POLCA-T Neutron Kinetics Model Benchmarking

Jurij Kotchoubey

Master Of Science Thesis
Stockholm 2015
Abstract

The demand for computational tools that are capable to reliably predict the behavior of a nuclear reactor core in a variety of static and dynamic conditions does inevitably require a proper qualification of these tools for the intended purposes. One of the qualification methods is the verification of the code in question. Hereby, the correct implementation of the applied model as well as its flawless implementation in the code are scrutinized. The present work concerns with benchmarking as a substantial part of the verification of the three-dimensional, multigroup neutron kinetics model employed in the transient code POLCA-T.

The benchmarking is done by solving some specified and widely used space-time kinetics benchmark problems and comparing the results to those of other, established and well-proven spatial kinetics codes. It is shown that the obtained results are accurate and consistent with corresponding solutions of other codes.

In addition, a sensitivity analysis is carried out with the objective to study the sensitivity of the POLCA-T neutronics to variations in different numerical options. It is demonstrated that the model is numerically stable and provide reproducible results for a wide range of various numerical settings. Thus, the model is shown to be rather insensitive to significant variations in input, for example. The other consequence of this analysis is that, depending on the treated transient, the computing costs can be reduced by, for instance, employing larger time-steps during the time-integration process or using a reduced number of iterations.

Based on the outcome of this study, one can finally conclude that the POLCA-T neutron kinetics is modeled and implemented correctly and thus, the model is fully capable to perform the assigned tasks.
Acknowledgments

First and foremost I would like to express my sincere gratitude to my supervisor at Westinghouse, Erwin Müller, for the introduction into the world of reactor physics methods, his numerous explanations, advice and anything else that made the collaboration really constructive and very pleasant during this project. I would also like to thank my supervisor at KTH Royal Institute of Technology Jan Dufek, my examiner Waclaw Gudowski, and the reviewer Sean Roshan Ghias for their valuable comments on this thesis.

All the support from my fellow students at KTH and my colleagues at Westinghouse that I got during my entire education at KTH is gratefully appreciated.

Finally, I would like to thank my lovely family for their understanding, support and, in particular, their patience during my studies.
Contents

1 Introduction ............................... 1
   1.1 Background .......................... 1
   1.2 Thesis Objectives ................... 3
   1.3 Outline of the Thesis ............... 4

2 Overview of Spatial Kinetics Methods ...... 5
   2.1 The Neutron Diffusion Equation ....... 5
   2.2 Spatial Discretization ............... 7
      2.2.1 Coarse-Mesh Flux-Expansion Method .. 8
      2.2.2 Nodal Methods ................... 9
         2.2.2.1 Nodal Expansion Method ........ 16
         2.2.2.2 Analytic Nodal Methods ....... 17
   2.3 Time Integration Methods ............. 18
      2.3.1 Alternating Direction Implicit Method .. 19
      2.3.2 Theta Method ................... 20
      2.3.3 GRK Method ................... 21
      2.3.4 Quasi-Static Method ............ 22

3 POLCA-T Neutron Kinetics Model .......... 23
   3.1 Static Form of the Nodal Equation ..... 24
   3.2 Solution of the Time Dependence ...... 26
   3.3 Numerical Procedure .................. 29
      3.3.1 Flux Inner Iterations .......... 29
      3.3.2 Time-Step Procedure .......... 30
      3.3.3 Cross-Section and Coupling Coefficient Updates 30
      3.3.4 Summary ....................... 32
## Contents

4 Description of the Benchmark Problems 35  
4.1 TWIGL Reactor Kinetics Problem .................... 35  
4.2 LMW Operational Transient .......................... 37  
4.3 LRA BWR Kinetics Problem .......................... 40  
4.4 Benchmarking Procedure ............................. 50  
  4.4.1 POLCA7 Core Modeling ......................... 50  
  4.4.2 POLCA-T Modeling .............................. 53  

5 Numerical Study 55  
  5.1 Studied Parameters ................................. 55  
  5.2 Results ........................................... 56  
  5.3 Conclusions ....................................... 68  

6 Benchmarking Results and Discussion 71  
  6.1 TWIGL Reactor Kinetics Problem .................... 71  
    6.1.1 Ramp Perturbation Transient ................... 72  
    6.1.2 Step Perturbation Transient ................... 76  
  6.2 LMW Operational Transient ........................ 79  
    6.2.1 Static Results ................................. 79  
    6.2.2 Transient Results .............................. 82  
  6.3 LRA BWR Kinetics Problem ......................... 91  
    6.3.1 Static Results ................................. 91  
    6.3.2 LRA Four-Rod Transient ....................... 97  
    6.3.3 LRA Single-Rod Transient ..................... 101  

7 Summary and Conclusions 105  

Bibliography 107
List of Figures

2.2.1 Node and current labeling for nodes in x-direction as an example. ............................................... 12

3.3.1 An overview over processes involved during a kinetics time-step in POLCA-T. .............................. 33

4.1.1 The TWIGL reactor core model. ......................... 36
4.2.1 LMW reactor core: Horizontal and vertical cross-sections. 39
4.2.2 LMW reactor core at the end of the transient. ........ 41
4.3.1 LRA four-rod reactor core model: Horizontal cross-section. 44
4.3.2 LRA reactor core model: Vertical (modified) cross-section. 46

6.1.1 Power versus time for the TWIGL ramp perturbation transient. .................................................. 75
6.1.2 Reactivity profile for the TWIGL ramp perturbation transient. ..................................................... 75
6.1.3 Power versus time for the TWIGL step perturbation transient. ..................................................... 77
6.1.4 Reactivity profile for the TWIGL step perturbation transient. ..................................................... 77
6.2.1 LMW benchmark, initial condition: Normalized assembly power densities in comparison with MGRAC (octal symmetry is applied). ..................................................... 82
6.2.2 LMW benchmark, final condition: Normalized assembly power densities in comparison with MGRAC (octal symmetry is applied). ..................................................... 83
6.2.3 Power versus time for the LMW operational transient. . 84
6.2.4 Reactivity versus time for the LMW operational transient. 84
6.2.5 Rod cusping effect during the LMW transient. ......... 85
List of Figures

6.2.6 LMW transient results with and without cusping correction method AXHOM. .................................................. 86
6.3.1 LRA benchmark problem: Normalized fuel assembly power densities for the initial configuration (octal symmetry is applied). .......................................................... 93
6.3.2 LRA four-rod case: Normalized fuel assembly power densities for the final configuration. ................................. 94
6.3.3 LRA single-rod case: Normalized fuel assembly power density for the final configuration. ................................. 95
6.3.4 Assembly power density map for the final conditions in the single-rod case. The normalized power densities are plotted on the z-axis. .......................................................... 96
6.3.5 LRA four-rod transient: Power density as a function of time. .............................................................................. 98
6.3.6 LRA four-rod transient: Average fuel temperature versus time. ........................................................................ 99
6.3.7 LRA single-rod transient: Power density as function of time. .............................................................................. 101
## List of Tables

4.1.1 Material properties of the TWIGL reactor problem. . . . 38  
4.2.1 LMW operational transient: Material data. . . . . . . . 42  
4.2.2 LMW operational transient: Delayed neutron precursor  
   data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43  
4.3.1 LRA benchmark problem: Material properties. . . . . . 48  
4.3.2 LRA benchmark problem: Delayed neutron precursor  
   data. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49  
4.3.3 Albedo values at the core periphery as applied in POLCA7.  
   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49  
5.1.1 Default values of the described numerical parameters. . 57  
5.2.1 Relative power versus time for the initializing TWIGL  
   ramp perturbation calculation (constant 5 ms case). . . . 58  
5.2.2 Impacts of various kinetics time-step sizes on the solution  
   accuracy. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59  
5.2.3 Impact of the convergence criterion on the solution ac-  
   curacy. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60  
5.2.4 Impact of a varying number of allowed non-linear itera-  
   tions on the solution accuracy. . . . . . . . . . . . . . . . . . . . . 61  
5.2.5 Impact of a varying number of flux inner iterations on  
   the solution accuracy. . . . . . . . . . . . . . . . . . . . . . . . . . 63  
5.2.6 Impact of a reduced number of iterations on the solution  
   accuracy. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64  
5.2.7 Impacts of variable kinetics time-steps on the solution  
   accuracy. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65  
5.2.8 Impacts of maximum permitted relative flux changes  
   over a kinetics time-step. . . . . . . . . . . . . . . . . . . . . . . . . 67  
5.2.9 Impacts of different theta values on the solution accuracy. 69
List of Tables

6.1.1 TWIGL static eigenvalue results for initial conditions. . 73
6.1.2 Relative power versus time for the TWIGL ramp perturbation problem in comparison with other codes (5 ms step size). . . . . . . . . . . . . . . . . . . . . . . . . . . 74
6.1.3 Relative power versus time for the TWIGL step perturbation problem in comparison with different codes. . . . 78
6.2.1 LMW benchmark: Eigenvalue results for both the initial and the final core state. . . . . . . . . . . . . . . . . . . . . . . . . . . 80
6.2.2 LMW operational transient: Examination of the axial homogenization procedure. . . . . . . . . . . . . . . . . . . . . . . . . . . 88
6.2.3 LMW operational transient: Comparison between different codes. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
6.3.1 LRA benchmark problem: Eigenvalues for initial and final core configurations. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
6.3.2 LRA benchmark problem: Temporal mesh (feedback time-step sizes) employed during the transient calculations. . 97
6.3.3 LRA four-rod transient: Comparison of POLCA-T results with other codes. Power densities are given in W/cc.100
6.3.4 LRA single-rod transient: Comparison of POLCA-T results with other codes. Power densities are given in W/cc.103
1 Introduction

“Errare humanum est, sed in errare perseverare diabolicum.”
Attributed to Seneca the Younger (c. 4 BC – AD 65)

1.1 Background

During the recent decades nuclear energy has become an essential part of a safe and cost-efficient power generation industry worldwide. The exploitation of the nuclear fission process, however, is not limited to power generation only. The production of a wide range of radioactive isotopes for medical and industrial purposes as well as the supply of the fundamental materials research with high neutron fluxes and fluences rely almost completely on the utilization of nuclear reactors. Their safe and dependable operation requires a profound knowledge of the underlying physical processes.

For this purpose, the nuclear engineer has a variety of calculation tools (or computer codes) at his/her disposal. These tools are able to simulate the involved processes in a nuclear core and to predict the progression of all those parameters which are crucial for the preservation of safe and stable conditions. At this point, however, one might immediately pose the question of the reliability of the calculated results. In fact, all the computational tools do not represent anything but a man-made “creation” (even though a highly advanced one), which thus might comprise many sources of errors and traps (“Errare humanum est...”). The question becomes even more important in view of the severity of negative consequences arising from abnormal operational conditions in a
reactor core. Such considerations result in an *inevitable requirement* on a proper qualification of the utilized computer code. The qualification implies that the code in question has to be scrutinized against other tools and/or procedures that possess a high degree of reliability.

The code qualification, also known as the verification and validation (V&V) procedure, can be done in different ways. One example is a simulation of an event that already occurred in the past and a subsequent proof for the consistency of the computed results with the real outcome. This process is referred to as the code validation. Validation shall ensure that the code provides an outcome that is required for the consecutive analysis. The question associated with this kind of code qualification might be posed as: “Do we do the right thing?” On the contrary, verification\(^1\) implies a proof of the correct model implementation and its flawless integration into the code. The question associated with verification is rather: “Do we do the thing right?”

This work is concerned with the verification of the computer code POLCA-T [23]. The code that was developed by Westinghouse Electric Sweden AB couples neutron kinetics with thermal-hydraulics for reactor transient analyses with the objective of studying the reactor core behavior during specified transients. POLCA-T utilizes also some procedures implemented in its static counterpart, the three-dimensional nodal core simulator POLCA7 [17, 20].

Before proceeding further, however, one issue has still to be clarified: what is *the* way for a proper verification procedure in view of the complexity of nuclear facilities and the underlying physics as well as the large variety of different scenarios and their interdependence? In other words, how can one assure that the “qualified” code is fit for use in simulating and predicting of any other, imaginable situations?

