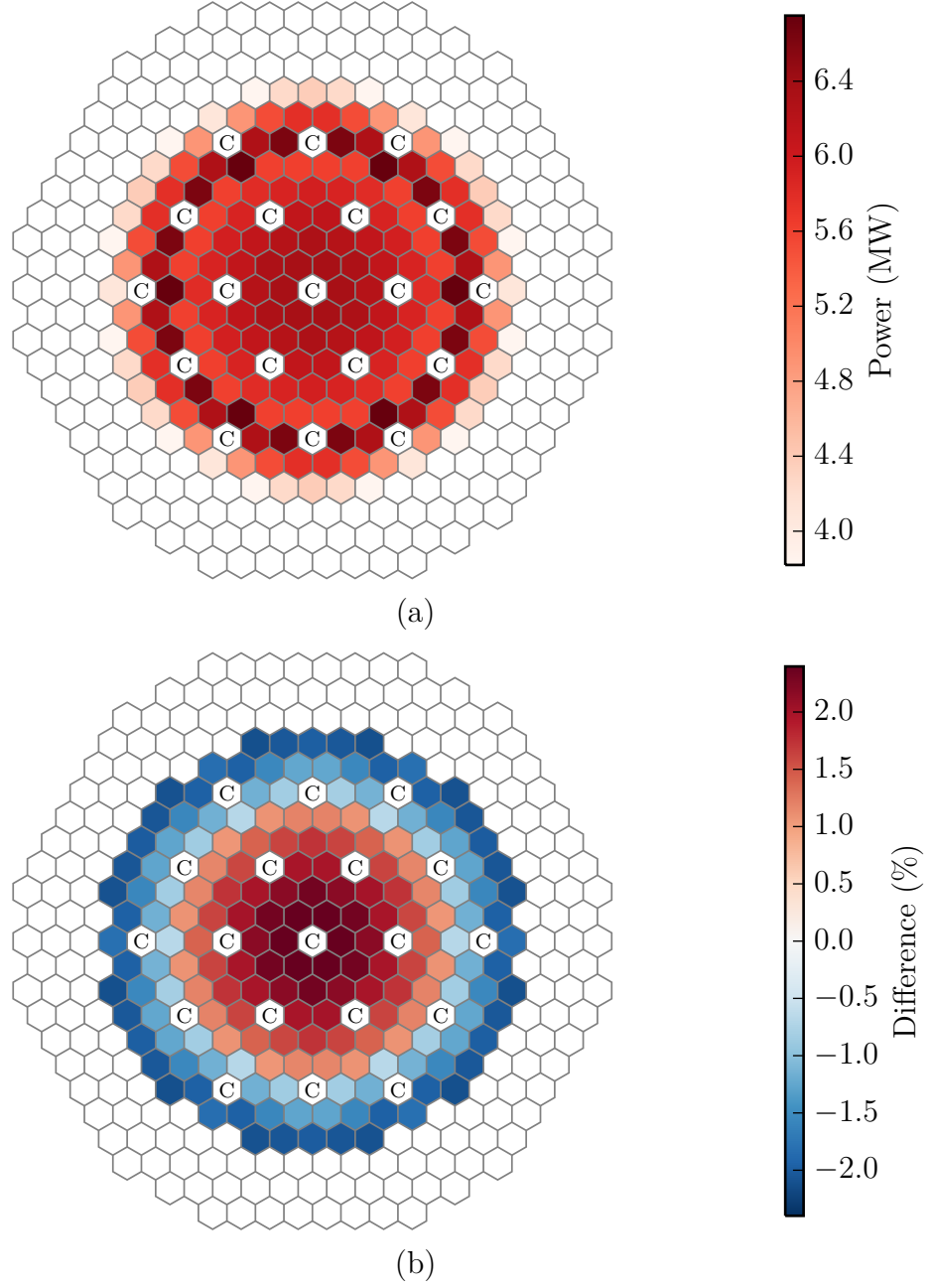


Figure 5.3: (a) Assembly powers calculated with Serpent and (b) assembly power differences between Apros and Serpent of the medium metallic core with all control rods withdrawn from the core (case 1). The maximum positive and negative differences are 2.40 % and -2.12 %, respectively.

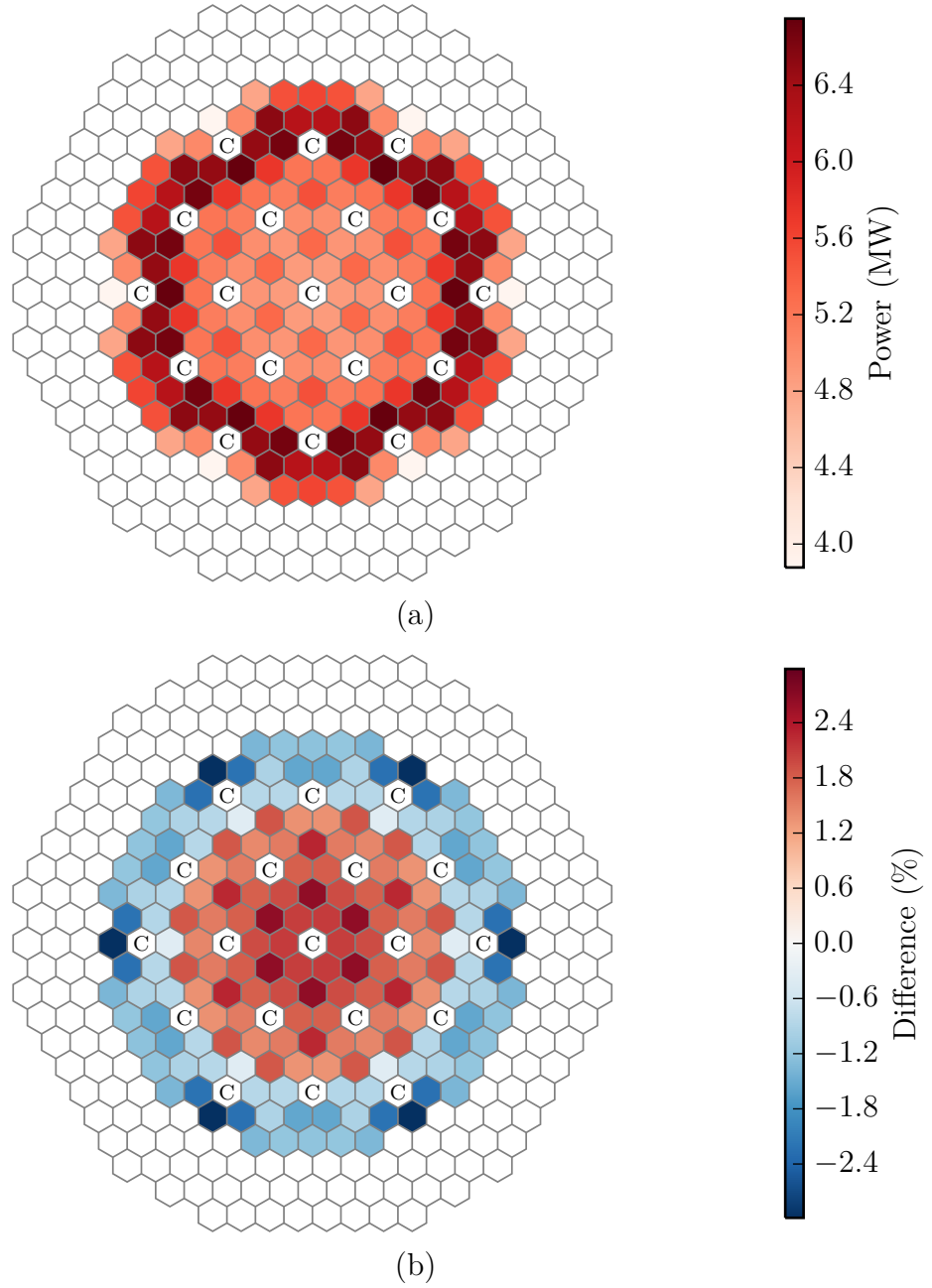
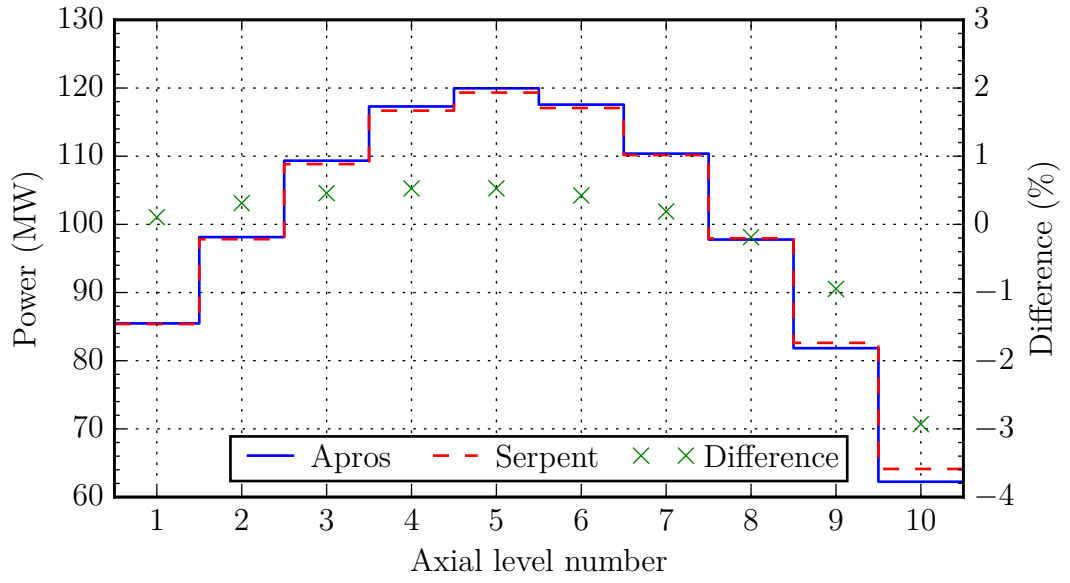
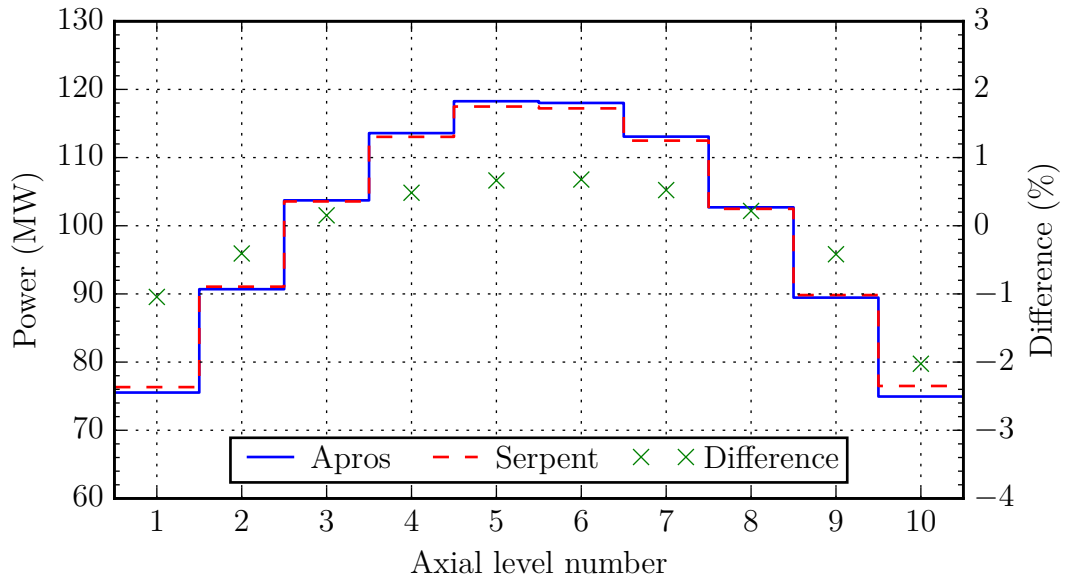


Figure 5.4: (a) Assembly powers calculated with Serpent and (b) assembly power differences between Apros and Serpent of the medium metallic core with all control rods inserted in the core (case 2). The maximum positive and negative differences are 2.66 % and -2.99 %, respectively.



(a)



(b)

Figure 5.5: Axial powers of the medium metallic core calculated with Apros and Serpent with all control rods (a) withdrawn from the core (case 1) and (b) inserted in the core (case 2).

distorts this pattern. In this case, the assembly powers in the assemblies next to the control rods tend to be underestimated compared with the other assemblies.

In the case 1, the axial powers are slightly overestimated in the lower part and underestimated in the upper part of the core. In the case 2, the axial powers are underestimated at the bottom and top of the core and slightly overestimated in the middle of the core. In both cases, the highest axial power is overestimated. The underestimation of the axial power at the top of the core in the case 1 might be caused by the presence of the control rods above the active core. In the case 2, the axial power difference pattern is more symmetric, but the uppermost axial power is underestimated more than the lowest. This might be caused by the control rod cusping effect, as the control rods partially extend to the nodes above the active core. This is due to the control rods being slightly longer than the active core. When comparing the axial power distributions, the insertion of control rods decreases the magnitude of the highest difference, whereas the mean difference stays the same.

The signs of the assembly power differences are related to the division of the core into the inner and outer zones with different fuel compositions. In addition, the control rods also alter the assembly power difference distribution locally. Therefore, the differences could probably be decreased by utilizing the radial discontinuity factors of the generalized equivalency theory [40]. The utilization of the discontinuity factors allows the surface averaged fluxes to be discontinuous across the nodal surface and can reduce the errors in assembly powers notably. In addition, as especially in the case 1 the outermost assemblies have high power differences, the differences might be decreased by generating the group constants for these assemblies taking into account the neutron energy spectrum softening in the radial reflector. However, this is studied only in the case of the large oxide core. The axial power differences might also be reduced by utilizing axial discontinuity factors as the core consists of axially nonuniform nodes [41]. However, this subject has not been thoroughly studied.

5.2.2 Large oxide core

5.2.2.1 Different group constant sets

The large oxide core was first simulated with all control rods withdrawn from the core, denoted as cases 1 and 2, and all control rods inserted in the core, denoted as cases 3 and 4. Cases 1 and 3 were simulated with Apros using the group constants without taking into account the neutron energy spectrum softening in the radial reflector, whereas cases 2 and 4 were simulated using the group constants where the spectrum softening is taken into account. The resulting multiplication factors and assembly and axial power distribution differences are shown in Tab. 5.3.

5.2. STEADY STATE MODEL ACCURACY

Table 5.3: (a) Effective multiplication factors and (b) assembly and axial power differences of the large oxide core calculated with Apros and Serpent. The control rods are either withdrawn from the core (cases 1 and 2) or inserted in the core (cases 3 and 4).

(a)						
Case	Apros	Serpent	Uncertainty (10^{-4} %)	Difference (%)		
1	1.02977	1.03404	4.2	−0.41		
2	1.02994	1.03404	4.2	−0.40		
3	0.97509	0.98161	4.6	−0.66		
4	0.97551	0.98161	4.6	−0.62		

(b)						
Case	Assembly power difference			Axial power difference		
	Max (%)	Min (%)	Mean (%)	Max (%)	Min (%)	Mean (%)
1	1.58	−3.21	1.09	0.71	−2.64	0.81
2	1.14	−1.79	0.63	0.70	−2.57	0.79
3	2.06	−2.77	0.98	1.04	−3.25	1.11
4	2.29	−3.88	1.31	1.01	−3.16	1.08

Compared with the differences of the multiplication factors, the Serpent uncertainties are insignificant. All the multiplication factors are underestimated. The differences are somewhat high although they are smaller than with the medium metallic core. The differences are smaller in the cases 2 and 4 when compared with the cases 1 and 3.

All the Serpent relative statistical errors of the node- and assemblywise powers are smaller than 6.6×10^{-2} % and 2.9×10^{-2} %, respectively, in the cases 1 and 2, and smaller than 6.2×10^{-2} % and 2.9×10^{-2} %, respectively, in the cases 3 and 4. Therefore, they are insignificant compared with the differences between the codes.

When the control rods are withdrawn, the assembly power differences are smaller in the case 2 than in the case 1. On the other hand, when the control rods are inserted, the assembly power differences are higher in the case 4 than in the case 3. However, the differences between the cases are smaller when the control rods are inserted than when they are withdrawn. In addition, as the multiplication factor and axial power differences are smaller when utilizing the group constants taking into account the neutron spectrum softening in the radial reflector, this group constant set is utilized for the rest of the calculations.

The assembly power distributions calculated with Serpent and their differences between Apros and Serpent are presented in Figs. 5.6 and 5.7 for the cases 2 and 4, respectively. The axial power distributions calculated with Apros and Serpent and their differences are presented in Fig. 5.8 for both cases.

In the case 2, the assembly powers are overestimated in the inner core, and mostly underestimated in the outer core. The highest differences are at the outermost fuel assemblies which have the lowest powers. The highest assembly powers are slightly over- or underestimated. In the case 4, the assembly powers are underestimated in the inner core and over- or underestimated in the outer core. The highest differences are in the middle of the core in low power assemblies next to control rods. The highest assembly powers are overestimated. When comparing the cases, the insertion of the control rods increases the magnitude of the assembly power differences.

In both cases, the axial powers are underestimated in the bottom and top of the core and overestimated in the middle of the core. The insertion of the control rods increases the magnitude of the differences. In both cases, the highest axial powers are overestimated.

Especially the case 2 shows the same behavior of the signs of the assembly power differences to be dependent on the division of the core to inner and outer zones as was with the medium metallic core. In the case 4, the highest differences are again in the assemblies next to the control rods. Therefore, the same observations about the radial discontinuity factors hold here.

The axial power differences increased when the control rods were inserted, even though the core became axially more symmetric. Therefore, the accuracy might be improved by improving the axial control rod modeling with for example specifying the control rod group constants axially. Additionally, the utilization of axial discontinuity factors might improve the simulation accuracy.

5.2.2.2 Additional cases

The large oxide core was also simulated with all control rods at the interface of axial levels which brings the reactor closest to criticality with Apros. This interface was found to be between the fifth and sixth axial fuel levels of the active core. This simulation case is denoted as case 5. In addition, the core was simulated with all but one control rod withdrawn from the core and the one rod inserted in the core, denoted as case 6, and with all but one control rod inserted in the core and the one rod withdrawn from the core, denoted as case 7. The resulting multiplication factors and assembly and axial power distribution differences are shown in Tab. 5.4. The assembly power distributions calculated with Serpent and their differences between Apros and Serpent are presented in Figs. 5.9, 5.10 and 5.11 for the cases

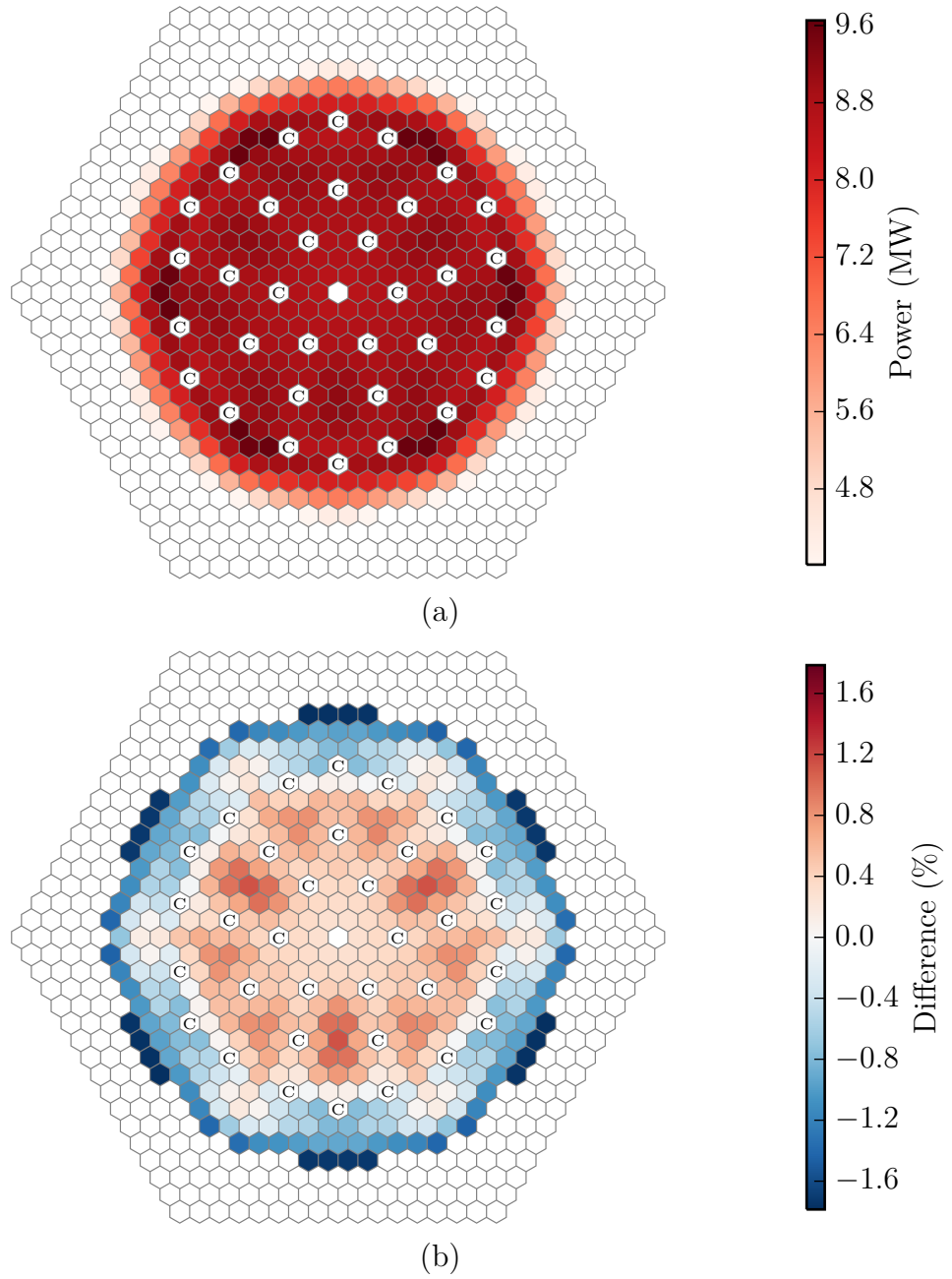


Figure 5.6: (a) Assembly powers calculated with Serpent and (b) assembly power differences between Apros and Serpent of the large oxide core with all control rods withdrawn from the core (case 2). The maximum positive and negative differences are 1.14 % and -1.79 %, respectively.