In order to tackle this hurdle, an advisable approach is the “divide and conquer” strategy. In principle, the code is separated into its individual models and the latter are verified separately. This is especially recom-

---

\(^1\) Verification (from lat. “*veritas*” - truth and “*facere*” - to do): Confirmation of an assertion or an assumption by a perceptive examination or by a reasoned proof [7, 11].
mended for large, coupled computer codes such as POLCA-T. Here, it is reasonable to “divide” the code into its neutronics and thermal-hydraulic parts. In this work, verification of the POLCA-T neutronics is addressed.

1.2 Thesis Objectives

In the previous section it has been concluded that it is reasonable to separate a large coupled three-dimensional code in its modules and verify each of them individually. The objective of the present work is benchmarking of neutron kinetics employed in POLCA-T, Version 2.1. The work that represents a keystone of the overall verification procedure is done by solving several widely used two- and three-dimensional kinetics problems. The results are then compared to those of other, established and well-proven spatial kinetics codes, or to known reference results. Since the thermal-hydraulic module is completely decoupled, it is not the subject of the overall discussion below.\(^2\)

The benchmarking of the POLCA-T kinetics model involves four computational stages:

1. A separate free-standing POLCA7 static simulation of the initial and final steady-state core. This step verifies that a selected benchmark problem is correctly modeled.

2. A POLCA-T static simulation of the steady-state core. This step serves to confirm that the integration of POLCA7 into POLCA-T is correct and that the input data to POLCA7 is utilized correctly.

3. A POLCA-T transient simulation of the initial steady-state core without any perturbation. This is the so-called null-transient (or zero-transient), which serves to confirm that the time-dependent kinetics equations are consistent with the static equations for the steady-state situation.

\(^2\)In fact, this is not completely true as it will be shown in Chapter 6: the LRA benchmark is fully dependent on the fuel temperature feedback but this feedback was coded in an own sub-routine being thus not part of the POLCA-T thermal-hydraulics.
1 Introduction

4. A POLCA-T transient simulation of the actual transient situation. This serves as basis for comparison of the POLCA-T kinetics results against those of other codes.

Though, before going on with the benchmarking, impacts of different numerical options on the solution accuracy are evaluated by using the two-dimensional TWIGL problem (for a detailed description of the problem it is referred to Chapter 4). In particular, impacts of such parameters as the convergence criteria, the number of iterations and the time-step size are examined. This numerical study is also used to obtain some indications for preferable numerical options that are to be used in the benchmarking calculations.

1.3 Outline of the Thesis

Any discussion on the performance of POLCA-T and the comparisons with other codes requires at least some basic understanding of underlying mathematical models employed in these codes. Hence, Chapter 2 describes briefly the different solutions of the time-dependent neutron diffusion equation with focus on the reference codes used in this work. Similarly, the neutron kinetics model of POLCA-T is outlined in Chapter 3. Spatial-kinetics benchmark problems considered in this work as well as the benchmarking procedure are detailed in Chapter 4. The numerical study, which describes the impact of different numerical options on the results is covered in Chapter 5. Chapter 6 addresses the benchmarking results and their discussion. Finally, Chapter 7 summarizes the work and draws the most significant conclusions.
2 Overview of Spatial Kinetics Methods

This chapter addresses some key features of various spatial kinetics methods, which are employed in the reference codes with the time-dependent neutron diffusion equation as the starting point. It is not only the matter of completeness that makes it reasonably to discuss this topic. It is also a significant property of this benchmarking that considers results computed with a variety of different mathematical models. Though, the chapter does not represent any attempt to describe the existing methods in detail. A comprehensive review on present solutions of the neutron diffusion equation can be found in such excellent articles as [16] and [28] instead.

2.1 The Neutron Diffusion Equation

The distribution of a group of neutrons in time, energy and space is exactly described by the well-known Boltzmann transport equation. However, for reactor core analyses on a routine basis, and, in particular, for dynamic calculations, the equation is hardly applicable due to its complexity. An adequate approximation to the equation is represented by the neutron diffusion equation. In fact, the major part of the commonly used reactor computational tools is based on solutions of the latter even though similar methods of resolution were applied to the transport equation directly [16].

As already pointed out above, the starting point for the discussed space-time discretization methods is the time-dependent neutron diffusion equation. The equation consists of coupled differential equations for
2 Overview of Spatial Kinetics Methods

neutron flux and delayed neutron precursor concentration. For an energy group \( g \) in a system with \( G \) energy groups and \( L \) delayed neutron families, the neutron diffusion equation is formulated as following:

\[
\frac{1}{v_g} \cdot \frac{\partial}{\partial t} \phi_g(\vec{r}, t) = \nabla \cdot D_g(t) \nabla \phi_g(\vec{r}, t) - \Sigma_{t,g}(\vec{r}, t) \phi_g(\vec{r}, t) \\
+ \frac{1}{k_{eff}} \chi_p^g (1 - \beta) \sum_{g'=1}^{G} \nu \Sigma_{f,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \\
+ \sum_{g'=1}^{G} \Sigma_{s,g,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \\
+ \sum_{l=1}^{L} \chi_d^g \lambda_l C_l(\vec{r}, t) \tag{2.1.1a}
\]

\[
\frac{\partial C_l(\vec{r}, t)}{\partial t} = \frac{1}{k_{eff}} \beta_l \sum_{g=1}^{G} \nu \Sigma_{f,g}(\vec{r}, t) \phi_g(\vec{r}, t) \\
- \lambda_l C_l(\vec{r}, t) \tag{2.1.1b}
\]

Here:
\[
\begin{align*}
\frac{1}{v_g} & \quad \text{reciprocal neutron average speed of energy group } g \\
\phi_g & \quad \text{neutron flux in energy group } g \\
D_g & \quad \text{diffusion coefficient in energy group } g \\
\Sigma_{t,g} & \quad \text{total collision cross-section for neutrons in group } g \\
k_{eff} & \quad \text{eigenvalue of the equation under the steady-state conditions} \\
\chi_p^g & \quad \text{fraction of prompt neutrons emitted into the group } g \\
\beta & \quad \text{total delayed neutron fraction, with } \beta \equiv \sum_{l=1}^{L} \beta_l, \text{ where } \beta_l \text{ is the delayed neutron fraction for the precursor } l \\
\nu \Sigma_{f,g} & \quad \text{product of the averaged number of neutrons produced in a fission process and the fission cross-section for neutrons in group } g
\end{align*}
\]
2.2 Spatial Discretization

\[ \Sigma_{s,gg'} \] scattering cross-section for neutrons in group \( g' \) that are emitted into the group \( g \)

\[ \chi^d_{g,l} \] fraction of delayed neutrons of the precursor \( l \) emitted into the group \( g \)

\[ \lambda_l \] decay constant of the precursor \( l \)

\[ C_l \] concentration of the delayed neutron precursor \( l \)

In the next sections, the main emphasis is placed on the discussion of the solution of the neutron flux equation, but the precursor equation is also treated wherever it is indispensable for the overall discussion. The equation for flux is a second-order partial differential equation (PDE) and can be solved by discretization in space first, followed by discretization in time.

2.2 Spatial Discretization

Classical solutions of the few-group diffusion equations were based on finite-difference (FD) techniques [16]. These methods imply basically an approximation of the differential operator \( \nabla \cdot D \nabla \) in the vicinity of the point of interest by a low-order Taylor series expansion. In reactor core applications, and in particular in the so-called mesh-centered FD method [22], the point of interest represents the center point of a finite volume \( V^n \), and the sum of these finite volumes (or boxes) yields the total core volume. Material properties (i.e., the diffusion coefficients and the cross-sections) are assumed to be spatially homogeneous within each box. In order to minimize the spatial truncation error, the size of the volume elements should be kept sufficiently small. As a rule of thumb, the mesh spacing is chosen to be not larger than the diffusion length of thermal neutrons, which lies approximately in the range of 2–3 cm [25].

The rapid expansion of nuclear power and the concomitant requirements on larger core power output in the ’60s and ’70s necessitated also a significant increase in the core volume. A proper application of FD techniques would imply a substantial step-up in the number of
2 Overview of Spatial Kinetics Methods

boxes and thus, in unacceptably high computational efforts. An obvious improvement is the extension of the boxes to the larger ones without, however, sacrificing the overall accuracy of the results.

2.2.1 Coarse-Mesh Flux-Expansion Method

Methods that employ models with larger volume elements (i.e., the mesh size is definitely larger than the diffusion length of neutrons and corresponds to dimensions of a LWR fuel assembly on radial plane) normally exploit the fact that often, one is primarily interested in assembly-wise power distributions. To reduce the spatial truncation errors when employing such coarse-mesh methods one has to apply higher order spatial approximations [22].

One example of such coarse-mesh applications is the flux-expansion method employed in the QUABOX/CUBBOX transient code [15]. In this method, the principle unknown is the box center-point flux that in conjunction with the six surface fluxes at the mid-points of each of the box sides form the points of support for the basis functions that approximate the spatial flux distribution within each box.

These basis functions can be formulated as various quadratic (as in QUABOX) or cubic (as in CUBBOX) polynomials.\(^1\) The coefficients of the quadratic polynomials are fully specified by the flux values of the seven supporting points. Higher order approximations, however, require additional side conditions in order to determine the polynomial coefficients (as the system becomes indeterminate). Hereby, residual weighting techniques such as the Galerkin method are used to fully specify these conditions.

By differentiating a considered polynomial at the box boundaries and applying continuity conditions across the box interface, the surface fluxes are subsequently expressed by center-point fluxes and coupling coefficients establishing so the coupling between two adjacent boxes.

\(^1\)Hereby, it was shown [15] that the performance of CUBBOX in terms of calculated errors and computation times is considerably better than that of QUABOX. Hence, the present work alludes mostly to CUBBOX as the reference code.
The procedure finally results in a system of equations that has to be advanced numerically from one time level to another.

### 2.2.2 Nodal Methods

Another approach to solve the neutron diffusion equation is developed in the so-called consistent nodal methods where volume elements are known as nodes. Like the coarse-mesh flux expansion method, these techniques divide the core total volume into a number of non-overlapping nodes with node dimensions being in the range of the dimensions of a fuel assembly (on the radial plane) or even larger. These “modern” nodal methods are consistent in the sense that with infinite mesh refinement they produce the numerically exact solution to the diffusion equation. In contrast to the coarse-mesh flux expansion method, the principal unknown in all here discussed nodal methods is the node-averaged flux, while it is the center point flux in the first-mentioned method.

Furthermore, in modern nodal methods, the nodalization process relies on a preceding homogenization procedure that, as the name suggests, reduces within each node the heterogeneity effects existent in the material properties to equivalent, node-averaged (or homogeneous) levels. Basically, this is achieved by abandoning the requirement for continuity in the homogeneous, face-averaged flux (i.e., flux computed with homogenized parameters) across the node interface. Yet, it is essential to be able to reproduce the same reaction rates and leakages that would be computed with the heterogeneous transport calculations.