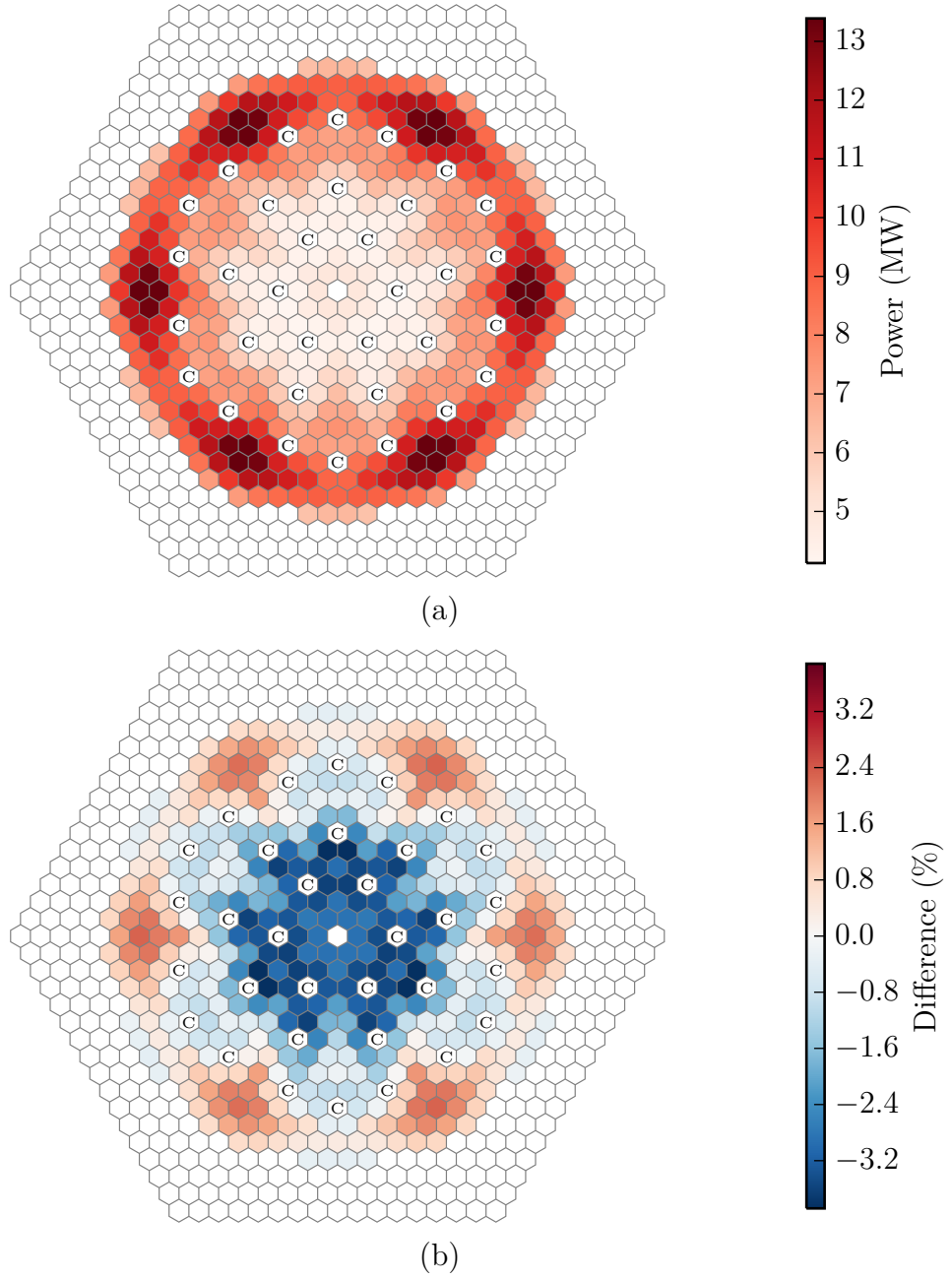
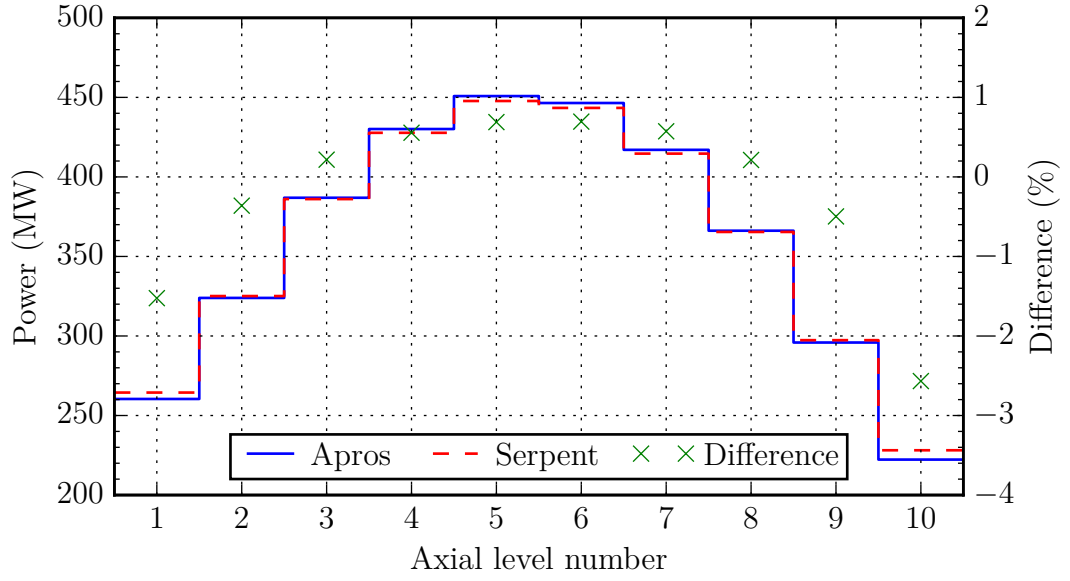
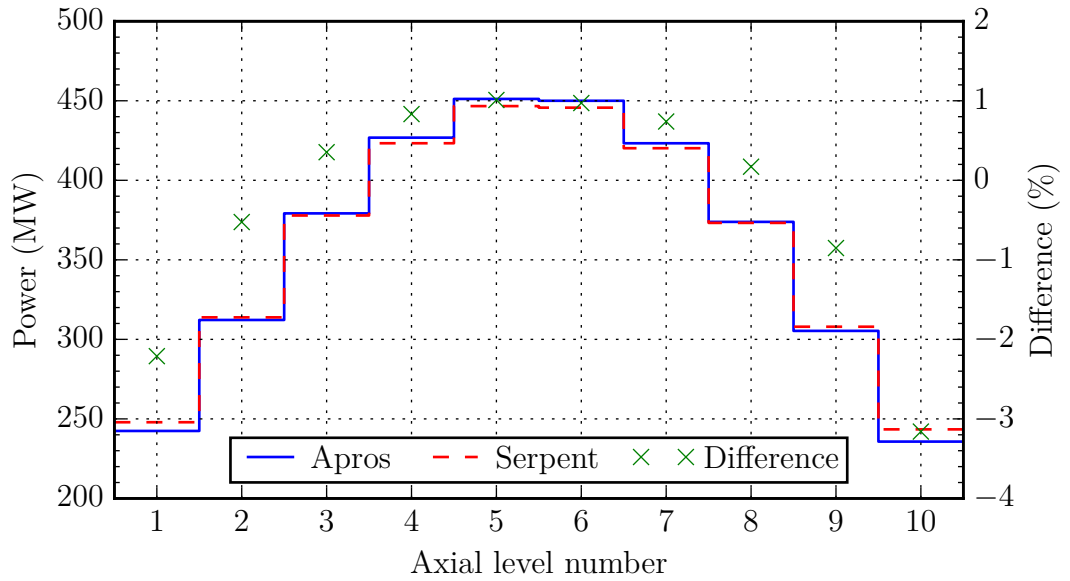


Figure 5.7: (a) Assembly powers calculated with Serpent and (b) assembly power differences between Apros and Serpent of the large oxide core with all control rods inserted in the core (case 4). The maximum positive and negative differences are 2.29 % and -3.88 %, respectively.



(a)



(b)

Figure 5.8: Axial powers of the large oxide core calculated with Apros and Serpent with all control rods (a) withdrawn from the core (case 2) and (b) inserted in the core (case 4).

5.2. STEADY STATE MODEL ACCURACY

Table 5.4: (a) Effective multiplication factors and (a) assembly and axial power differences of the large oxide core calculated with Apros and Serpent. The control rods are either at the interface of the fifth and sixth axial fuel levels of the active core (case 5), all but one control rod withdrawn from the core and the one rod inserted in the core (case 6) or all but one control rod inserted in the core and the one rod withdrawn from the core (case 7).

(a)

Case	Apros	Serpent	Uncertainty (10^{-4} %)	Difference (%)
5	0.99993	1.00508	4.4	-0.51
6	1.02865	1.03276	4.2	-0.40
7	0.97619	0.98229	4.6	-0.62

(b)

Case	Assembly power difference			Axial power difference		
	Max (%)	Min (%)	Mean (%)	Max (%)	Min (%)	Mean (%)
5	1.08	-1.23	0.45	0.96	-4.34	1.11
6	1.30	-2.29	0.63	0.70	-2.59	0.80
7	2.60	-3.86	1.16	1.00	-3.13	1.07

5, 6 and 7, respectively. The axial power distributions calculated with Apros and Serpent and their differences are presented in Fig. 5.12 for the case 5.

Compared with the differences of the multiplication factors, the Serpent uncertainties are insignificant. All the multiplication factors are underestimated. The difference of the case 5 is in the middle of the differences of the cases 2 and 4, and the differences of the cases 6 and 7 are the same as those of the cases 2 and 4.

All the Serpent relative statistical errors of the node- and assemblywise powers are smaller than 6.2×10^{-2} % and 2.8×10^{-2} %, respectively, in the case 5, 6.9×10^{-2} % and 3×10^{-2} %, respectively, in the case 6, and 6.1×10^{-2} % and 3×10^{-2} %, respectively, in the case 7. Therefore, they are insignificant compared with the differences between the codes.

In the case 5, the highest assembly power differences are at the outermost fuel assemblies which have the lowest powers. The assemblies with the highest powers have also relatively high differences. The highest assembly powers are overestimated. Compared with the cases 2 and 4, the assembly power differences are smaller.

In the case 6, the assembly power differences are otherwise similar to those of the case 2, but the magnitude of the maximum differences are higher and the inserted control rod makes the differences on its side of the inner core negative instead of

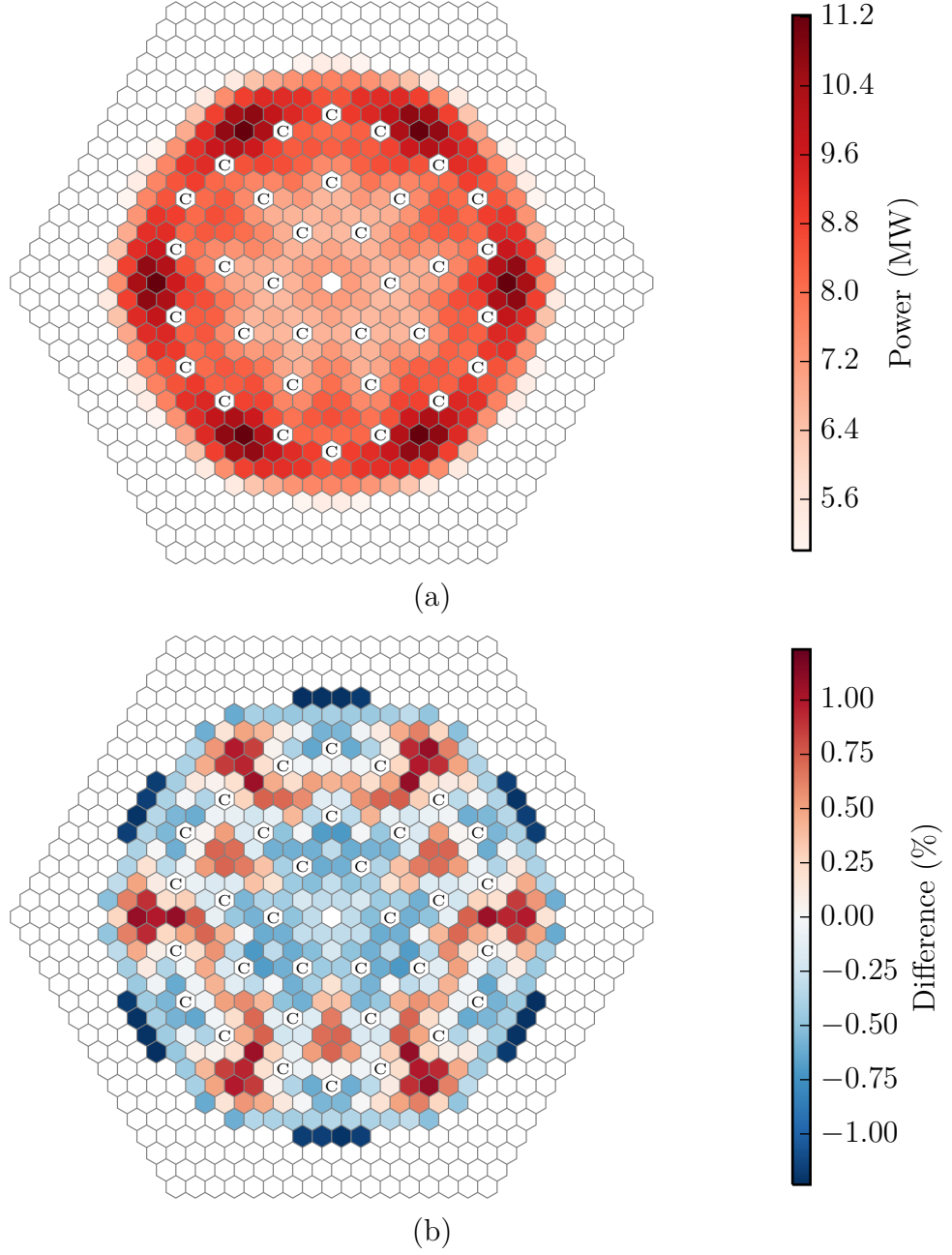


Figure 5.9: (a) Assembly powers calculated with Serpent and (b) assembly power differences between Apros and Serpent of the large oxide core with all control rods at the interface of the fifth and sixth axial fuel levels of the active core (case 5). The maximum positive and negative differences are 1.08 % and -1.23 %, respectively.

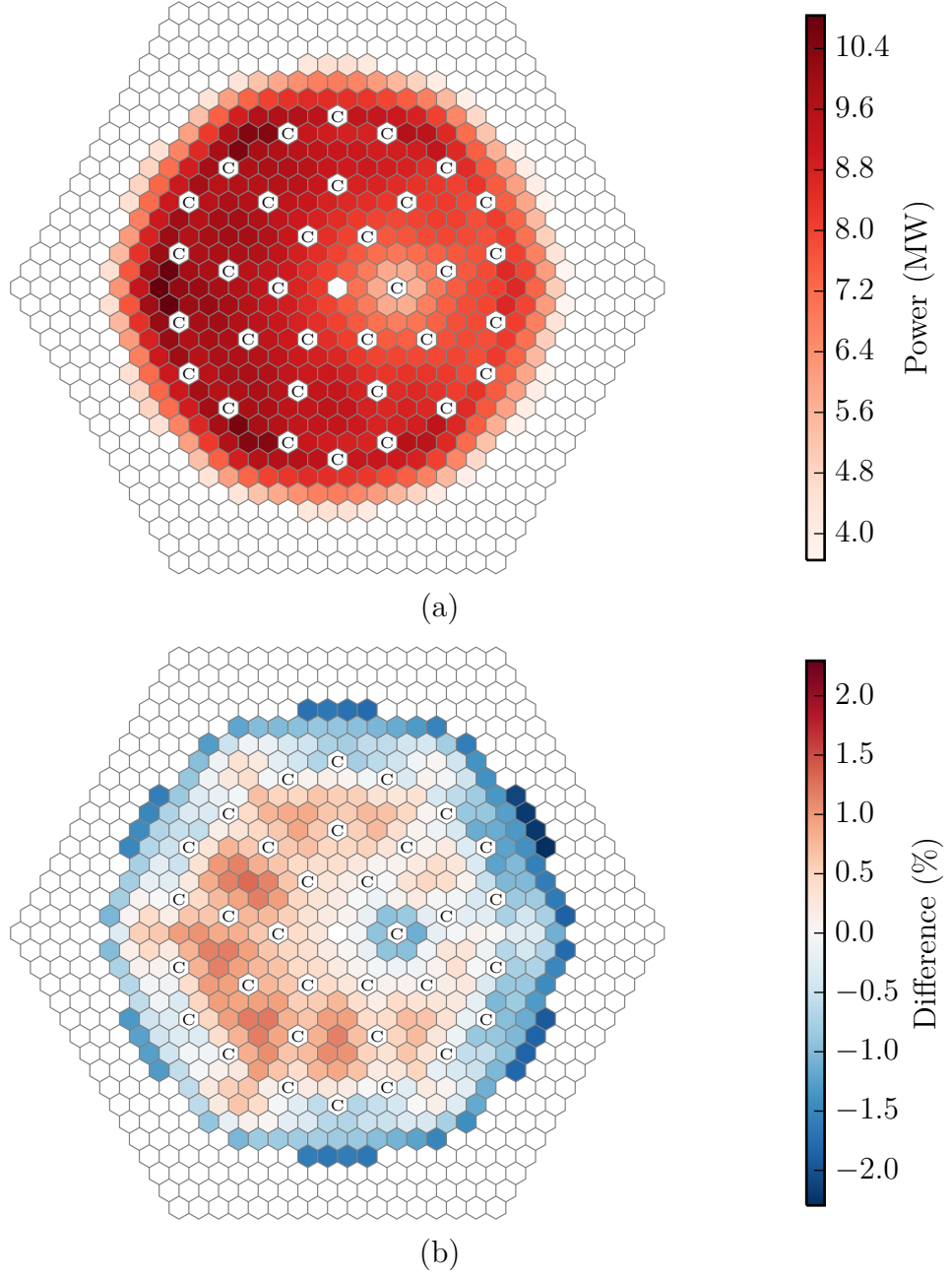


Figure 5.10: (a) Assembly powers calculated with Serpent and (b) assembly power differences between Apros and Serpent of the large oxide core with all but one control rod withdrawn from the core and the one rod inserted in the core (case 6). The maximum positive and negative differences are 1.30 % and -2.29 %, respectively.

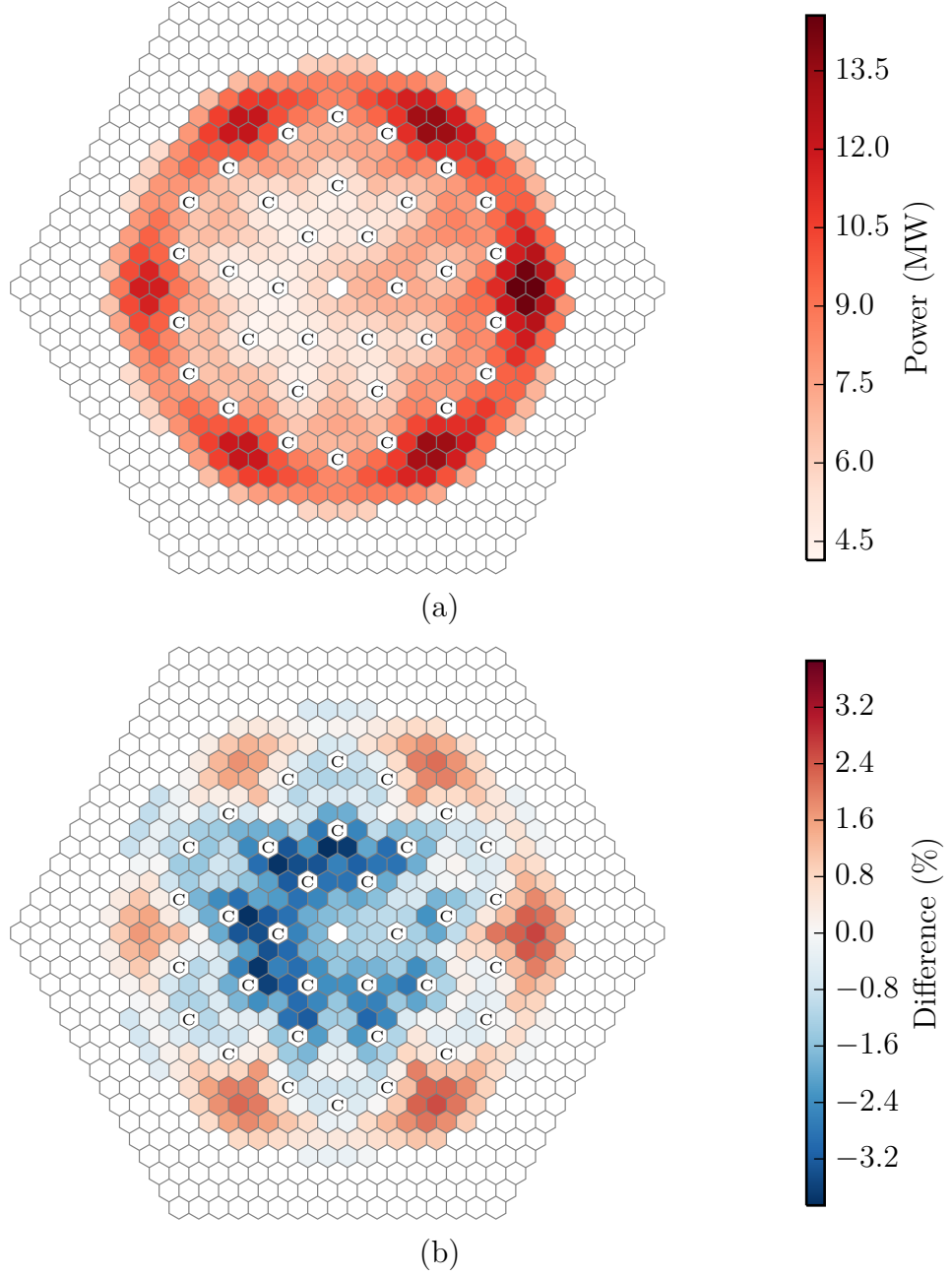


Figure 5.11: (a) Assembly powers calculated with Serpent and (b) assembly power differences between Apros and Serpent of the large oxide core with all but one control rod inserted in the core and the one rod withdrawn from the core (case 7). The maximum positive and negative differences are 2.60 % and -3.86 %, respectively.

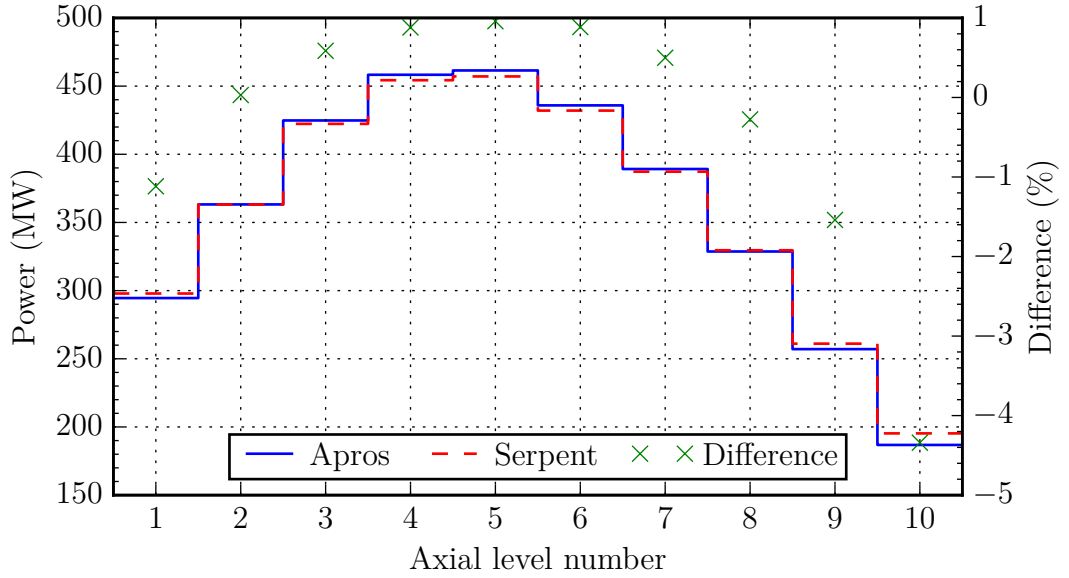


Figure 5.12: Axial powers of the large oxide core calculated with Apros and Serpent with all control rods at the interface of the fifth and sixth axial fuel levels of the active core (case 5).

positive. The powers of the outermost fuel assemblies are also underestimated more on the side of the core where the control rod is inserted. In addition, the assemblies next to the inserted control rod have somewhat high differences. These assemblies have relatively low powers.

In the case 7, the withdrawn control rod slightly decreases maximum and mean assembly power differences when compared with the case 4. The assembly powers on the side of the core with the withdrawn control rod are increased and the assembly power differences next to the control rod are decreased. However, the differences are increased at the outer boundary of its side of the core.

In the case 5, the axial powers are underestimated in the bottom and top of the core and overestimated in the middle of the core. The highest axial power is overestimated. The axial power differences are much higher in the top of the core where the axial powers are the lowest than in the bottom of the core. Compared with the cases 2 and 4, the maximum difference is higher. The mean difference is almost equal to that of the case 4. The axial power distributions of the cases 6 and 7 are almost identical to those of the cases 2 and 4 and are therefore not presented here.

In conclusion, the assembly power differences were smaller with the control rods near critical positions than with the control rods fully withdrawn or inserted. In contrast, the axial power differences were higher. The multiplication factor

difference was between those of the cases 2 and 4. A single control rod in a different position than the other control rods does not really alter the maximum and mean assembly power differences, the axial power differences or the multiplication factor differences. However, the assembly power difference distribution changes locally near the control rod and is also altered on the full core level.