In order to close the “gap” between the heterogeneous and homogeneous conditions, additional parameters have to be introduced. These parameters are commonly known as the discontinuity factors. They establish coupling relationships between the homogeneous face-averaged fluxes of two adjacent nodes by ensuring continuity in heterogeneous flux instead (which is also more comprehensible from the “physical” point of view). A review of prevailing homogenization procedures can

---

2 The homogenization procedure mentioned here is known as the “General Equiv-
be found in [27].

The nodal method procedure continues with integrating the Eq. 2.1.1a over the node volume $V^n$. By applying the general Fick’s Law that describes the diffusion of particles in a medium

$$\vec{J}(\vec{r}, t) = -D(\vec{r}, t) \nabla \phi(\vec{r}, t) \quad (2.2.1)$$

with $\vec{J}$ as the net current of particles and $D$ as the diffusion coefficient, one obtains the so-called nodal balance equation, which in Cartesian coordinates and for a group $g$ and a node $n$ can be written as:

$$\frac{1}{v_g^n} \cdot \frac{d}{dt} \Phi_g^n(t) = -\frac{1}{h_x} \left( J_{gx}^{n+}(t) + J_{gx}^{n-}(t) \right) - \frac{1}{h_y} \left( J_{gy}^{n+}(t) + J_{gy}^{n-}(t) \right)$$

$$- \frac{1}{h_z} \left( J_{gz}^{n+}(t) + J_{gz}^{n-}(t) \right) + Q^n_g(t)$$

$$- \Sigma_{t,g}^n(t) \Phi_g^n(t) \quad (2.2.2)$$

with

$$Q^n_g(t) = \frac{1}{k_{eff}} \sum_{g'=1}^{G} \chi_{g',p}^{n,p} (1 - \beta) \nu \Sigma_{f,g'}^n(t) \Phi_{g'}^n(t)$$

$$+ \sum_{g'=1}^{G} \Sigma_{s,gg'}^n(t) \Phi_{g'}^n(t)$$

$$+ \sum_{l=1}^{L} \chi_{g,l}^{n,d} \lambda_l^n C_l^n(t) \quad (2.2.3)$$

“Partial Current Equivalence Theory” [27] that – as it is outlined above – is based on preservation of the heterogeneous face-averaged flux across a node interface. Alternatively, the homogenization can also be achieved by preservation of partial currents (“Partial Current Equivalence Theory”). The advantage of the latter method is that it enables an additional degree of freedom during the homogenization procedure [21, 22].
2.2 Spatial Discretization

\[ J_{gx}^{n \pm} (t) = \frac{1}{h_y h_z} \int_{-h_y/2}^{h_y/2} \int_{-h_z/2}^{h_z/2} J_{gx}^n (x = \pm h_x/2, y, z, t) \, dy \, dz \]  \hspace{1cm} (2.2.4)

\[ J_{gy}^{n \pm} (t) = \frac{1}{h_x h_z} \int_{-h_x/2}^{h_x/2} \int_{-h_z/2}^{h_z/2} J_{gy}^n (x, y = \pm h_y/2, z, t) \, dx \, dz \]  \hspace{1cm} (2.2.5)

\[ J_{gz}^{n \pm} (t) = \frac{1}{h_x h_y} \int_{-h_x/2}^{h_x/2} \int_{-h_y/2}^{h_y/2} J_{gz}^n (x, y, z = \pm h_z/2, t) \, dx \, dy \]  \hspace{1cm} (2.2.6)

\[ x, y, z \in V^n \]

and

\[ \Phi_g^n \]
node-averaged flux in group \( g \), defined as

\[ \Phi_g^n (t) \equiv \frac{1}{V^n} \int_{V^n} \phi (x, y, z, t) \, dV \]

\[ C_l^n \]
node-averaged precursor concentration of the delayed neutron family \( l \), defined as

\[ C_l^n (t) \equiv \frac{1}{V^n} \int_{V^n} C_l (x, y, z, t) \, dV \]

\[ J_{gu}^{n+}, J_{gu}^{n-} \]
group \( g \) face-averaged net currents on the right (+) and the left (-) node side respectively, with \( u \in x, y, z \). The net currents here are defined as outgoing normal net currents as seen from the node under consideration. See also Figure 2.2.1.

\[ h_u \]
node width in the \( u \)-direction with \( u \in x, y, z \). See also Figure 2.2.1.

The nodal balance equation for the delayed neutron precursors is correspondingly:
2 Overview of Spatial Kinetics Methods

Figure 2.2.1: Node and current labeling for nodes in x-direction as an example.

\[ \frac{dC^n_l}{dt} = \frac{1}{k_{eff}} \beta^n_l \sum_{g=1}^{G} \nu \Sigma^n_{f,g} (t) \Phi^n_g (t) - \lambda^n_l C^n_l (t) \]  

(2.2.7)

The nodal fluxes and precursor densities can be treated as the primary unknowns of the problem provided a relationship between the node-averaged fluxes and the node side-averaged net currents can be established.

In order to find this relationship, nearly all modern nodal methods apply the transverse integration procedure. By this procedure, one arrives at three coupled, one-dimensional equations where each of them can be solved analytically. The procedure implies separate integration of the Eq. 2.1.1a over the area orthogonal to the direction of interest. For example, the integration transverse to the x-direction, yields:

\[ \frac{1}{v^n_g} \frac{\partial}{\partial t} \Phi^n_{gx} (x,t) = D^n_g (t) \frac{\partial^2}{\partial x^2} \Phi^n_{gx} (x,t) + Q^n_{gx} (x,t) \]

\[ -\Sigma^n_{i,g} (t) \Phi^n_{gx} (x,t) - L^n_{gx} (x,t) \]  

(2.2.8)

with \( \Phi^n_{gx} \) being the transverse to the x-direction integrated flux, defined
as:

\[
\Phi_{nx}^n (x, t) \equiv \frac{1}{h_x h_y} \int_{-h_z/2}^{h_z/2} \int_{-h_y/2}^{h_y/2} \Phi_g^n (x, y, z, t) \, dy \, dz
\]

Furthermore, \( L_{gx}^n \) is the transverse leakage term that accounts for the leakages in the \( y \)- and \( z \)-directions:

\[
L_{gx}^n (x, t) = -\frac{1}{h_y h_z} \int_{-h_z/2}^{h_z/2} D^n_g (t) \left\{ \frac{\partial}{\partial y} \Phi_g^n (x, y = \frac{h_y}{2}, z, t) - \Phi_g^n (x, y = -\frac{h_y}{2}, z, t) \right\} \, dz
\]

\[
-\frac{1}{h_z h_y} \int_{-h_y/2}^{h_y/2} D^n_g (t) \left\{ \frac{\partial}{\partial z} \Phi_g^n (x, y, z = \frac{h_z}{2}, t) - \Phi_g^n (x, y, z = -\frac{h_z}{2}, t) \right\} \, dy
\]

(2.2.9)

and \( Q_{gx}^n \) is the transverse integrated source term:

\[
Q_{gx}^n (x, t) = \frac{1}{k_{eff}} \sum_{g' = 1}^{G} \chi_{n,p}^{n-p} (1 - \beta) \nu \Sigma_{f,g' \nu}^n (t) \Phi_{g'x}^n (x, t)
\]

\[
+ \sum_{g' = 1}^{G} \Sigma_{s,gg'}^n (t) \Phi_{g'x}^n (x, t)
\]

\[
+ \sum_{l=1}^{L} \chi_{n,d}^{n,d} \lambda_{lx}^n C_{lx}^n (x, t)
\]

(2.2.10)

Finally, \( C_{lx}^n \) is the transverse integrated precursor concentration (for the delayed neutron family \( l \) and, again, transverse to the \( x \)-direction):
2 Overview of Spatial Kinetics Methods

\[
C_{lx}^n(x,t) \equiv \frac{1}{h_x h_y} \int_{-h_z/2}^{h_z/2} \int_{-h_y/2}^{h_y/2} C_l^n(x,y,z,t) \, dy \, dz
\]

For the other two directions, \( y \) and \( z \), one obtains likewise expressions. Obviously, the transverse integrated equation above resembles its static counterpart apart from the presence of the transverse integrated precursor concentrations (these are contained in the source term \( Q^n_{gx} \)) and the time-derivative term. If the equation can be manipulated in such a manner that the precursor concentrations can be expressed in terms of the fission rate and the time-derivative term eliminated in some way, then the existing steady-state procedures can be used to find the relationship between the node-averaged flux and face-averaged currents even in the kinetics mode. This procedure is in fact applied in many kinetics codes including POLCA-T. Lastly, in order to solve the Eq. 2.2.8 analytically, the shape of the transverse leakage term \( L^n_{gx} \) has to be specified as well.

As concerns the latter, the procedure is similar for the most discussed codes. The approximation of the transverse leakage term by a second-order polynomial was shown to be sufficiently accurate for accounting for the transverse leakages. The polynomial coefficients are chosen in a way that preserves the averaged transverse leakages of a given node and its two adjacent neighbors in a given direction [26, 28]. A somewhat different procedure is applied in the code IQSBOX [28]. Obviously, on core boundaries a simplification of this approach has to be applied.

The elimination of the time-derivative terms as well as the reduction of the node-averaged precursor concentrations consists of the introduction of dynamic frequencies \( \omega_\phi \) and \( \omega_C \) for the time-dependence of flux and the time-dependence of the precursor densities, respectively. This approach is based on the assumption that the temporal dependence of transverse integrated flux or precursor concentration can be approximated by an exponential function. For a given time-step \( t_i \leq t \leq t_{i+1} \), one thus can formulate [23]:
2.2 Spatial Discretization

\[
\Phi^n_{gu}(u, t) \approx \Phi^n_{gu}(u, t_i) \exp \left[ \omega^n_{\phi,g,i+1} (t - t_i) \right]
\]

\[
C^n_{lu}(u, t) \approx C^n_{lu}(u, t_i) \exp \left[ \omega^n_{C,l,i+1} (t - t_i) \right]
\]

\( u \in x, y, z \)

With the time-step size \( \Delta t_{i+1} = t_{i+1} - t_i \), the dynamic frequencies are defined as [23]:

\[
\omega^n_{\phi,g,i+1} \equiv \frac{1}{\Delta t_{i+1}} \ln \left( \frac{\Phi^n_g(t_{i+1})}{\Phi^n_g(t_i)} \right) \tag{2.2.11a}
\]

and accordingly:

\[
\omega^n_{C,l,i+1} \equiv \frac{1}{\Delta t_{i+1}} \ln \left( \frac{C^n_l(t_{i+1})}{C^n_l(t_i)} \right) \tag{2.2.11b}
\]

The transverse integrated precursor concentration equation

\[
\frac{dC^n_{lx}(x, t)}{dt} = \frac{1}{k_{eff}} \beta^n_l \sum_{g=1}^{G} \nu \Sigma^n_{f,g}(t) \Phi^n_{gx}(x, t) - \lambda^n_l C^n_{lx}(x, t) \tag{2.2.12}
\]

can now be solved for \( C^n_{lx} \) and the obtained expression is substituted for the precursor concentration term included in the source term \( Q^n_{gx} \). By also replacing the time-derivative term in the transverse integrated flux equation, one arrives at a static diffusion equation that resembles the steady-state diffusion equation.

The procedure so far was rather similar for almost all nodal methods. The differences lie rather in the way of finding a relationship between the face-averaged net (or partial) currents and the node-averaged flux. This sought relationship can generally be described by the following expression:
\[ J_{gu}^{n+} = W_{gu}^{n+} \Phi_g^n - W_{gu}^{m-} \Phi_g^m \]  

(2.2.13)

where \( W_{gu}^{n+} \) is a coupling coefficient from the node \( n \) in the direction of the node \( m \) lying to the right of the node \( n \) in the \( u \)-direction. Correspondingly, \( W_{gu}^{m-} \) is a coupling coefficient from the node \( m \) in the direction of the node \( n \) lying to the left of the node \( m \). This expression can now be substituted into the nodal balance equation (Eq. 2.2.2) yielding so a time-dependent ordinary differential equation (ODE) with the node-averaged flux and precursor densities as the only primary unknowns. Summing over all interfaces of the node \( n \), where the node \( m \) represents all to the node \( n \) adjacent nodes, this ODE can be written as:

\[
\frac{1}{v_g^n} \cdot \frac{d}{dt} \Phi_g^n(t) = \left( - \sum_m W_{nm}^{mn}(t) \frac{h_{mn}^n}{h_{mn}^n} - \Sigma_{t,g}^n(t) \right) \Phi_g^n(t) \\
+ \sum_m W_{mn}^{nm}(t) \frac{h_{mn}^m}{h_{mn}^m} \Phi_g^m(t) + Q_g^n(t) 
\]

(2.2.14)

with \( h_{mn}^n \) being the height of the node \( n \) in the direction of the node \( m \).

With respect to the method used for finding the relationship above, one can generally distinguish between following two approaches: the Nodal Expansion Method (NEM) and the Analytic Nodal Method (ANM). The former will be discussed first.