5.2.2.3 Different number of energy groups

Finally the cases 2, 4 and 5 were simulated with Apros also using 12 and 6 energy groups. The resulting multiplication factors and assembly and axial power distribution differences are shown in Tab. 5.5. The results of the previous simulations using 24 energy groups are also included in the table for comparison.

Compared with the differences of the multiplication factors, the Serpent uncertainties are insignificant. With 12 groups, the multiplication factors are slightly better than with 24 groups in the cases 2 and 5 and slightly worse in the case 4. With 6 groups, the multiplication factor differences are almost twice as high as those of with 24 groups.

All the Serpent relative statistical errors of the node- and assemblywise powers, presented earlier, are insignificant compared with the differences between the codes. When comparing the assembly power differences with 12 groups with those of with 24 groups, the maximum difference is smaller and the mean difference is slightly higher in the case 2. In the case 4 and 5, the maximum and mean differences are smaller. When comparing the differences with 6 groups with those of with 24 and 12 groups, the maximum and mean differences are the highest in the cases 2 and 5, whereas the differences are the smallest in the case 4.

When comparing the axial power differences with 12 groups to those of with 24 groups, the maximum and mean differences are lower in all the cases. When comparing the differences with 6 groups to those of with 24 and 12 groups, in the cases 2 and 4 the maximum and mean differences are smaller than with 24 groups but higher than with 12 groups. In the case 5, the maximum and mean differences are smaller than with 24 groups and the mean difference is smaller but the maximum difference is higher than with 12 groups.

To conclude, the multiplication factors were enhanced or very slightly worsened when the number of energy groups were decreased from 24 to 12, and they were the worst with 6 energy groups. The power distribution differences were smaller with 12 groups than with 24 groups in almost all the cases. With 6 groups, the assembly power differences were the highest in two cases and smallest in one case, and the axial power distributions were better than with 24 groups in all the cases and almost always worse than with 12 groups.

A more expected result would have been the worsening of the results when the number of energy groups is decreased. Therefore, these results might be caused

5.2. STEADY STATE MODEL ACCURACY

Table 5.5: (a) Effective multiplication factors and (b) assembly and axial power differences of the large oxide core calculated with Apros and Serpent. The Apros simulations were performed using different numbers of energy groups. The control rods are either withdrawn from the core (case 2), inserted in the core (case 4) or at the interface of the fifth and sixth axial fuel levels of the active core (case 5).

(a)

Case	G	Apros	Serpent	Uncertainty (10^{-4} %)	Difference (%)
2	24	1.02994	1.03404	4.2	-0.40
2	12	1.03036	1.03404	4.2	-0.36
2	6	1.02648	1.03404	4.2	-0.73
4	24	0.97551	0.98161	4.6	-0.62
4	12	0.97547	0.98161	4.6	-0.63
4	6	0.97046	0.98161	4.6	-1.14
5	24	0.99993	1.00508	4.4	-0.51
5	12	1.00018	1.00508	4.4	-0.49
5	6	0.99567	1.00508	4.4	-0.94

(b)

Case	G	Assembly power difference			Axial power difference		
		Max (%)	Min (%)	Mean (%)	Max (%)	Min (%)	Mean (%)
2	24	1.14	-1.79	0.63	0.70	-2.57	0.79
2	12	1.25	-1.28	0.68	0.38	-2.11	0.45
2	6	2.14	-1.51	0.87	0.40	-2.41	0.54
4	24	2.29	-3.88	1.31	1.01	-3.16	1.08
4	12	1.81	-3.27	1.10	0.63	-2.04	0.67
4	6	1.73	-2.69	1.00	0.61	-2.43	0.71
5	24	1.08	-1.23	0.45	0.96	-4.34	1.11
5	12	1.07	-0.88	0.39	0.71	-3.92	0.96
5	6	2.25	-1.32	0.58	0.70	-4.09	0.95

by suitable error canceling, especially with 12 energy groups. In addition, the group structures with less energy groups were formed by merging adjacent groups. A better way to form the energy group division would be based on the importance of the different energy intervals in the analysis of sodium fast reactors. All in all, a more thorough study of the energy group structures would be needed to provide recommendations for the number of energy groups to be used to achieve suitable accuracy whereas also improving the simulation speed, even though the results of the studied cases were the most accurate with 12 energy groups.

5.3 Transient model

The large oxide core was simulated with the three different control rod movement patterns presented in Sec. 4.5 with both the Apros transient model and DYN3D. The simulations were performed using a fixed time step length of 0.05s. The group constants taking into account the neutron energy spectrum softening in the radial reflector were used. Stricter convergence criteria for the multiplication factor $\varepsilon_1 = 10^{-9}$ and pointwise fission source $\varepsilon_2 = 10^{-6}$ and the default criterion for the average fission source $\varepsilon_3 = 10^{-5}$ were used. The first criterion was used to initialize the transient calculation by a well-converged steady state calculation and the second criterion was used to reduce the power fluctuations during the first time steps of the transient simulations.

The effective multiplication factor of the initial state was 0.999919 with Apros and 0.999149 with DYN3D, resulting in a difference of 0.08 %. The assembly powers at the initial state calculated with Apros and their differences between Apros and DYN3D are presented in Fig. 5.13. Compared with DYN3D, Apros underestimates the assembly powers in the middle of the core and overestimates them at the outer parts of the core.

The reactor total power during the slow insertion of all control rods is shown in Fig. 5.14 and the relative axial powers at the beginning and end of the simulation are shown in Fig. 5.15. The total power as a function of time is approximately the same for both codes although the Apros power decreases slightly faster. However, the growth rate of the difference decreases towards the end of the simulation. The relative axial powers are very close to each other at the beginning and end of the simulation. In addition, the differences are mainly very slightly smaller at the end of the simulation. The differences of the actual absolute axial powers are however larger, as the reactor total power is higher with DYN3D.

The reactor total power during the fast insertion of all control rods is shown in Fig. 5.16 and the relative axial powers at the beginning and end of the simulation are shown in Fig. 5.17. The total power as a function of time is approximately the same for both codes. However, different from the case of the slow insertion

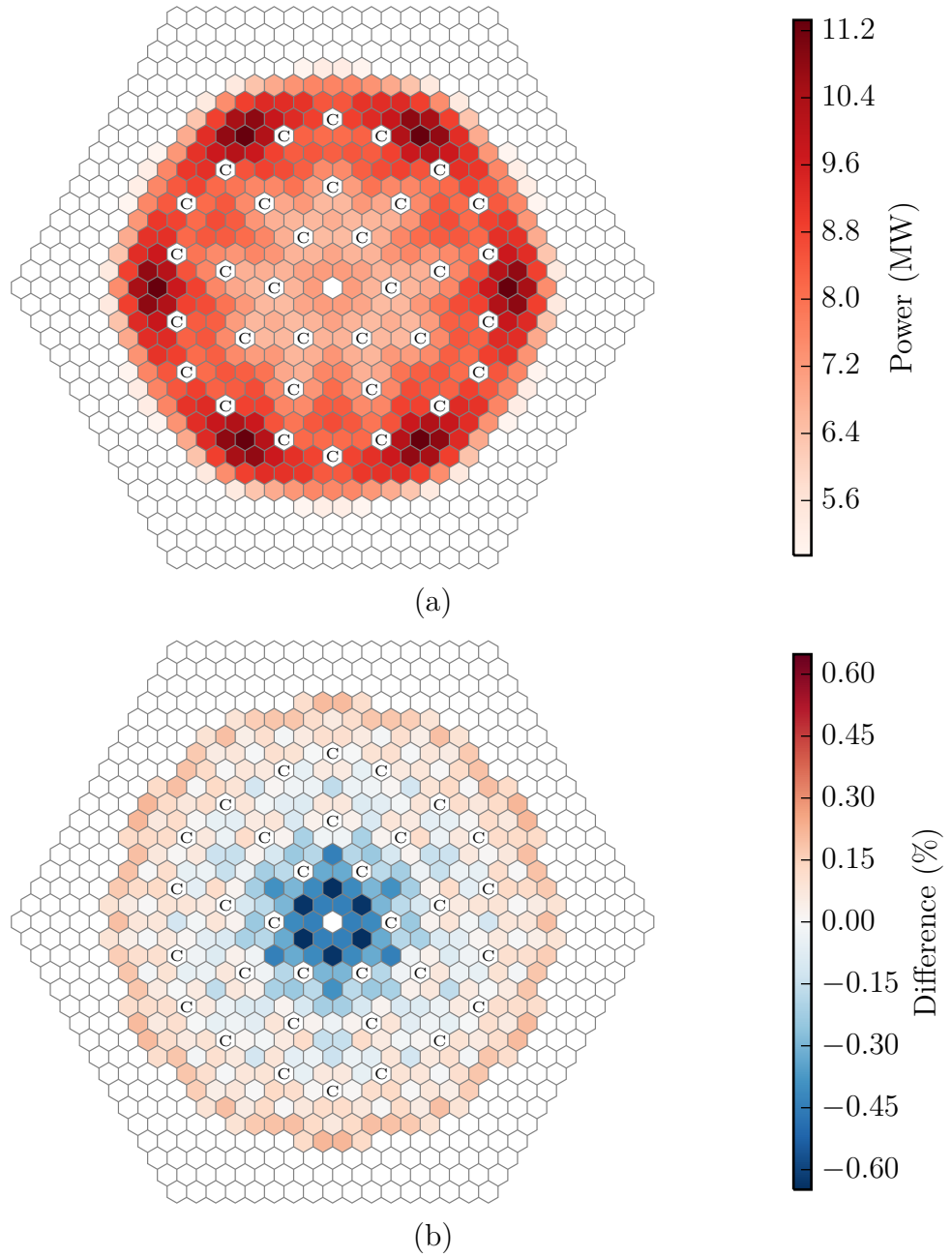


Figure 5.13: (a) Assembly powers calculated with Apros and (b) assembly power differences between Apros and DYN3D of the large oxide core at the initial state of the transient simulations. The maximum positive and negative differences are 0.22 % and -0.65 %, respectively.

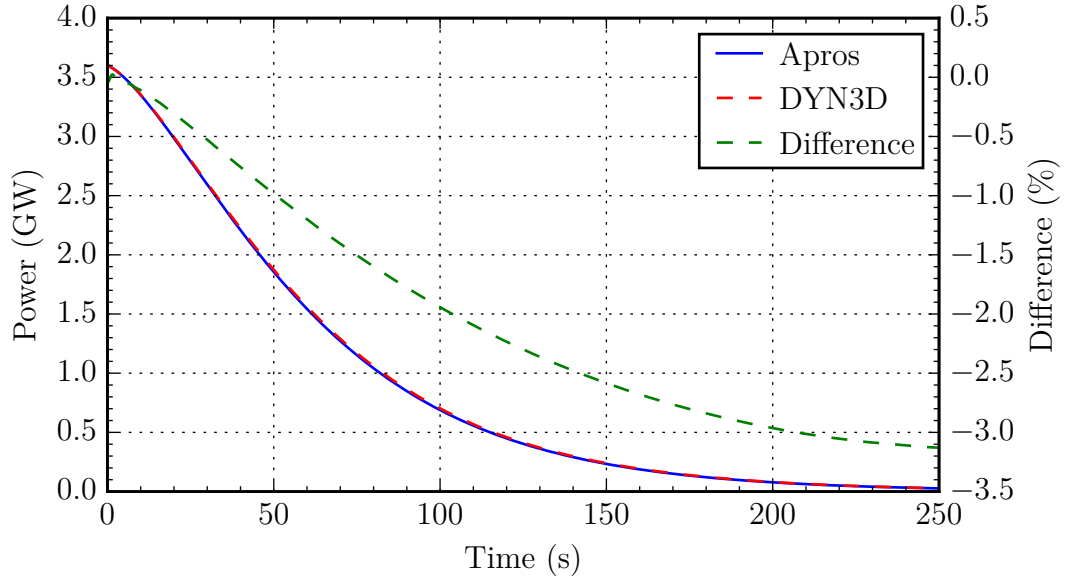


Figure 5.14: Total power of the large oxide core in the case of slow insertion of control rods calculated with Apros and DYN3D, and the difference between the codes.