### 2.2.2.1 Nodal Expansion Method

The Nodal Expansion Method belongs to the class of polynomial methods and proceeds with approximating the solution for the transverse to the \( u \)-direction integrated flux (Eq. 2.2.8) by expanding the latter with a higher-order polynomial. With respect to the \( x \)-direction, the expression becomes then [4]:

16
2.2 Spatial Discretization

\[ \Phi_{nx}^n(x) = \Phi_g^n + \sum_{b=1}^{B} a^n_b f_b^n(x) \]  \hspace{1cm} (2.2.15)

with \( \Phi_{nx}^n \) as the node-averaged flux and \( f_b^n(x) \) being a polynomial of order \( b \). By taking the derivative of this expansion, a relationship for net current can be obtained. The expansion coefficients \( a^n_1, a^n_2, \ldots, a^n_B \) are then determined by applying continuity relations at node interfaces in combination with a weighted residual procedure such as the moment weighting, described in [16].

Nodal expansion method codes, which it is referred to in this benchmarking are IQSBOX [8], SPANDEX [3] and - to some extent - CONQUEST [9].

2.2.2.2 Analytic Nodal Methods

As its name suggests, the Analytic Nodal Method (ANM) elaborates the necessary relationship between the net currents and node-averaged fluxes in an analytic way. The method does not implement any approximations (apart from the transverse leakage shape) or arbitrary values. Since the method will be briefly described in Chapter 3, its depiction will be skipped at this point. Along with POLCA7/POLCA-T, the ANM method is employed in codes such as QUANDRY [26] and PANTHER [12, 13].

Some codes like AETNA and NEREUS utilize a combination of both polynomial and analytic nodal methods. In these codes, the relationship between currents and flux is derived in an analytic way while the source term that includes the reaction rates for fission and down-scattering is expanded with a polynomial. This approach is sometimes referred as the semi-analytic method. See [14, 29] for details.
2 Overview of Spatial Kinetics Methods

2.3 Time Integration Methods

So far, the derivation resulted in a nodal, time-dependent ordinary differential equation (Eq. 2.2.14) where the relationship between currents and flux (Eq. 2.2.13) has been derived in a particular way. The time integration can be performed by using a variety of numerical schemes. Any numerical integration process is characterized by two properties. The method convergence defines if the global truncation error approaches zero in case the temporal mesh size diminishes. Method stability addresses the sensitivity of the system to small perturbations in initial boundary conditions [5, 28]. As concerns the transient neutronic calculations, where neutron flux excursions can be several orders of magnitude during split-seconds only, the time integration method in question has to be able to achieve high-accuracy results (i.e., with sufficiently tight convergence criteria) by using acceptably large time-steps (in terms of computational efforts). Therefore, the unconditionally stable implicit methods possesses substantial assets in comparison to the explicit methods and thus are preferably used in neutron kinetics applications.

This section will briefly describe some time integration methods with the emphasis on concepts being employed in the reference codes. The starting point for the ongoing discussion is the compact matrix formulation of the time-dependent flux vector in a system with $G$ groups and $N$ nodes:

$$\frac{d}{dt}\Phi = A\Phi + \Gamma$$ (2.3.1)

Here, $A$ is a $N \times N$ nodal block matrix (operator), which includes cross-section and coupling coefficient data. Each matrix element consists of $G \times G$ energy group sub-matrices. $\Phi$ is an $N \times 1$ flux vector with elements comprising $G \times 1$ subvectors. $\Gamma$ is an $N \times 1$ source vector that accounts for the delayed neutrons contribution with elements consisting of $G \times 1$ subvectors as well. Note that the nodal block matrix, which is sometimes referred to as the coefficient matrix, is dependent on the flux solution itself.
2.3 Time Integration Methods

2.3.1 Alternating Direction Implicit Method

Originally, the Alternating Direction Implicit Method (ADI) was developed for solving the two-dimensional, time-dependent heat equation and was shown to be unconditionally stable regardless of the time-step size [5, 10, 30]. The ADI method starts with the decomposition of the operator $A$ in $x$- and $y$-dependent matrices $A_x$ and $A_y$ and a directionally independent matrix $E$. The direction-dependent matrices contain basically the operators $(\partial/\partial x) D_g (\partial/\partial x)$ and $(\partial/\partial y) D_g (\partial/\partial y)$, while cross-section and neutron speed terms are combined to the directionally independent matrix.

The solution is now advanced in two partial time-steps with the size $\Delta t_{i+1/2}$, where $\Delta t_{i+1} = t_{i+1} - t_i$. During the first time-step from the time-point $t_i$ to $t_{i+1/2}$, one of the two spatial variables is processed explicitly (in this example, it is the $x$-dependent operator) and the other variable (the $y$-dependent operator) implicitly:

$$\Phi(t_{i+1/2}) = \left[ I - \frac{\Delta t_{i+1}}{2} \left\{ A_y \left( t_{i+1/2} \right) + \Theta_E E(t_{i+1/2}) \right\} \right]^{-1} \times \left\{ I + \frac{\Delta t_{i+1}}{2} \left( A_x \left( t_i \right) + (1 - \Theta_E) E(t_i) \right) \right\} \times \Phi(t_i) + \frac{\Delta t_{i+1}}{2} \left\{ \Theta R \Gamma \left( t_{i+1/2} \right) \\ + (1 - \Theta R) \Gamma(t_i) \right\}$$ (2.3.2a)

During the second partial time-step (from the time-point $t_{i+1/2}$ to $t_{i+1}$), the process is reversed, i.e., the second variable (the $y$-dependent operator) is treated explicitly and the first one (the $x$-dependent operator)

---

3For the sake of simplicity, only the two-dimensional case is considered in this example.
2 Overview of Spatial Kinetics Methods

implicitly:

$$
\Phi(t_{i+1}) = \left[ I - \frac{\Delta t_{i+1}}{2} \left\{ A_x (t_{i+1}) + \Theta_E E(t_{i+1}) \right\} \right]^{-1} \\
\times \left[ \left\{ I + \frac{\Delta t_{i+1}}{2} \left( A_y \left( t_{i+1/2} \right) \right) + (1 - \Theta_E) E \left( t_{i+1/2} \right) \right\} \right] \\
\times \Phi \left( t_{i+1/2} \right) + \frac{\Delta t_{i+1}}{2} \left\{ \Theta_E \Gamma \left( t_{i+1} \right) \right\} + \Delta t_{i+1} \left\{ \Theta \Gamma \left( t_{i+1} \right) + (1 - \Theta) \Gamma \left( t_i \right) \right\}
$$

(2.3.2b)

where $\Theta_E$ and $\Theta_F$ are scalar weighting factors in the range $0 \leq \Theta \leq 1$ whereby the former can be individual for any matrix included in $E$.

The ADI method was extended by Langenbuch et al. [15] to three-dimensional systems and the so established scheme was employed in the CUBBOX code. In the three-dimensional case, a time-step is split into three partial time-steps and the three spatial variables are sequentially treated implicitly one at a time [15].

2.3.2 Theta Method

The theta method is an established and – in terms of the solution strategy – flexible time integration method [23]. The method is also the choice for POLCA-T and QUANDRY. It starts with a similar procedure as the previously described ADI method, viz., with approximating the integral of the right hand side (see Eq. 2.3.1) by some form of trapezoidal rule. In contrast to the ADI method, however, the matrix $A$ is not decomposed in its one-dimensional operators. Another difference is that a time-step is not separated into partial time-steps. Thus, one can write similarly to the Eqs. 2.3.2a and 2.3.2b:

$$
\Phi(t_{i+1}) = \left[ I - \Theta \Delta t_{i+1} A(t_{i+1}) \right]^{-1} \\
\times \left[ \left\{ I + (1 - \Theta) \Delta t_{i+1} A(t_i) \right\} \Phi \left( t_i \right) \right] \\
+ \Delta t_{i+1} \left\{ \Theta \Gamma \left( t_{i+1} \right) + (1 - \Theta) \Gamma \left( t_i \right) \right\}
$$

(2.3.3)
Here, $\Theta$ is again a scalar weighting factor in the range of $0 \leq \Theta \leq 1$. For $\Theta = 0$, one obtains the explicit Euler scheme, while setting $\Theta = 1$ yields the fully implicit backward Euler method [23]. According to [23, 28], the fully implicit method is unconditionally stable for any time-step size, thus making it to an absolutely preferable choice.

In contrast to the ADI method, where the time-consuming computation of the inverse of the coefficient matrix $A$ is eased by splitting it into its one-dimensional parts, codes relying on the theta method typically decompose the matrix in its upper, diagonal and lower parts. The utilization of theta method for flux iterations is described in more detail in Chapter 3, where the time integration procedure of POLCA-T is discussed.

### 2.3.3 GRK Method

Classified as explicit methods, the original Runge-Kutta methods (GRK) necessitated small time-steps in order to achieve the required stability. Hence, the methods were initially not widely used in neutron kinetics [28]. The development of implicit Runge-Kutta methods, such as the Kaps-Rentrop GRK, has entailed larger potential for their utilization in core simulators. In general, the solution for flux at $t_{i+1}$ is found by solving the implicit fourth-order GRK [3, 28]:

$$\Phi(t_{i+1}) = \Phi(t_i) + \sum_{n=1}^{4} c_n k_n(t_{i+1})$$  \hspace{1cm} (2.3.4)

Here, one can notice an analogy to the classical fourth-order GRK, where $k_n$ is calculated explicitly [5], while in the discussed example it is treated implicitly. The coefficient $c_n$ is a problem independent fixed constant. The $k_n(t_{i+1})$ vectors are then found by the procedure described in [3], where the implementation of the implicit GRK method in the SPANDEX code is briefly discussed. The code also exploits the possibility for an automatic time-step selection, which is established by comparing the maximum local truncation error with a user-specified tolerance. The step-size is adjusted dependent on the error and the solution order [3].
2 Overview of Spatial Kinetics Methods

2.3.4 Quasi-Static Method

Another time integration procedure is employed in the quasi static method that is chosen for the CONQUEST code. Here, one originally pursued the objective to reduce the number of spatial calculations during the time integration process. This aim is met by factorizing the flux function into a product of two functions: the space and time dependent shape function $\psi(\vec{r}, t)$, computation of which requires an expensive determination of the coefficient matrix inverse, and the time dependent amplitude function $\varphi(t)$:

$$\phi(\vec{r}, t) = \psi(\vec{r}, t) \varphi(t)$$

(2.3.5)

It was observed that the shape function varies more slowly with time, and hence, does not need to be computed as frequent as the amplitude function. Consequently, larger time-step sizes can be used for spatial calculations reducing so the computational efforts. For an extensive description of the method utilization it can be referred to [9], for example.
3 POLCA-T Neutron Kinetics Model

The subject of discussion in the present work is the three-dimensional, transient core simulator POLCA-T. The code was developed by Westinghouse Electric Sweden AB mainly for BWR applications but it can also be used for PWR-transients. Basically, the code consists of two models: the neutron kinetics and the thermal-hydraulics model. The non-linear coupling scheme between neutronics and thermal-hydraulics is based on a consistent fixed point iteration that is fully converged at each feedback time-step. At the end of a feedback time-step parameters such as power, fuel temperature, coolant density, etc. are checked for fulfillment of specified convergence criteria. By this means, it becomes also possible to run the both processes individually, which is also exploited in the present work. Since the thermal-hydraulics model is not of interest in this thesis, the emphasis is placed on neutron kinetics model solely.

The kinetics model is based on the solution of the already mentioned neutron diffusion equation that results in the global balance equation and the subsequent time integration of the latter. In order to obtain nodal coupling coefficients that link the node-averaged flux with the face-averaged currents, POLCA-T is supported by the core simulator POLCA7, or, to be more precisely, by its neutronics model NEU4 [23]. In this chapter, the derivation of the final global balance equation as well as the numerical solution thereof is described more in detail.
3 POLCA-T Neutron Kinetics Model

3.1 Static Form of the Nodal Equation

In the previous chapter some methods for solving the time-dependent neutron diffusion equation were presented. The emphasis was placed mainly on techniques employed in those codes that are referred to in this benchmarking. It was also mentioned that the neutronics model in POLCA7 is classified as the Analytic Nodal Method. At present, two neutronics models are employed: NEU3 and NEU4. In this work, both are used for their specific purposes: while steady-state calculations in POLCA7 utilize the NEU3 option, transients are run with the NEU4 model as their “support”. Both models, however, apply the same transverse integrating procedure, which was named in Section 2.2.2 and they consequently end up in the three one-dimensional, transverse integrated equations (see Eq. 2.2.8 as an example for one direction). The respective transverse leakage terms are approximated by a second-order Legendre polynomial, as described in [17]. The remaining unknown in the nodal balance equation (Eq. 2.2.2), which is the basis for the overall solution, is the coupling between the node-averaged flux $\Phi^n_g$ and the face-averaged net currents $J_{gu}^{n\pm}$ with $u \in x, y, z$.