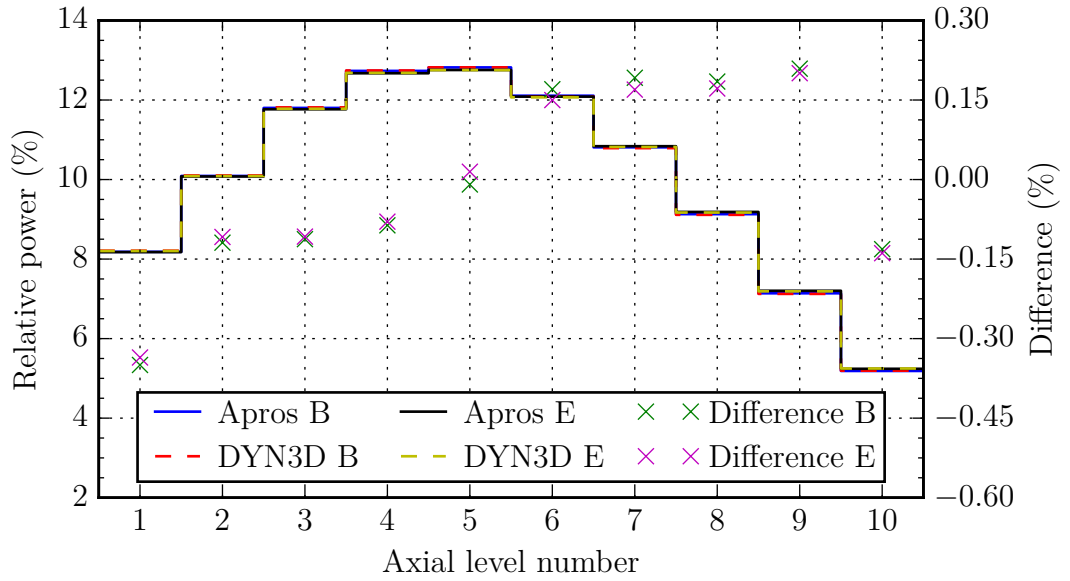


Figure 5.15: Relative axial powers and the difference between the relative axial powers at the beginning (B) and end (E) of the transient simulation in the case of slow insertion of control rods.

of all control rods, the total power difference increases and decreases periodically with a cycle of about 100s. As the control rods move at the speed of 1 mm/s and as the axial fuel levels of the active core are 10.0566 cm long, the cycle period is caused by the control rod movement over one axial fuel level. Therefore, when the control rods are first inserted into an axial fuel level, Apros overestimates the rate on which the power decreases when compared with DYN3D, and after about one third or half of the axial level the difference starts to decrease. In addition, compared with the first case, the maximum relative difference is smaller and the difference is relatively close to zero at the end of the simulation. Also in this case the relative axial powers are very close to each other at the beginning and end of the simulation, even though the axial power shape changes much more. In addition, the differences are mainly smaller at the end of the simulation.

The reactor total power during the very fast step withdrawal of all control rods is shown in Fig. 5.18 and the relative axial powers at the beginning and end of the simulation are shown in Fig. 5.19. In this case, the total power difference has a jump at the beginning of the simulation and increases during the whole simulation. The absolute total power difference is also larger, as the reactor power is steadily increasing instead of decreasing. However, the total power time behaviors are still somewhat the same, and the relative difference is not higher than in the case of slow insertion of all control rods. The axial power shapes do not practically change during the simulation.

To summarize, with the two control rod insertion transient simulations, the reactor total power behaves almost similarly with both codes. The increasing and decreasing of the differences are related to the control rod positions in partially rodded nodes. With the step control rod withdrawal simulation, the reactor total power difference increases during the whole simulation almost linearly. However, the relative difference is not higher than in the case of the slow insertion of control rods. The relative axial powers are very close between the codes at the beginning and end of the simulations, even though the relative axial power shape visibly changes in the case of the fast insertion of control rods. Taking into account the magnitude of the total power and axial power differences between the codes, the Apros transient model can be concluded to be correctly implemented to solve time-dependent problems.

The reactor total power differences between the codes might be caused by the different initial power distributions. They are in turn caused by the different solution methods of the codes. For example, in the axial direction Apros uses a third degree polynomial for the one-dimensional partially integrated flux, whereas DYN3D uses a second degree polynomial and two exponential functions. When comparing the results, the differences between Apros and DYN3D can not be interpreted as faults in the simulation accuracy of Apros, as DYN3D is only another nodal diffusion code

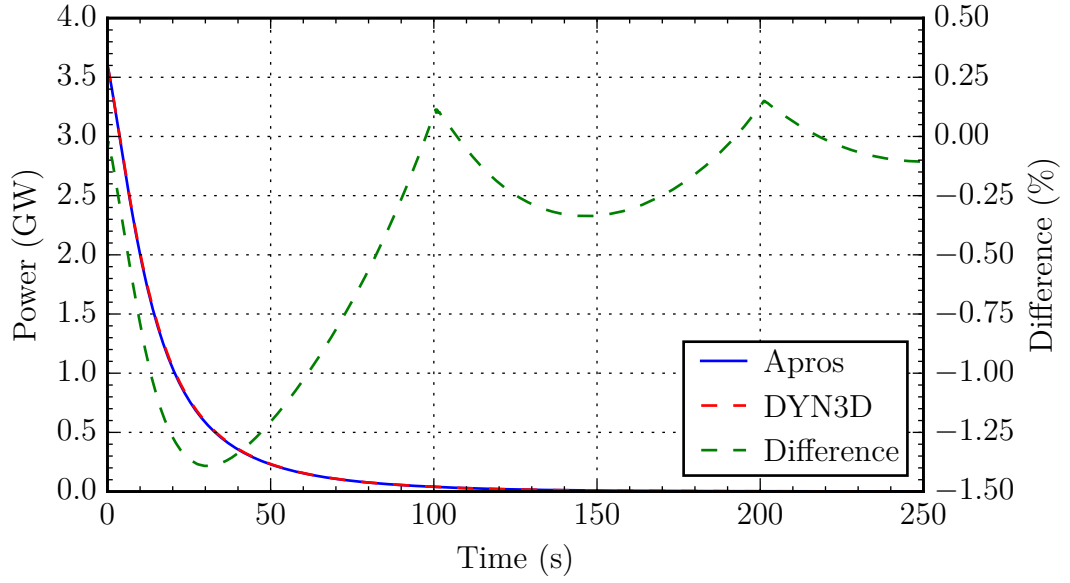


Figure 5.16: Total power of the large oxide core in the case of fast insertion of control rods calculated with Apros and DYN3D, and the difference between the codes.

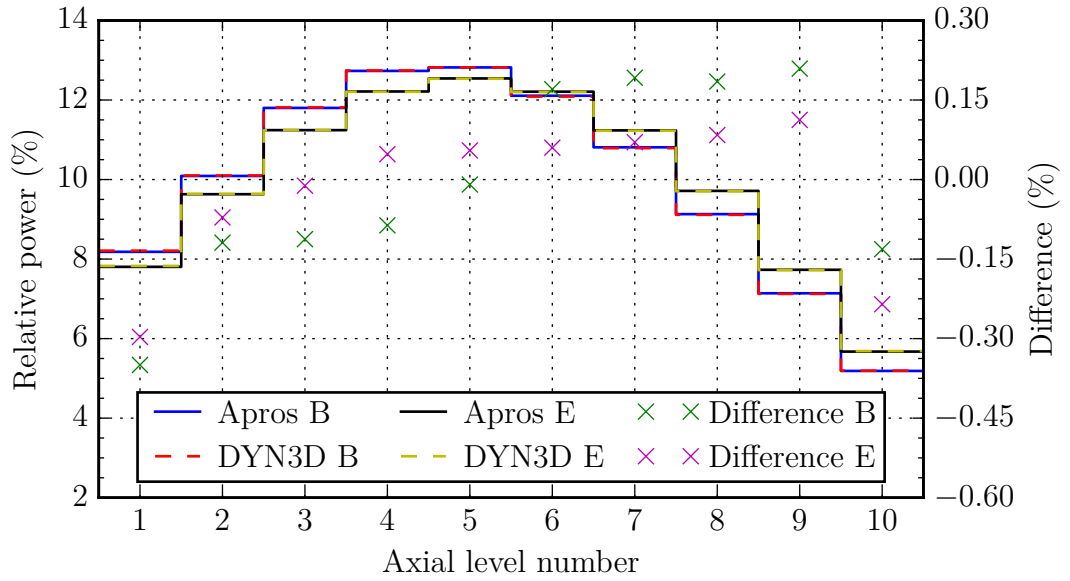


Figure 5.17: Relative axial powers and the difference between the relative axial powers at the beginning (B) and end (E) of the transient simulation in the case of fast insertion of control rods.

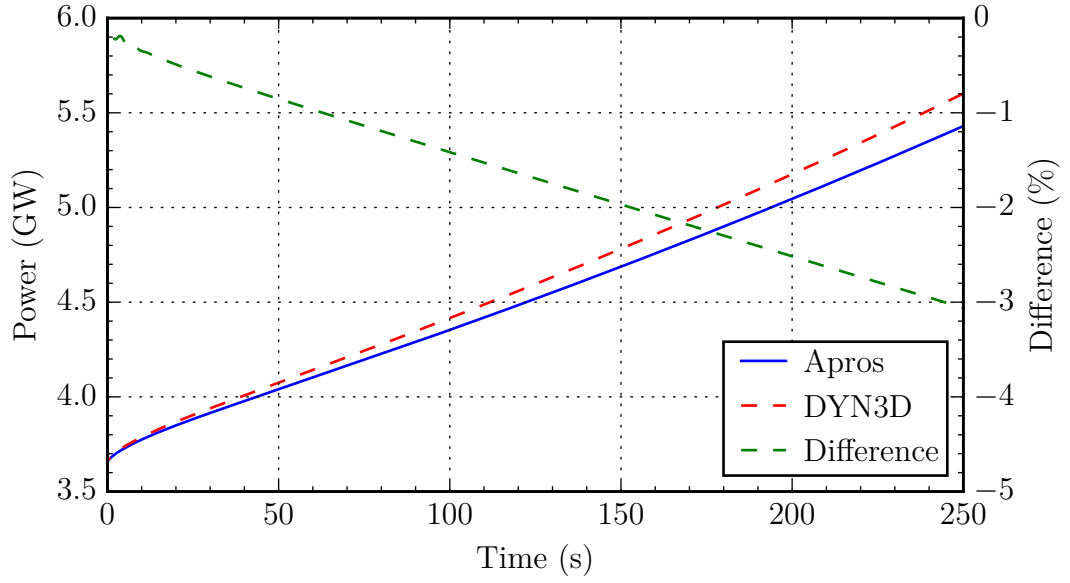


Figure 5.18: Total power of the large oxide core in the case of very fast step withdrawal of control rods calculated with Apros and DYN3D, and the difference between the codes.

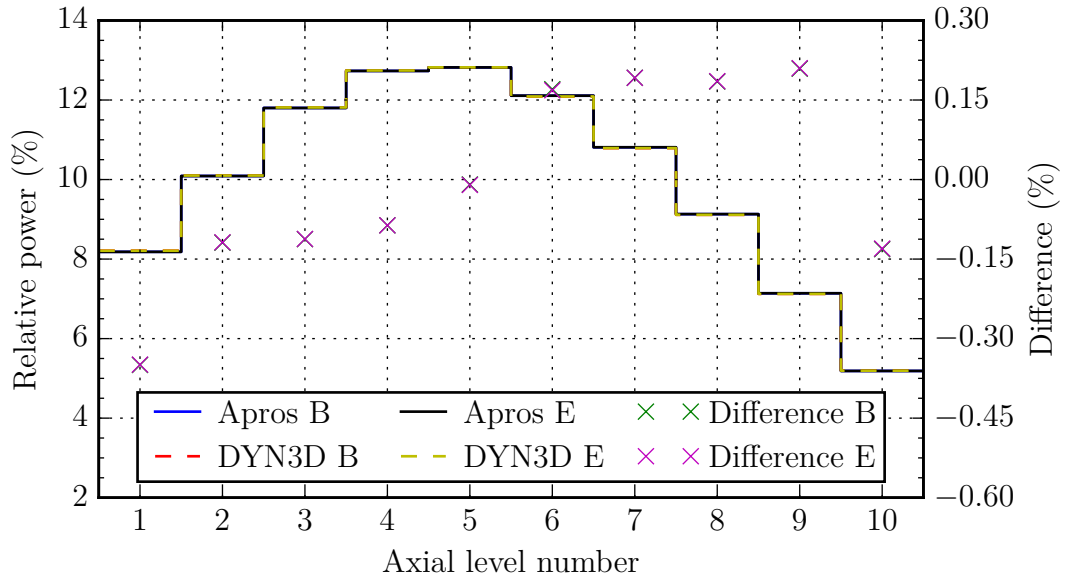


Figure 5.19: Relative axial powers and the difference between the relative axial powers at the beginning (B) and end (E) of the transient simulation in the case of very fast step withdrawal of control rods.

instead of a reference solution. To provide accurate estimations of Apros simulation accuracy, simulations with thermal hydraulic feedbacks and comparisons against measurements or other codes would be needed. For example in the case of the step withdrawal of control rods, as the DYN3D total power rises faster than the Apros power, the typically negative temperature feedback coefficients would slow down the power increase, whereas the slower Apros power rise would not be slowed down as much.