In order to find this relationship, the transverse integrated (and time-independent) equation is transformed into a dimensionless form like [20]:

$$\frac{d^2}{d\xi^2} \Phi^n (\xi) + \left( b^n_\xi \right)^2 \Phi^n (\xi) = Q^n (\xi) \quad (3.1.1)$$

Hereby, $\xi$ is a dimensionless spatial variable and $b^n_\xi$ the material buckling matrix, which contains the diffusion coefficient and all cross-section terms. Transverse leakages and intra-nodal cross-section variations\(^1\) are combined in the source term $Q^n (\xi)$.

In this form, however, the equation is not applicable for POLCA-T: it is the transient behavior expressed by the time derivative that has still

\(^1\)The intra-nodal dependence of the cross-sections, however, is not exploited in POLCA-T [23].
3.1 Static Form of the Nodal Equation

to be “integrated” into the static solution. As already named in Section 2.2.2, this can be achieved by introducing the dynamic frequencies \( \omega_\phi \) and \( \omega_C \) (see also Eqs. 2.2.11a and 2.2.11b), which are implemented into the buckling matrix \( \hat{b}^n_\xi \) as it is shown in [23]. Then, the static form of the transverse integrated kinetics equation resembles its steady-state counterpart, thus enabling the reuse of the NEU4 neutronics procedures for computing nodal coupling coefficients at any time \( t \).

The differential equation above is solved analytically by involving node boundary conditions, surface-averaged fluxes and currents. Different buckling matrix functions are computed by a preceding diagonalization of the dimensionless buckling matrix through similarity transformation (see also [19]). All the above described manipulations are not subject of this work. Instead, it is referred to [17] as concerns the solution procedure employed in NEU3 and [20] as for the NEU4 option. In this context, main differences between the two options should also be mentioned. The so-called fixed-source problem that is essential for solving the kinetics equations can only be “handled” by NEU4. Furthermore, NEU4 employs an algorithm with the objective to avoid (or, at least, to minimize) results containing negative, i.e. non-physical, fluxes [22].

Still, regardless of the neutronics option employed, the solution ends up in the following non-linear relationship between node-averaged fluxes and currents of the two nodes \( m \) and \( n \) [20]:

\[
\vec{J}^{mn} = \begin{cases} 
\hat{W}^{mn} \vec{\Phi}^n - \hat{W}^{nm} \vec{\Phi}^m & \text{for non-peripheral node sides} \\
\hat{W}^{nm} \vec{\Phi}^n & \text{for peripheral node sides} 
\end{cases} \tag{3.1.2}
\]

The notation \( mn \) indicates the direction from the node \( n \) to the node \( m \). \( \hat{W}^{mn} \) and \( \hat{W}^{nm} \) are two \( G \times G \) matrices, containing coupling coefficients from the node \( n \) in the direction of the node \( m \) and vice versa (observe that \( \hat{W}^{mn} \neq \hat{W}^{nm} \)). The computation of these coupling coefficients represents an important step during the POLCA-T non-linear iterations, as it will be shown in the next two sections.

By inserting the Eq. 3.1.2 into the nodal balance equation (Eq. 2.2.2)
and summing over all interfaces of the node $n$, one obtains the static global balance equation:

$$
\left( \sum_{m \neq n} \frac{\hat{W}_{mn}}{h_{mn}^{n}} - \hat{\Sigma}_{l}^{n} \right) \vec{\Phi}^{n} - \sum_{m \neq n} \frac{\hat{W}_{mn}}{h_{mn}^{n}} \vec{\Phi}^{m} = 0 \quad (3.1.3)
$$

Hereby, $h_{mn}^{n}$ refers to the height of the node $n$ in the direction of its neighbor node $m$. The overall interaction cross-section term $\hat{\Sigma}_{l}^{n}$ includes the fission, scattering and total-collision matrices. The equation represents an eigenvalue problem with flux vectors as eigenvectors. The static eigenvalue $k_{\text{eff}}$ is embedded in the matrix $\hat{\Sigma}_{l}^{n}$.

The subject of the next section is the temporal solution of the time-dependent counterpart to the Eq. 3.1.3.

### 3.2 Solution of the Time Dependence

The exact procedure for deriving the time-dependent global balance equations is extensively described in [23]. Here, it is sufficient to mention that Eq. 3.1.3 can basically be transformed into the time-dependent function by introducing the temporal change of flux and formulating the delayed neutron source $\vec{S}_{n,d}$ explicitly:

$$
\frac{1}{\hat{\nu}^{m}(t)} \frac{d}{dt} \vec{\Phi}^{m}(t) = \vec{S}_{n,d}^{m}(t) - \left( \sum_{m \neq n} \frac{\hat{W}_{mn}(t)}{h_{mn}^{n}} - \hat{\Sigma}_{l}^{n*}(t) \right) \vec{\Phi}^{n}(t) + \sum_{m \neq n} \frac{\hat{W}_{mn}(t)}{h_{mn}^{n}} \vec{\Phi}^{m}(t) \quad (3.2.1)
$$

$$
\vec{S}_{n,d}^{m}(t) = \sum_{l=1}^{L} \chi_{l}^{n,d} \lambda_{l}^{n} C_{l}^{m}(t) \quad (3.2.2)
$$

---

2 The fission matrix accounts also for the steady-state delayed neutron population which can be expressed in terms of a fission rate.
The time-dependent global precursor concentration equation is:

\[
\frac{dC^n_l(t)}{dt} = \frac{1}{k_{eff}} \beta^n_l \left( \nu \Sigma^n_f(t) \right)^T \Phi^n(t) - \lambda^n_l C^n_l(t)
\]  

(3.2.3)

Now, the fission matrix, embedded in the overall interaction cross-section term \( \hat{\Sigma}^n_f(t) \), contains merely the prompt neutron matrix. As the next step, the time-dependent equations are integrated using the theta method.\(^3\) By substituting the time-integrated delayed neutron precursor densities into the time-integrated equation for flux and sorting out all at the current time-point \( t_{i+1} \) unknown terms to the one side of the equation and the - from the previous time-point \( t_i \) - known values to the other side, one finally arrives at the following expression for the neutron balance equation [23]:

\[
\sum_m \left[ \delta^{nm} \frac{1}{\hat{v}^n} + \Theta \Delta t_{i+1} \hat{X}^{nm}(t_{i+1}) \right] \Phi^m(t_{i+1}) = \Delta t_{i+1} \hat{S}^{n,d}_\Theta(t_i)
\]

\[
+ \sum_m \left[ \delta^{nm} \frac{1}{\hat{v}^n} - (1 - \Theta) \right] \times \Delta t_{i+1} \hat{X}^{nm}(t_i)
\]

\[
\times \Phi^m(t_i)
\]

(3.2.4)

with

\[
\hat{X}^{nm}(t) = \delta^{nm} \left( \sum_{m \neq n} \frac{\hat{W}^{mn}(t)}{h_{mn}^n} - \hat{\Sigma}^n_{f,\Theta}(t) \right)
\]

\[
+ (\delta^{nm} - 1) \frac{\hat{W}^{nm}(t)}{h_{mn}^n}
\]

(3.2.5)

\(^3\)The integration of the time-dependent precursor concentration equation uses the same theta-value.
\[ \delta^{nm} = \begin{cases} 
0 & \text{if } n \neq m \\
1 & \text{if } n = m 
\end{cases} \]

\[ \tilde{S}_{\Theta}^{n,d}(t_i) = \sum_{l=1}^{L} \chi_{l}^{-n,d} \frac{\lambda_{l}^{n}}{1 + \Theta \Delta t_{i+1} \lambda_{l}^{n}} C_{l}^{n}(t_i) \quad (3.2.6) \]

\[ \hat{\Sigma}_{l,\Theta}^{n}(t) = \frac{1}{k_{\text{eff}}(1 - \beta)} \left[ \chi^{n,p} + \Theta \Delta t_{i+1} \sum_{l=1}^{L} \frac{\lambda_{l}^{n} \rho_{l}^{n} \chi_{l}^{-n,d}}{1 + \Theta \Delta t_{i+1} \lambda_{l}^{n}} \right] \]
\[ \times \left( \nu \tilde{\Sigma}_{f}^{n}(t) \right)^{T} + \tilde{\Sigma}_{s}^{n}(t) - \hat{\Sigma}_{l}^{n}(t) \quad (3.2.7) \]

For \( \Theta = 1 \), Eq. 3.2.4 is simplified to:

\[ \sum_{m} \left[ \tilde{\delta}_{\Theta}^{nm} + \Delta t_{i+1} \hat{X}^{nm}(t_{i+1}) \right] \tilde{\Phi}_{m}^{n}(t_{i+1}) = \Delta t_{i+1} \tilde{S}_{\Theta}^{n,d}(t_{i}) \]
\[ + \tilde{\delta}_{\Theta}^{nm} \tilde{\Phi}_{m}^{n}(t_{i}) \quad (3.2.8) \]

And the final delayed neutron precursor equation (likewise for \( \Theta = 1 \)) is [23]:

\[ C_{l}^{n}(t_{i+1}) = \frac{1}{1 + \Delta t_{i+1} \lambda_{l}^{n}} \left[ \Delta t_{i+1} \frac{1}{k_{\text{eff}}} \beta_{l}^{n} \left( \nu \tilde{\Sigma}_{f}^{n}(t_{i+1}) \right)^{T} \tilde{\Phi}_{l}^{n}(t_{i+1}) \right] \]
\[ + \frac{1}{1 + \Delta t_{i+1} \lambda_{l}^{n}} C_{l}^{n}(t_{i}) \quad (3.2.9) \]

Formally, the Eq. 3.2.8 can be expressed as

\[ \hat{A}(t_{i+1}) \Phi(t_{i+1}) = q(t_{i}) \quad (3.2.10) \]

that represents the aforementioned fixed-source problem. This formulation facilitates one to proceed in advancing the solution for flux \( \Phi(t_{i+1}) \) at the end of the current time-step under consideration. The way, in which the solution is obtained, is addressed in the next section.
3.3 Numerical Procedure

3.3.1 Flux Inner Iterations

Clearly, the Equation 3.2.10 results in a non-linear procedure where the large block matrix $A$ has to be inverted in order to solve the equation for flux iteratively. As already mentioned above, a reduction of computational effort is reached by splitting the matrix in its upper, lower and diagonal parts that is also done in POLCA-T. Consequently, one needs to invert the diagonal part only. The iterations, which are also called flux inner iterations, can be performed in two modes. The default solution procedure is a scheme where during one iteration step all energy groups are treated simultaneously. It is however also possible to iterate by calculating each group individually. For differences between these two modes, see [23]. The expression for the simultaneous group iteration with $j$ indicating an iteration step is written as:

$$\Phi_{j+1}^{(t_{i+1})} = \left[ A_{DIAG}^{(t_{i+1})} \right]^{-1} \times \left[ q^{(t_{i})} - \left\{ A_{UPP}^{(t_{i+1})} + A_{LOW}^{(t_{i+1})} \right\} \Phi_{j}^{(t_{i+1})} \right]$$

(3.3.1)

Note that at $j = 1$, flux is assigned the value from the preceding non-linear kinetics iteration (update of coupling coefficients), see Section 3.3.3. Solution for flux from the foregoing time-step $\Phi^{(t_{i})}$ is embedded in the fixed-source vector $q$. The flux inner iterations are performed by using the well established Gauss-Seidel method with a checker-board sweep of nodes [20, 23].