Chapter 6

Discussion

In this chapter, first the main validation results of the Apros multigroup nodal diffusion model are recapitulated. Second, possible research topics related to the model and development suggestions to improve the model are presented. Finally, the entire work is summarized.

6.1 Validation results

The validation of the Apros multigroup nodal diffusion model consisted of three parts. First, the implementation of the steady state model was validated against DIF3D on which the model is based. Second, the accuracy of the model was studied against Serpent. Finally, the validation of the implementation and similarity of the transient model results was performed against DYN3D. All the reactor models used in the validation were obtained from the OECD/NEA Sodium-cooled fast reactor (SFR) Benchmark Task Force benchmark definitions [20].

The validation of the implementation of the steady state model was performed with two different sized reactor cores with all control rods withdrawn. The obtained multiplication factor differences were extremely small between the codes, being under $4 \times 10^{-7} \%$ in both cases. The differences of node- and groupwise average fluxes were also very small. Almost all the differences were within 0.00125 % between the codes. The maximum differences were under 0.07 % in both cases. This can be considered an excellent agreement between the codes. Therefore, the implementation of the Apros steady state model can be stated to be correct.

The validation of the accuracy of the steady state model was performed also with the two different sized cores. The studied cases with the medium sized core were with all control rods withdrawn and inserted. The resulting multiplication factor differences were -1.03% and -2.65% , which can be considered relatively high. All the assembly power differences were under 2.40 % and 2.99 % with mean differences

of 1.60 % and 1.53 %. All the axial power differences were under 2.93 % and 2.02 % with the mean differences of 0.66 % in both cases. The assembly and axial power differences were on an acceptable level. The distributions of the assembly power differences suggest that the accuracy of the powers could be enhanced by utilizing radial discontinuity factors not yet implemented in the model. Additionally, the accuracy of the powers of the outermost fuel assemblies might be enhanced by taking into account the neutron energy spectrum softening in the radial reflector.

With the large sized core, multiple comparisons were performed. First, the effect of the neutron energy spectrum softening in the radial reflector was studied with all control rods withdrawn and inserted. With all control rods withdrawn, the multiplication factor difference was decreased from -0.41 % to -0.40 % when the spectrum softening was taken into account. With all control rods inserted, the difference decreased from -0.66 % to -0.62 %. The maximum assembly power differences were decreased from 3.21 % to 1.79 % and increased from 2.77 % to 3.88 %. The mean differences were decreased from 1.09 % to 0.63 % and increased from 0.98 % to 1.31 %. The maximum axial power differences were decreased from 2.64 % to 2.57 % and from 3.25 % to 3.16 % and the mean differences were decreased from 0.81 % to 0.79 % and from 1.11 % to 1.08 %.

Even though the assembly power differences increased when the control rods were inserted, the group constants taking into account the spectrum softening were utilized in the rest of the calculations as all the other studied results were improved. The observations about the radial discontinuity factors in the case of the medium sized core hold also in the case of the large sized core. In addition, the accuracy of the axial powers might be enhanced by improving the modeling of the control rods, such as by specifying different group constants for the control rods at different axial levels of the core. The axial modeling accuracy of both cores might also be improved by utilizing axial discontinuity factors. However, this subject has not been thoroughly studied.

Second, an additional case involving control rods at positions bringing the reactor nearest to criticality with Apros and two cases with asymmetrically positioned control rods were studied. In these two cases, all but one control rod were withdrawn and the one rod was inserted, and all but one control rod were inserted and the one rod was withdrawn. The obtained multiplication factor differences were -0.51 %, -0.40 % and -0.62 %, respectively. The obtained maximum assembly power differences were 1.23 %, 2.29 % and 3.86 % and the mean differences were 0.45 %, 0.63 % and 1.16 %. The maximum axial power differences were 4.34 %, 2.59 % and 3.13 % and the mean differences were 1.11 %, 0.80 % and 1.07 %. Therefore, the accuracy of the assembly powers was the best of all the studied cases with the control rods nearest to criticality. However, the maximum axial power difference was the highest in this case. A control rod at a different position did not really alter the multipli-

cation factor or the maximum and average power differences. Of course, the actual assembly power distributions were altered more.

To conclude the comparisons between Apros and Serpent using 24 energy groups, the following statements can be made. The multiplication factor differences are relatively high between the codes, especially in the case of the medium sized core. The differences are smaller with the large sized core. With both cores, the insertion of the control rods increased the multiplication factor differences. Studied in the case of the large oxide core, the generation of group constants for the assemblies adjacent to the radial reflector taking into account the neutron energy spectrum softening in the radial reflector improved the multiplication factors slightly and improved the assembly powers when the control rods were withdrawn. However, the powers were worsened when the control rods were inserted. All in all, the assembly power differences with both cores were on an acceptable level.

Finally, the effect of the number of energy groups on the simulation results was studied by merging adjacent energy groups to form 12- and 6-group structures. In short, in almost all of the studied cases the multiplication differences were slightly smaller with 12 groups than with 24 groups, and almost twice as high as with 6 groups compared with the results of with 24 groups. The maximum and mean assembly and axial power differences were almost always smaller with 12 groups than with 24 groups. With 6 groups, the maximum and mean assembly power differences were the smallest in one studied case and the highest in two studied cases. The maximum and mean axial power differences were between those of with 24 and 12 groups in all cases, except for the mean difference in one case, which was slightly the smallest. The improvement of the results when decreasing the number of energy groups from 24 to 12 was an unexpected result, and might be caused by suitable error canceling. Therefore, a more thorough study on the suitable number of energy groups is needed. In addition, the smaller group structures were not constructed based on the importance of the different energy intervals in the simulations of sodium fast reactors. The importance should also be studied to provide recommendations for the required number of energy groups for the simulations.

The validation of the implementation of the Apros transient model and its accuracy against another nodal diffusion code was performed with the large sized core. As DYN3D is only another nodal diffusion code, its results can not be treated as reference results. However, the results can be compared to study whether they are alike. Three simulations with different control rod movement patterns were performed. The simulations were performed without simulating the thermal hydraulics and with a fixed time step length. In the initial state, the multiplication factor difference between the codes was -0.08% , the maximum assembly power difference was 0.65% and the maximum axial power difference was 0.35% .

The first case was a slow insertion of all control rods. The reactor total power time evolution was approximately the same for both codes. However, the power decreased faster with Apros, resulting in a maximum difference of -3.13% during the simulation. The differences in the relative axial powers between the codes mainly slightly decreased during the simulation.

The second case was a fast insertion of all control rods. The reactor total power time evolution was somewhat closer between the codes than in the previous case. Also in this case, the power decreased faster with Apros, resulting in a maximum difference of -1.39% during the simulation. In this case, the power difference did not increase during the whole simulation. When the control rods began descending into an axial level, the difference first increased. After about one third or one half of an axial level, the difference began to decrease. This pattern was repeated in all of the axial levels the control rods moved in during the simulation. At the end of the simulation, the power difference was small. Even though the shape of the axial power changed significantly during the simulation, the differences between the relative axial powers were still very small between the codes at the end of the simulation. The differences decreased in almost all of the axial levels.

The final case was a very fast step withdrawal of all control rods. In this case, the reactor total power time evolution was more different between the codes, as now the power was increasing instead of decreasing. However, the maximum relative difference -3.06% between the codes was in the same order as in the first case. The relative axial powers did not visibly change during the simulation.

Taking into account the magnitude of the total power and axial power differences between the codes, the Apros transient model can be concluded to be correctly implemented to solve time-dependent problems. The differences might be caused by the different solution methods of the codes, which result in different power distributions. As the simulation results of DYN3D are not reference results, the true accuracy of the Apros model should be validated against measurements or against multiple codes including the simulation of thermal hydraulics. The thermal hydraulics brings its own effect to the comparisons, especially due to different feedback effects, but also additional uncertainty due to the different models utilized in the codes.

6.2 Further research and model development

The basic functionality of the model has now been implemented and proven to work. However, there are still multiple subjects on which the model could be studied and improved. These subjects are divided here in four main topics. The first is the inclusion of the model in the release version of Apros, the second is the improvement

of the model usability, the third is the validation of the model and the final is the modeling improvements of the model.

First, the model has not been a part of the development version of Apros, and the development of the model had been halted for some time before this work. Therefore, it is partly incompatible with the newest development version. In order to enable the inclusion of the model in the release version of Apros when desired, these incompatibilities should be resolved and the model should be included in the development version of Apros to prevent the development of new incompatibilities.

Second, as the model has now only been in development use, it is not yet ready for production use. In order to improve the usability of the model, several improvements have to be made. First, error checking for user input has to be implemented. Where possible, the model should be made error tolerant. At the least, the model should inform the user about what has gone wrong. Second, the model should inform the user about its state when needed. One such case could be a long lasting calculation of a time step. Finally, as a full core model consists of hundreds of assemblies with multiple axial levels with different materials and burnups, the actual Apros models can not in reality be constructed manually. Therefore, some processing scripts to automate the model construction are required. Such scripts exist for the group constant generation and the Apros model creation. Their versatility, usability and documentation should however be improved before taking them into production use.

Third, all the performed validation of the model is described in this work. As stated earlier, both the steady state and the transient model should be validated together with thermal hydraulics to determine the true usability and accuracy of the model for fast reactor simulations. These studies should also be conducted for other types of liquid metal cooled reactors than sodium fast reactors, such as for lead-cooled fast reactors. In addition, as the effect of reducing the number of energy groups was unexpected, it should be further studied. Similar sensitivity studies should be conducted for the axial node division and time step lengths. Also the group constant generation methods should be further studied. These studies could include the effect of B_1 leakage corrected group constants produced by Serpent and the effect of generating the group constants with a three-dimensional single assembly model, as has been done for example in Ref [39].

Finally, the model has various existing features that could be improved and new features could be implemented to increase the accuracy and versatility of the model. The improvements include the following:

- The group constant model consists now only of two independent quadratic polynomials for coolant and fuel temperatures. This model could be improved by taking into account more feedback parameters, such as the coolant density, and to be modified to cover a wider range of parameter variations. The model

should also include cross terms instead of only having independent fits for the parameters.

- The control rods are now modeled as consisting of only one material with a single set of group constants. This model could be improved by allowing the control rods to have different group constants depending on the axial level.
- The control rods should also be able to consist of two or more material types, if they have for example tips consisting of different materials.
- Another important improvement related to control rods is the reduction of the control rod cusping effect by modeling the fluxes inside the rodded and unrodded parts of a partially rodded node. Using the fluxes, the group constants of the partially rodded node could be calculated with proper flux weighting instead of the volume weighting.
- The axial flux description of the model is now performed with a third-degree polynomial. In Ref. [42], the degree of the polynomial was increased to four, which improved the calculation results. The increase might be more important with thermal reactors, but the effect should be studied. The additional degree of the polynomial is derived for example in Ref. [43].