The flux inner iterations are not the only iteration procedure involved in the time integration process. Recall that the coefficient matrix itself depends on the solution for flux at $t_{i+1}$. In particular, both the cross-sections and the coupling coefficients are affected by the actual flux progression and thus have to be iterated individually. These iterations are addressed in Section 3.3.3. Before, some comments on the time-step scheme of the overall transient computation have to be spent.
3 POLCA-T Neutron Kinetics Model

3.3.2 Time-Step Procedure

As a reminder it should be noted that a complete transient run is divided into several, so-called feedback time-steps. At the end of each feedback time-step, the neutron kinetics and the thermal-hydraulic solutions are updated in a non-linear iterative fashion (so-called feedback iterations at which data are exchanged between the two processes). During a feedback time-step, both physical processes involve own iteration procedures (such as the above described flux inner iteration, which is evidently embedded in the neutron kinetics). As concerns the neutronics, every feedback time-step can be subdivided into several, equidistant kinetics time-steps (for details on how the process is configured, see [23]). The number of kinetics time-steps can vary between different feedback time-steps. Changes in neutron flux and temporal truncation error or the requirement for a positive solution of the source term and/or precursor densities (in case $\Theta < 1$) may require changes in the kinetics time-step size. The modified kinetics time-step size becomes effective during the next feedback time-step or, alternatively, during the current one, if so required.

After this short introduction to the time-step procedure employed in POLCA-T, the discussion of further iteration levels during a given outer feedback time-step proceeds in the next section.

3.3.3 Cross-Section and Coupling Coefficient Updates

One of the iteration levels involved in the advancement of the temporal solution was already presented in Section 3.3.1, where the Eq. 3.2.10 was solved for flux at the end of the given kinetics time-step. However, coefficient matrix parameters, i.e., cross-sections and coupling coefficients, also vary with time and so have to be recomputed at specified time levels.

Neutron cross-sections depend strongly on material properties, described by thermal-hydraulic parameters, and actual control rod positions. The latter are specified for every feedback time-step by the user or by the reactor process controller and so, can be interpolated linearly during
3.3 Numerical Procedure

a feedback time-step (in other words, the control rod positions are known at each kinetics time-step). The same is (almost) true for material properties but these have to be updated at every feedback iteration affecting so the time gradient of the temperatures and material densities that impact the cross-sections. These time gradients are used to determine the state parameters at the end of a kinetics time-step so that the corresponding end-of-step cross-sections can be updated. Accordingly, the cross-sections are evaluated for every kinetics time-step once, except for the case where the prompt Doppler feedback model is activated. In that case cross-sections may be updated simultaneously with the nodal coupling coefficients.

The coupling coefficients are dependent on the solution itself resulting in the so-called non-linear iterations. During a kinetics time-step, the number of coupling coefficient updates (iterations) is dependent on variations in dynamic frequencies $\omega_\phi$ and $\omega_C$ (which in their turn are functions of flux), and the iterations are terminated upon a specified convergence criterion is being fulfilled or by reaching the maximum number of updates that is a user-specified value.

In this context, some comments on the different convergence criteria applied in POLCA-T have to be made. The convergence criteria are expressed as numerical tolerances that determine a certain level below which the iterations are considered to have converged to a desired degree (and thus, can be stopped). In this work, the tolerances for coupling coefficients iterations, flux inner iterations and temporal truncation error are the most relevant ones. In order to manipulate these values the user has to specify the tolerance for the outer feedback iterations (those iterations at which feedback data between neutronics and thermal-hydraulics are exchanged), denoted as $\varepsilon_{\text{feed}}$. The dependence between the three mentioned tolerances and $\varepsilon_{\text{feed}}$ is as follows:

\[
\varepsilon_{\text{coup}} = \varepsilon_{\text{feed}} \times 10^{-1}
\]

\[
\varepsilon_{\text{inners}} = \min \left( \varepsilon_{\text{feed}} \times 10^{-2}, \frac{\Delta t}{100} \right)
\]
3 POLCA-T Neutron Kinetics Model

\[ \varepsilon_{\text{trunc}} = \max (\varepsilon_{\text{feed}}, 10^{-3}) \]

where \( \varepsilon_{\text{coup}}, \varepsilon_{\text{inners}} \) and \( \varepsilon_{\text{trunc}} \) are the tolerances for the coupling coefficients iterations, the flux inner iterations and the temporal truncation error, respectively.

3.3.4 Summary

A simple way to summarize the above processes taking place during a specific kinetics time-step is the representation by a computational flowchart as shown in Figure 3.3.1. Here, the feedback time points are denoted as \( T_I \) and \( T_{I+1} \) for the last and the next time-step, respectively. Thermal-hydraulic (TH) parameters advanced to \( T_{I+1} \) as well as neutron flux, dynamic frequencies and precursor densities from the foregoing kinetics time-step serve as input values at the beginning of each kinetics time-step. It is also noteworthy that the presented scheme is valid only for \( \Theta = 1 \).
3.3 Numerical Procedure

TH-Data at $T_i$ and $T_{i+1}$
Flux, dynamic
frequencies, prec. densities from the preceding $t_i$

Start kinetics time-step

Evaluate cross-sections at the end of the time-step

Solve Eq. 3.3.1 for flux

Calculate coupling coefficients

Does flux converge?

Yes

Update dynamic frequencies

Update of coupling coeff. required?

Yes

No

End of kinetics time-step

No

Do coupling coeff. converge?

Yes

End of kinetics time-step

No

Figure 3.3.1: An overview over processes involved during a kinetics time-step in POLCA-T.
4 Description of the Benchmark Problems

In this chapter, all considered benchmark problems are extensively described. In detail, these are: the two-dimensional TWIGL Reactor Kinetics Problem, the three-dimensional LWR Operational Transient Problem LMW and the three-dimensional BWR Kinetics Problem LRA. The three benchmark problems are widely used for similar neutron kinetics verification procedures (they do neither require any thermal-hydraulic feedback) so that a broad spectrum of reference solutions is available. The problems cover a wide range of flux changes in space and time, thus scrutinizing effectively the spatial and temporal discretization methods employed in the neutron kinetics model. In the second part of this chapter, the benchmarking procedure is outlined.

4.1 TWIGL Reactor Kinetics Problem

The two-dimensional TWIGL reactor kinetics problem was originally described in the paper of Hageman and Yasinsky [10], where the authors compared the accuracy and stability of the ADI time integration method with the theta method.\(^1\) Even though this two-dimensional problem does not reflect the realistic conditions as the three-dimensional problems do, it eases the comparability of the underlying model as no additional treatment of control rod effects is required, for example. The problem consists of two different transients that employ the same core geometry and materials but differ in the specification of the perturbations. The reactor core model is depicted in Figure 4.1.1.

\(^1\)In this paper, the theta method was referred to as the TWIGL method.
4 Description of the Benchmark Problems

Figure 4.1.1: The TWIGL reactor core model.
The reactor core consists of two different fuel regions: the blanket zone, denoted as “3”, and the seed regions “1” and “2”. The two seed regions do not differ in their material properties but in location where the positive reactivity is inserted. The material properties are compiled in Table 4.1.1 [10]. The reactor has no explicit reflector and zero flux boundaries (on radial plane). It employs two energy groups and one delayed neutron family.

The positive reactivity is inserted in the fuel region “1” by decreasing the thermal absorption cross-section \( \Sigma_{a,2} \). The way of reducing the cross-section separates the problem into two different transients. While in the case of the ramp perturbation the cross-sections are decreased linearly, according to:

\[
\Sigma_{a,2}(t) = \begin{cases} 
\Sigma_{a,2}(t = 0) [1 - 0.11667 \times t] & \text{for } 0 \text{ s} < t \leq 0.2 \text{ s} \\
\Sigma_{a,2}(t = 0) \times 0.97666 & \text{for } t > 0.2 \text{ s}
\end{cases}
\]

the same amount of reactivity is inserted suddenly in the step perturbation case:

\[
\Sigma_{a,2}(t) = \Sigma_{a,2}(t = 0) \times 0.97666 \text{ for } t > 0 \text{ s}
\]

### 4.2 LMW Operational Transient

The Langenbuch-Maurer-Werner (LMW) benchmark problem was initially described in [15]. It reflects an operational transient initiated by control rod movements in a highly simplified PWR. Later, the problem was extended such that it included even the thermal-hydraulic feedback effects. In this work, however, the latter are not considered.

The quarter-symmetric reactor core (as shown in Figure 4.2.1) consists of two different fuel types (with 77 fuel assemblies in the full core) surrounded radially by an explicit water reflector. The fuel assemblies
4 Description of the Benchmark Problems

Table 4.1.1: Material properties of the TWIGL reactor problem.

<table>
<thead>
<tr>
<th>Fuel region</th>
<th>Group g</th>
<th>$D_g$ [cm]</th>
<th>$\Sigma_{a,g}$ [cm$^{-1}$]</th>
<th>$\nu\Sigma_{f,g}$ [cm$^{-1}$]</th>
<th>$\Sigma_{21}$ [cm$^{-1}$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2</td>
<td>1</td>
<td>1.4</td>
<td>0.01</td>
<td>0.007</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.4</td>
<td>0.15</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.3</td>
<td>0.008</td>
<td>0.003</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.5</td>
<td>0.05</td>
<td>0.06</td>
<td></td>
</tr>
</tbody>
</table>

$v_1 = 10^7$ cm/s, $v_2 = 2 \times 10^5$ cm/s, $\nu = 2.43$, $\chi_1 = 1.0$

$\beta = 0.0075$, $\lambda = 0.08$ s$^{-1}$

have a cross-section area of 20x20 cm$^2$. Two energy groups and six delayed neutron families are modeled. All material properties are listed in Table 4.2.1 and for the delayed neutron precursor data it is referred to Table 4.2.2 (all values are quoted from [15]). Control rods (CR) that are represented as smeared absorber in the respective fuel assembly are combined in two CR banks denoted as Rod Group 1 and Rod Group 2. Initially, Rod Group 2 is partially inserted in the core while Rod Group 1 is located above the active core.

The initial reactor core power is 739.2 MW$_{th}$ [26] and the initial core power density in all published (reference) solutions is 150.0 W/cc. This implies an active core volume (that is defined as the sum of all fuel assembly volumes with uniform dimensions of 20x20x160 cm$^3$) of 4.928 m$^3$. From this it is deduced that the active fuel height that is used to compute volumetric power densities, whether per assembly or for the whole core, is 160 cm. A peculiarity of the core model, however, is the fact that the control rod material (as specified for the benchmark) contains

---

2The compositions denoted as “1” and “2” represent the same fuel type but differ in the absorption cross-sections $\Sigma_{a,1}$ and $\Sigma_{a,2}$ due to the presence of the control rods in “2”.

38
4.2 LMW Operational Transient

Figure 4.2.1: LMW reactor core: Horizontal and vertical cross-sections.
fissile material and hence, the top reflector contains fissile materials in the control rod locations. Since POLCA7 does not allow fissionable material in the axial reflector region and also requires all fuel bundles to be modeled with an uniform length, the presence of fissile material in the top reflector has the consequence that - in contrast to the reference solutions - the fuel assemblies are modeled with a length of 180 cm in POLCA7. This fact must be carefully considered when comparing assembly power densities as will be explained later in Chapter 6.

The transient is initiated by starting an upward withdrawal of Rod Group 2 at a constant speed of 3 cm/s. The CR withdrawal proceeds until the bank ends up just above the active core. At 7.5 seconds into the transient, Rod Group 1 starts to move down into the core with the same speed. The CR insertion lasts 40 seconds so that the bank is stopped 60 cm above the zero level (or 40 cm above the bottom of the active core). The final core condition is clarified in Figure 4.2.2. The entire transient is followed for 60 seconds.

### 4.3 LRA BWR Kinetics Problem

Originally, the LRA (LRA: Laboratorium für Reaktorregelung und Anlagensicherung) Kinetics Problem was developed as a two-dimensional, quadrant-symmetric core problem. The original problem description was published in the ANL Benchmark Problem Book [1]. Smith has extended it to three dimensions involving the ejection of four control rods and added a transient scenario, where only one of the original four transient control rods is ejected (see also [2, 26]). This problem is suitable to scrutinize the rod-drop transient (or prompt reactivity insertion accident) behavior in BWR applications.

The LRA benchmark problem employs two energy groups and two delayed neutron families. The initial reactor core configuration shown in Figures 4.3.1 and 4.3.2 consists of three\(^3\) different fuel types surrounded

---

\(^3\)The fuel regions marked as “3” and “4” differ only by the presence of control rods in “3” (resulting thus in a larger $\Sigma_{a,2}$).
Figure 4.2.2: LMW reactor core at the end of the transient.
Table 4.2.1: LMW operational transient: Material data.

<table>
<thead>
<tr>
<th>Material Group</th>
<th>Group</th>
<th>Init Power Q[cm]</th>
<th>( \delta f )</th>
<th>( \delta a )</th>
<th>( \delta c )</th>
<th>( \delta D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.9493631</td>
<td>0.000206</td>
<td>0.00647769</td>
<td>0.0175555</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.634272</td>
<td>0.00087662</td>
<td>0.01127328</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.9493631</td>
<td>0.000206</td>
<td>0.00647769</td>
<td>0.0175555</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.9493631</td>
<td>0.000206</td>
<td>0.00647769</td>
<td>0.0175555</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Initial reactor core power \( Q_C = 739.2 \text{ MW} \)

\( Q_1 = 1.25 \times 10^7 \text{ cm/s} \), \( v_1 = 1.0 \times 10^7 \text{ cm/s} \), \( \nu = 1.25 \times 10^7 \text{ cm/s} \), \( \chi = 1.0 \times 10^7 \text{ cm/s} \), \( s = 1.25 \times 10^7 \text{ cm/s} \), \( a = 1.0 \times 10^7 \text{ cm/s} \).
4.3 LRA BWR Kinetics Problem

Table 4.2.2: LMW operational transient: Delayed neutron precursor data.

<table>
<thead>
<tr>
<th>Family $l$</th>
<th>$\beta_l$</th>
<th>$\lambda_l [s^{-1}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000247</td>
<td>0.0127</td>
</tr>
<tr>
<td>2</td>
<td>0.0013845</td>
<td>0.0317</td>
</tr>
<tr>
<td>3</td>
<td>0.001222</td>
<td>0.115</td>
</tr>
<tr>
<td>4</td>
<td>0.0026455</td>
<td>0.311</td>
</tr>
<tr>
<td>5</td>
<td>0.000832</td>
<td>1.4</td>
</tr>
<tr>
<td>6</td>
<td>0.000169</td>
<td>3.87</td>
</tr>
</tbody>
</table>

both radially and axially by the water reflector. In total, 312 fuel assemblies with square dimensions of $15 \times 15 \text{cm}^2$ are modeled in the full core. Four symmetrically placed control rod cells (a single control rod cell covers four assembly locations) are identified as transient control rod locations and indicated as “CR”. The radial reflector, which has a thickness of at least twice the size of a fuel bundle, poses a problem for POLCA7 because it only can handle single layer of radial reflectors, which must be of the same thickness as a fuel assembly. Hence, the original problem had to be slightly modified by applying partial-current albedo boundary conditions on the core periphery in order to account for the omitted (excessive) reflector layers. The modified LRA core model is also illustrated in Figure 4.3.1. The albedo values found in Table 4.3.3 are taken from [24], where the correctness of these values was verified by comparing static results with reference solutions. Axially, no modification of the reflector is needed.

Material properties and delayed neutron precursor data are found in Tables 4.3.1 and 4.3.2, respectively. These data were taken from Brega et al. [6]. In this context, it should be noted that considerable discrepancies between values used in various published solutions and those specified in the original benchmark problem were pointed out by Sut-
4 Description of the Benchmark Problems

(a) Original problem.

(b) Modified reflector as used in POLCA7.

Figure 4.3.1: LRA four-rod reactor core model: Horizontal cross-section.
4.3 LRA BWR Kinetics Problem

ton et al. [28]. The values given by Brega agree with those that Sutton concluded were used in most of the reported analyses. The parameters in question are the Doppler feedback constant $\gamma$ and the precursor decay constant $\lambda_1$. The mentioned discrepancies in these parameters are significant in terms of their impact on the transient results (they do not impact the static results, however). Sutton noted discrepancies in particular with regard to the three-dimensional LRA version and did not discuss the data for the two-dimensional LRA problem. Therefore, some uncertainty concerning the “correct” parameters to be used in the two-dimensional case persists. Hence, in the present work it was decided to focus on three-dimensional calculations only. This is also motivated by the fact that three-dimensional calculations are considered to be a more realistic test of spatial kinetics methods. In Tables 4.3.1 and 4.3.2, “correct” values (i.e., those, which were used in POLCA-T and the referenced calculations) are given [6, 28].

The three-dimensional LRA problem is approached by analyzing two different transient cases:

1. The four-rod case, where the four symmetrically placed transient control rods are ejected simultaneously. In the literature, this case is often referred to as the quarter core case. In the present work, it is preferred to call it the four-rod case for the sake of clarity.

2. The single-rod case, in which only the transient rod in the top right quadrant of the core is ejected. In the literature, this case is known as the full core case.

The transient rod ejection involves transient control rod withdrawal in the downwards direction at a speed of 150 cm/s. A transient control rod is subsequently fully withdrawn after two seconds. The initial core state is at very low power, with a core power density of [1]:

$\gamma = 2.034 \times 10^{-3} \text{K}^{-0.5}$ and $\lambda_1 = 0.00654 \text{s}^{-1}$ whereas values used here (as taken from Brega) are $\gamma = 3.034 \times 10^{-3} \text{K}^{-0.5}$ and $\lambda_1 = 0.0654 \text{s}^{-1}$.

\footnote{The original values are $\gamma = 2.034 \times 10^{-3} \text{K}^{-0.5}$ and $\lambda_1 = 0.00654 \text{s}^{-1}$ whereas values used here (as taken from Brega) are $\gamma = 3.034 \times 10^{-3} \text{K}^{-0.5}$ and $\lambda_1 = 0.0654 \text{s}^{-1}$.}
Figure 4.3.2: LRA reactor core model: Vertical (modified) cross-section.
4.3 LRA BWR Kinetics Problem

\[ \bar{P} = \frac{\kappa}{V_C} \int_{V_C} \left( \sum_g \Sigma_{f,g} (\vec{r}, t) \phi_g (\vec{r}, t) \right) \, dV = 10^{-6} \text{ W cm}^{-3} \]

with \( V_C \) being the active core volume and \( \kappa \) being a constant given in Table 4.3.1. The active core volume is defined as the sum of all fuel assembly volumes \( V_i \) [1]. Considering uniform fuel assembly dimensions of 15x15x300 cm\(^3\) [9, 24], the active core volume becomes then 21.06 m\(^3\). This corresponds to an initial core power of 21.1 W.

Since the amount of reactivity induced by the rod ejection exceeds 1$ (i.e., the transient is super-prompt critical), some feedback is required to limit the power excursion. The only feedback that takes place on the same time scale as the neutron kinetics processes is the fuel temperature (Doppler) feedback. For this purpose, an adiabatic fuel heat-up and Doppler feedback model was implemented in POLCA-T according to the specifications in the LRA benchmark [1]:

\[ \frac{\partial T_f (\vec{r}, t)}{\partial t} = \alpha \left[ \sum_g \Sigma_{f,g} (\vec{r}, t) \phi_g (\vec{r}, t) \right] \]

\[ \Sigma_{a,1} (\vec{r}, t) = \Sigma_{a,1} (\vec{r}, 0) \left[ 1 + \gamma \left( \sqrt{T_f (\vec{r}, t)} - \sqrt{T_f (0)} \right) \right] \]

where \( T_f \) is the fuel temperature and \( \alpha \) and \( \gamma \) are constants given in Table 4.3.1.

The LRA benchmark problem is considered to be extremely difficult. Large radial flux gradients stress the spatial discretization method implemented in the code, while the extraordinary power excursion, which is more than ten orders of magnitude during some tenths of seconds only, and the subsequent shifts thereof stress the time integration process. More about the transient progression is presented in Chapter 6.
4 Description of the Benchmark Problems

Table 4.3.1: LRA benchmark problem: Material properties.

<table>
<thead>
<tr>
<th>Material Group</th>
<th>$g$</th>
<th>$D_g$ [cm]</th>
<th>$\Sigma_{\alpha,g}$ [cm$^{-1}$]</th>
<th>$\nu\Sigma_{\nu,g}$ [cm$^{-1}$]</th>
<th>$\Sigma_{21}$ [cm$^{-1}$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.255</td>
<td>0.008252</td>
<td>0.004602</td>
<td>0.02533</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.211</td>
<td>0.1003</td>
<td>0.1091</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1.268</td>
<td>0.007181</td>
<td>0.004609</td>
<td>0.02767</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.1902</td>
<td>0.07047</td>
<td>0.08675</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.259</td>
<td>0.008002</td>
<td>0.004663</td>
<td>0.02617</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.2091</td>
<td>0.08344</td>
<td>0.1021</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.259</td>
<td>0.008002</td>
<td>0.004663</td>
<td>0.02617</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.2091</td>
<td>0.073324</td>
<td>0.1021</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1.257</td>
<td>0.0006034</td>
<td>0.0</td>
<td>0.04754</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.1592</td>
<td>0.01911</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

$v_1 = 3.0 \times 10^7$ cm/s, $v_2 = 3.0 \times 10^5$ cm/s, $\nu = 2.43$, $\chi_1 = 1.0$

Fission rate to fuel temperature conversion factor:
$\alpha = 3.83 \times 10^{-11}$ Kcm$^3$
Doppler temperature feedback constant: $\gamma = 3.034 \times 10^{-3}$ K$^{-0.5}$
Energy conversion factor: $\kappa = 3.204 \times 10^{-11}$ Ws/fission
Initial fuel temperature: $T_f(0) = 300$ K
4.3 LRA BWR Kinetics Problem

Table 4.3.2: LRA benchmark problem: Delayed neutron precursor data.

<table>
<thead>
<tr>
<th>Family $l$</th>
<th>$\beta_l$</th>
<th>$\lambda_l ,[s^{-1}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0054</td>
<td>0.0654</td>
</tr>
<tr>
<td>2</td>
<td>0.001087</td>
<td>1.35</td>
</tr>
</tbody>
</table>

Table 4.3.3: Albedo values at the core periphery as applied in POLCA7.

<table>
<thead>
<tr>
<th>Albedo $A_{gg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast</td>
</tr>
<tr>
<td>Down-scattering</td>
</tr>
<tr>
<td>Thermal</td>
</tr>
</tbody>
</table>
4 Description of the Benchmark Problems

4.4 Benchmarking Procedure

After the presentation of the benchmark problems involved in this project but prior to the discussion of the benchmarking results, it is reasonable to describe the benchmarking procedure. To put it briefly, the procedure involves core modeling in the static code POLCA7 for the computation of the initial steady-state solution. Afterwards, the same core model (i.e., POLCA7 input data) is re-used by POLCA-T with the addition of some boundary conditions. The initializing steady-state solution is then verified by running both a static case and a zero-transient (null-transient) case. The static case is just a POLCA7 case initiated from within POLCA-T. The zero-transient case is actually an unperturbed transient case that is run for some period (in this work for 1 second) prior to initiating the actual transient perturbation (which then is initiated at 1 second). In POLCA-T, the three stages take place in a single execution (run) of the code.

4.4.1 POLCA7 Core Modeling

In general, the modeling of a reactor core implies a (correct) set-up of both the core geometry and the properties of all materials involved. In addition, some physical parameters, such as the initial power, and various computing (or numerical) options have to be specified. The geometry includes, for example, the number of fuel rods and assemblies, their dimensions, control rod positions, etc. All data required for a specific solution are prescribed for the benchmark problem in question. However, the user is free to choose an adequate spatial mesh. In fact, several references employ different mesh sizes in order to study the sensitivity of the solution to the dimensions of homogenized nodes. Deviations of coarse-mesh solutions from fine-mesh applications serve also as clear indicators of spatial truncation errors associated with the respective discretization procedures.