The new features include the following:

- Based on the validation results of this work, the utilization of discontinuity factors should be added to the model to reduce the model errors.
- If a strict convergence criterion is used for the fission source, the simulation of a time step might last long with the transient model. Therefore, an acceleration method such as the coarse mesh rebalancing used in the steady state model should be implemented into the transient model to speed up the convergence.
- All of the power is now released in fission reactions and the decay heat is therefore not taken into account. Simulation of the decay heat would be an important addition to the model, if a transient ending to a shut down or low power state of the reactor is desired to be simulated.
- All of the power is now released directly in the fuel. As a part of the fission power is released as γ -radiation, a part of the power is in reality released directly into the coolant, which should also be modeled.
- The reactor boundaries are now described only with single boundary coefficients in the axial and radial directions. If modeling without explicit radial and axial reflectors is desired, the user should be able to use material- and groupwise albedos to describe the boundaries.
- An important aspect related to the simulation speed is the time step length. Apropos has an automatic time step control for the thermal hydraulic system and the other neutronics models. This should also be implemented into the model. The time step should be decreased when the core neutronics state undergoes too rapid changes to prevent errors resulting from too long time

steps and increased when the core neutronics state is only slowly changing to increase the simulation speed.

- Burnup calculation capability could be added to the model. This feature is somewhat controversial, as transient calculations are the main applications of Apros. It would however enable the user to begin the simulations in different parts of the operation cycles, if the burnup distribution of the fuel is not available from elsewhere.
- The model is now only capable of simulating hexagonal assemblies. The addition of quadratic geometry would enable the simulation of more reactor types. However, most of the fast reactor concepts use hexagonal geometry, and therefore the quadratic geometry is not such an important feature. It is more related to the possibility of simulating thermal reactors with the model using more than the two energy groups used in the other Apros neutronics models. This would also require the addition of the reactor poisoning simulation to the model. It should be noted that some assumptions while deriving the solution methodology were based on the conditions present in fast reactors. DIF3D-K has, however, been also used for coupled simulations of thermal reactors [32, 44].
- The final proposed improvements of the model are related to its computational speed. The code should be profiled in order to find the parts where optimizations would produce the greatest improvements. In addition, all possible locations for the parallelization of the model should be identified and the effect of the parallelization be studied. Other parts of Apros, such as the six-equation model, have already been parallelized.

6.3 Summary

This thesis continued the development of fast reactor simulation capability of the process simulation software Apros. During this work, the existing multigroup nodal diffusion model program code was reviewed. Some errors in the steady state model and in the parts common to the steady state and transient models were corrected. In addition, some features, such as better utilization of Serpent group constants were added to the steady state model. Many parts of the transient model not common with the steady state model were found to be flawed, and the transient algorithm was derived and almost completely rewritten during this work. In addition, a flux extrapolation transient acceleration method was implemented into the model.

The development of the steady state model is described in Ref. [5]. The temporal solution method of the transient model is based on implicit discretization of the time derivative of the neutron flux. This method is similar to the implicit theta method of the neutron kinetics code DIF3D-K [29]. The discretization enables

the problem to be described as a fixed source problem for each time step. The spatial solution method is similar to that of the steady state model, but with small modifications. The solution algorithm for one time step is based on a fission source iteration process. In addition to the boundary conditions of the neutron flux, average neutron fluxes and precursor concentrations at the beginning of a time step are required to simulate a time step.

The validation of the steady state and transient models was performed in three parts. First, the steady state model was compared against the hexagonal geometry steady state diffusion equation solver of DIF3D, on which the steady state model is based. The differences of multiplication factors were extremely small, and the differences of the node- and groupwise average fluxes were found to be very small. Therefore, the steady state model can be concluded to be correctly implemented.

Second, the steady state model was compared against the continuous-energy Monte Carlo reactor physics code Serpent, which was also used to generate the group constants utilized in Apros. The multiplication factor differences were found to be relatively high, especially in the case of a medium sized core. The differences of assembly powers were found to be on an acceptable level. When studying the effect of the number of energy groups utilized in Apros, the results were unexpectedly improved when the number of energy groups was decreased from 24 to 12. Therefore, this effect requires a more thorough study.

Finally, the transient model was compared against the reactor dynamics code DYN3D. DYN3D is only another nodal diffusion code, and not a reference solution. Three different transients with control rod movements were studied without simulating the thermal hydraulics. The time evolutions of the reactor total powers were approximately the same between the codes for the two control rod insertion simulations. The total power time evolution of a very fast step withdrawal of all control rods was more different between the codes. However, the relative difference was not higher than in the case of a slow insertion of all control rods. The relative axial powers were very close between the codes both at the beginning and end of all the simulations. The differences can be concluded to be caused by the differences between the solution methods of the codes and by the differences in the power distributions caused by them. Therefore, the transient model is also correctly implemented and it produces results similar to DYN3D.

The basic features of a steady state and transient multigroup nodal diffusion model have now been implemented in Apros and verified to work. There are many subjects on which the model could be improved and studied. The most important of them is the bringing of the model to the development version of Apros and eventually to the release version. Other topics include performing diverse validation of the model especially together with thermal hydraulics, improving the model usability and improving the existing and adding new features to the model.

Bibliography

- [1] S. Schlömer et al. “Annex III: Technology-specific Cost and Performance Parameters”. In: *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2014.
- [2] *The Generation IV International Forum*. <https://www.gen-4.org/> (accessed July 8, 2015).
- [3] A. E. Waltar, D. R. Todd, and P. V. Tsvetkov. *Fast Spectrum Reactors*. Springer, 2012.
- [4] C. Fazio et al. *Handbook on Lead-Bismuth Eutectic Alloy and Lead Properties, Materials Compatibility, Thermal-hydraulics and Technologies*. OECD/NEA, 2007.
- [5] P. Inkinen. *Implementation of Nodal Multigroup Diffusion Model into APROS*. Master’s thesis. Helsinki University of Technology, 2009.
- [6] P. Inkinen. *State of development of APROS neutronics multigroup diffusion model*. VTT Technical Research Centre of Finland, 2013.
- [7] *Apros - Dynamic Process Simulation Software for Nuclear and Thermal Power Plant Applications*. <http://www.apros.fi/> (accessed June 31, 2015).
- [8] A. Santamarina et al. *The JEFF-3.1.1 Nuclear Data Library*. JEFF report 22. OECD/NEA, 2009.
- [9] A. K. Prinja and E. W. Larsen. “General principles of neutron transport”. In: *Handbook of Nuclear Engineering*. Springer, 2010, pp. 427–542.
- [10] A. Hébert. “Multigroup neutron transport and diffusion computations”. In: *Handbook of Nuclear Engineering*. Springer, 2010, pp. 751–911.
- [11] J. J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis*. John Wiley & Sons, 1976.
- [12] W. M. Stacey. *Nuclear Reactor Physics*. John Wiley & Sons, 2007.
- [13] J. Leppänen. *Development of a New Monte Carlo Reactor Physics Code*. PhD thesis. VTT Technical Research Centre of Finland, 2007.

- [14] E. K. Puska. *Apros three-dimensional nuclear reactor model user's guide*. 2014.
- [15] J. Ylijoki. Private communication. 2015.
- [16] E. K. Puska. *Apros point kinetics nuclear reactor model user's guide*. 2014.
- [17] E. K. Puska. *Apros nuclear reactor one-dimensional neutronics model user's guide*. 2014.
- [18] J. Rintala. *Implementation of a new neutronics model into APROS*. Master's thesis. Helsinki University of Technology, 2008.
- [19] E. Kujawski, L. J. W, and W. R. M. *The Large Core Code Evaluation Working Group benchmark analyses of a homogeneous fast reactor*. Technical Information Center, US Dept. of Energy, 1981.
- [20] D. Blanchet et al. *AEN-WPRS Sodium Fast Reactor Core Definitions*. Version 1.2. CEA and Argonne National Laboratory, 2011.
- [21] D. Tenchine et al. "International benchmark on the natural convection test in Phenix reactor". In: *Nuclear Engineering and Design* 258 (2013), pp. 189–198.
- [22] P. Juutilainen. Private communication. 2015.
- [23] P. Inkinen. Private communication. 2012.
- [24] E. Fridman, J. Leppänen, and C. Wemple. "Comparison of Serpent and HELIOS-2 as applied for the PWR few-group cross section generation". In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering*. 2013.
- [25] J. Leppänen. *Serpent – a Continuous-energy Monte Carlo Reactor Physics Burnup Calculation Code*. VTT Technical Research Centre of Finland, 2013.
- [26] E. Fridman and J. Leppänen. "Revised methods for few-group cross section generation in the Serpent Monte Carlo code". In: *PHYSOR 2012*. 2012.
- [27] J. Leppänen. *Methodology for spatial homogenization in Serpent 2*. Memo. May 26, 2014.
- [28] R. D. Lawrence. *DIF3D Nodal Neutronics Option for Two- and Three-dimensional Diffusion Theory Calculations in Hexagonal Geometry*. Argonne National Laboratory, 1983.
- [29] T. A. Taiwo and H. S. Khalil. "The DIF3D Nodal Kinetics Capability in Hexagonal-Z Geometry-Formulation and Preliminary Tests". In: *International Topical Meeting on Advances in Mathematics, Computations, and Reactor Physics*. 1991.

- [30] T. A. Taiwo and H. S. Khalil. “An Improved Quasistatic Option for the DIF3D Nodal Kinetics Code”. In: *The American Nuclear Society Meeting, Advances in Reactor Physics*. 1992.
- [31] T. A. Taiwo et al. *Time-Step Selection Considerations in the Analysis of Reactor Transients with DIF3D-K*. Argonne National Laboratory, 1993.
- [32] H. S. Khalil. “Coupled Reactor Physics and Thermal-Hydraulics Computations”. In: *Joint International Conference on Mathematical Methods and Supercomputing for Nuclear Applications*. 1997.
- [33] U. Grundmann, U. Rohde, and S. Mittag. “DYN3D – three-dimensional core model for steady-state and transient analysis of thermal reactors”. In: *PHYSOR 2000*. 2000.
- [34] U. Grundmann et al. *DYN3D Version 3.2. Code for Calculation of Transients in Light Water Reactors (LWR) with Hexagonal or Quadratic Fuel Elements, Description of Models and Methods*. Forschungszentrum Rossendorf, Institute of Safety Research, 2005.
- [35] A. Pautz and A. Birkhofer. “DORT-TD: A transient neutron transport code with fully implicit time integration”. In: *Nuclear science and engineering* 145.3 (2003), pp. 299–319.
- [36] E. Fridman and E. Shwageraus. “Modeling of SFR cores with Serpent–DYN3D codes sequence”. In: *Annals of Nuclear Energy* 53 (2013), pp. 354–363.
- [37] R. Rachamin, C. Wemple, and E. Fridman. “Neutronic analysis of SFR core with HELIOS-2, Serpent, and DYN3D codes”. In: *Annals of Nuclear Energy* 55 (2013), pp. 194–204.
- [38] E. Dorval. “Directional diffusion coefficients and leakage-corrected discontinuity factors: Implementation in Serpent and tests”. In: *Annals of Nuclear Energy* 87 (2016), pp. 101–112.
- [39] E. Nikitin, E. Fridman, and K. Mikityuk. “Solution of the OECD/NEA neutronic SFR benchmark with Serpent-DYN3D and Serpent-PARCS code systems”. In: *Annals of Nuclear Energy* 75 (2015), pp. 492–497.
- [40] K. S. Smith. “Assembly homogenization techniques for light water reactor analysis”. In: *Progress in Nuclear Energy* 17.3 (1986), pp. 303–335.
- [41] E. Fridman et al. “Axial discontinuity factors for the nodal diffusion analysis of high conversion BWR cores”. In: *Annals of Nuclear Energy* 62 (2013), pp. 129–136.

- [42] Y. Shimazu, T. Takeda, and W. F. G. van Rooijen. “Development of a three-dimensional kinetics code for commercial-scale FBR full core analysis”. In: *PHYSOR 2014*. 2014.
- [43] R. D. Lawrence. “Progress in nodal methods for the solution of the neutron diffusion and transport equations”. In: *Progress in Nuclear Energy* 17.3 (1986), pp. 271–301.
- [44] M. H. Kim, T. A. Taiwo, and H. S. Khalil. “Analysis of the NEACRP PWR rod ejection benchmark problems with DIF3D-K”. In: *American Nuclear Society Topical Meeting on Advances in Reactor Physics*. 1994.