As concerns POLCA7, the mesh size on the radial plane is fixed and corresponds to the dimensions of the modeled fuel assembly. Furthermore, only one reflector layer (with a size corresponding to fuel assem-
4.4 Benchmarking Procedure

bly dimensions) can be modeled in POLCA7 radially. Axially, however, (and this concerns only the three-dimensional problems) one is able to employ variable node sizes. This feature attracts one’s attention in transients with control rod movements or in modeling large axial reflector regions. Albedo boundary conditions are applied on the periphery of the problem domain (such as zero flux, perfect reflection, etc.). The two-dimensional problems are modeled by specifying a single axial layer of 10 to 15 cm height with perfectly reflecting axial boundary conditions.

In standard calculations, the material properties are passed from the lattice code to POLCA7 by using cell data tables. For the purpose of benchmarking that does not require any lattice code calculations the necessary data are manually inserted in the cell data tables. This includes the fuel and reflector properties (cross-sections and diffusion coefficients), delayed neutron data as well as cross-section corrections induced by control rod insertions. The control rods are modeled as smeared absorbers in associated fuel assemblies (a single assembly per control rod in the case of a PWR and four assemblies per control rod in a BWR).

With respect to adjustable numerical options, these mainly include a set-up of iteration and acceleration parameters. The iteration parameters define, inter alia, the convergence criteria for the power density and the eigenvalue iterations. The acceleration options set the parameters for the inner (eigenvalue problem) and outer (power/void) iteration loops. In the present work, default values are used in most cases. Additionally, the user may specify calculation options that are to be employed. For example, it is possible to exclude any thermal-hydraulic calculations or to choose if axial homogenization of nodes is to be activated or not (see Chapter 6 for more details on this feature).

After the core configuration, the modeling is tested under steady-state conditions against reference solutions (i.e., static solutions of reference codes). In particular, the eigenvalues $k_{\text{eff}}$ and the assembly power densities of the initial core (prior to the relevant transient) and the final core configuration (at the end of the relevant transient) are addressed.
In this context, it should be pointed out that only the conformance of both parameters verify the correctness of the core model. For instance, it was observed that a confirmed eigenvalue did not automatically imply a proper core modeling. Rather, fortuitous error cancellation may occur. The relative difference between the POLCA7 and the reference solution for the power density in the $i$-th assembly is defined as

$$\epsilon_{rel,i} [\%] = \frac{P_{i,sol}^n - P_{i,ref}^n}{P_{i,ref}^n} \cdot 100$$

with $P_{i,sol}^n$ and $P_{i,ref}^n$ being the normalized power density predicted by POLCA7 and the reference solution, respectively. The normalization in POLCA7 is done such that the mean reactor power density is 1000. Consequently, all presented reference solutions are scaled up by a factor of 1000 in order to facilitate the comparison. The normalization is computed in the following way:

1. The power density in the $i$-th fuel assembly is computed as:

$$P_i = \frac{Q_i}{V_i}$$

where $Q_i$ is the total power generated in all nodes between axial reflectors in the assembly $i$ and $V_i$ is the volume of the assembly $i$ that is defined as the sum of all nodes between the upper and the lower reflector layers. That is, even non-power generating nodes are included if they exist in the active core region between axial reflectors.

2. According to the definition above, the sum of the normalized assembly power densities is equal to the number of fuel assemblies in the core $N$ times 1000:

$$\sum_i P_i^n = 1000 \cdot N$$

3. At the same time, it is also clear that

$$P_i^n = c \cdot P_i$$
where \( c \) is the normalization factor computed as

\[
c = \frac{1000 \cdot N}{\sum_i Q_i / V_i}
\]

4. Finally, one thus can formulate:

\[
\bar{P}_{\text{in}}^u = \frac{Q_i}{V_i} \cdot \frac{N}{\sum_i Q_i / V_i} \cdot 1000
\]

Note again that even non-power generating nodes (that lie between the axial reflectors) are included in \( V_i \). For the comparison of power densities computed by POLCA7 with those of reference solutions, special precautions are required in the LMW case (see also Section 4.2 and Chapter 6 for further details).

### 4.4.2 POLCA-T Modeling

The verified core model is now re-used by POLCA-T. Here, it is necessary to model some basic core surroundings, which include the upper and lower plenum, the core by-pass, the core barrel dimensions and materials, etc. Note that the arrangement does not affect the calculations since it is a part of the deactivated thermal-hydraulic calculations (thermal-hydraulics is not modeled in any of the benchmarks considered in this study). However, the code requires at least a simple configuration of core surroundings in order to be able to carry out the simulations.

Furthermore, numerical options are adjusted in order to meet the specific challenges arising from the respective kinetics problem. The options include appropriate settings for the time-step size, the convergence criterion \( \varepsilon_{\text{feed}} \) as well as the iteration parameters. As concerns the time-step size, the user can specify the size of the main feedback time-step, while either keeping the kinetics time-step constant and equal to the feedback time-step or letting the code automatically adapt the kinetics time-step based on the temporal behavior of the neutronics solution. Furthermore, one is able to vary the feedback time-step size during
4 Description of the Benchmark Problems

a transient, which is useful when rapid changes in power occur, for example.

In POLCA-T, disturbances are specified in a table where the perturbation variable is listed as a function of time. In three-dimensional cases, the control rod speed or position are used as the perturbation input, while in two-dimensional cases, the disturbances are specified by variations in absorption cross-sections. Prior to the perturbation onset, the user may specify a certain time frame, which then serves as the zero-transient period. By running the zero-transient both the actual input but also the correctness of the initial steady-state solution is verified. A complete or final verification of the neutron kinetics model is achieved after a successful running of the “real” transient and comparing to known results from other codes.

In this regard, the relative difference resulting from comparisons of POLCA-T results with those of other codes at any given time point is computed as:

$$\epsilon_{rel} [%] = \left| \frac{Q_{sol}^{rel} - Q_{ref}^{rel}}{Q_{ref}^{rel}} \right| \cdot 100 \quad (4.4.2)$$

with $Q_{sol}^{rel}$ and $Q_{ref}^{rel}$ being the relative core power (relative to nominal steady-state core power) predicted by POLCA-T and the reference solution, respectively.
5 Numerical Study

It is clear that the benchmarking procedure requires a profound understanding of the impacts of various numerical options on the outcome of the simulations. In this vein, the numerical options can be further optimized in terms of the solution accuracy and computational efforts. However, it is also important to address the numerical stability of the obtained solution in dependence of variations in these numerical options. Thus, the objective of the numerical study described in this chapter is twofold: to examine the impacts of different numerical options on the solution accuracy, stability and computing costs and to provide some indication for preferable numerical settings to be used in the subsequent benchmarking calculations.

As Smith and Gehin already pointed out [9, 26], the relative simplicity of the TWIGL problem makes it suitable to be employed as a model for such a deepened analysis of numerical features of the kinetics model in question. Smith used the ramp perturbation transient as the basis for his numerical study, described in [26]. For the sake of a better comparability, it is reasonable to employ the same transient problem even in the present work. Note that the study relates only to parameters associated with the neutron kinetics model itself, and not to any parameters associated with the initializing steady-state solution.

5.1 Studied Parameters

In this study, following parameters are manipulated and their consequent impacts analyzed:

**Convergence criterion:** The user-defined convergence criterion $\epsilon_{\text{feed}}$ controls also the convergence of all iteration levels (see Section 3.3).
5 Numerical Study

**Non-linear iterations (NONLIN):** Hereby, the number of permitted coupling coefficients updates is set.

**Flux inner iterations (MAXFLX, MINFLX):** The user defines both the maximum and the minimum number of flux inner iterations per coupling coefficients update.

**Time-step size:** The temporal mesh for feedback between neutronics and thermal-hydraulics is adjusted. Hereby, $\Delta T_{\text{min}}$ and $\Delta T_{\text{max}}$ denote, respectively, the minimum and the maximum feedback time-step size.

**Adaptive kinetics time stepping:** The user has the choice to employ constant kinetics time-steps or let the code define these. The latter option is the default setting.

**Delta flux:** Hereby, the user may define the maximum permissible relative change in neutron flux over a kinetics time-step. As an example: if $\Delta^\text{kin}_\Phi = 0.1$, the allowed change in flux over a kinetics time-step is 10% at most. This parameter affects the adaptive time-step selection procedure (if it is active).

**Theta value:** Hereby, the theta value for the time integration procedure is set.

Default values of the mentioned options and convergence criterion are listed in Table 5.1.1.

5.2 Results

As a first step in the analysis, the correctness of the core model is verified through static eigenvalue calculations (with POLCA7). The static results, however, are not subject to the discussion in this chapter and hence, it is referred to Chapter 6, where these results are described in detail. Instead, the starting point for the sensitivity study is a transient run with the same spatial and temporal mesh and nearly the same numerical options as they were used in the reference solution. As the reference solution for all subsequent calculations discussed in this section, QUANDRY results as quoted in [26] are used. The
5.2 Results

**Table 5.1.1:** Default values of the described numerical parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon_{feed}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>NONLIN</td>
<td>25</td>
</tr>
<tr>
<td>MAXFLX</td>
<td>25</td>
</tr>
<tr>
<td>MINFLX</td>
<td>5</td>
</tr>
<tr>
<td>$\Delta_{\Phi}^{kin}$</td>
<td>0.1</td>
</tr>
<tr>
<td>Theta $\Theta$</td>
<td>1</td>
</tr>
</tbody>
</table>

choice of QUANDRY is motivated by the fact that the code is similar to POLCA-T with respect to the spatial discretization and time integration methods.

The both solutions employ initially a constant kinetics time-step size of 5 ms and a spatial mesh with the dimensions of $8 \times 8 \, \text{cm}^2$. As concerns the numerical options, values listed in Table 5.1.1 are employed.\(^1\) This initializing POLCA-T solution is summarized in Table 5.2.1 where it is compared with QUANDRY results.

The maximum difference in relative power between POLCA-T and QUANDRY is 0.1%, which is considered very good since this lies well within the typical spread of results obtained for this benchmark with a variety of codes (see Chapter 6).

With this solution as the starting point, it is now of particular interest to examine if an improvement of the simulations in terms of increased computational efficiency (reduced computing time) can be achieved. Therefore, as the next step, the effects of different temporal meshes are analyzed. For this purpose, the constant kinetics time-step size is subsequently increased to 10, 25 and 50 ms. Additionally, a simulation

---

\(^1\)Since the numerical procedures used in POLCA-T and QUANDRY are slightly different, some numerical options are not directly comparable in between. Thus, default values of these options are used in the initializing POLCA-T simulation.
5 Numerical Study

Table 5.2.1: Relative power versus time for the initializing TWIGL ramp perturbation calculation (constant 5 ms case).

<table>
<thead>
<tr>
<th>Code</th>
<th>QUANDRY</th>
<th>POLCA-T</th>
<th>$\epsilon_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Relative power</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0 s</td>
<td>1.000</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td>0.1 s</td>
<td>1.308</td>
<td>1.308</td>
<td>0.0</td>
</tr>
<tr>
<td>0.2 s</td>
<td>1.959</td>
<td>1.961</td>
<td>0.1</td>
</tr>
<tr>
<td>0.3 s</td>
<td>2.077</td>
<td>2.076</td>
<td>0.0</td>
</tr>
<tr>
<td>0.4 s</td>
<td>2.090</td>
<td>2.093</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5 s</td>
<td>2.110</td>
<td>2.111</td>
<td>0.0</td>
</tr>
<tr>
<td>CPU time</td>
<td>-</td>
<td>6.68 s</td>
<td>-</td>
</tr>
</tbody>
</table>

with a time-step size of 1 ms is studied as well. The computed results are shown in Table 5.2.2.

It is noticeable that POLCA-T shows a strong stability in terms of the solution accuracy for a broad range of kinetics time-step sizes. If one was to consider changes in core relative power of less than 1% as insignificant, or at least as acceptable, then it seems that up to a time-step size of 25 ms the POLCA-T results change little and that a significant change first occurs at around a 50 ms time-step size. On the other hand, the adjustment of $\Delta t$ to 25 ms considerably reduces the CPU time by a factor of 2.3. Still, the difference to QUANDRY even for a time-step size of 50 ms is not larger than that between other codes (in comparison to QUANDRY) as it will be shown in the next chapter. Furthermore, it is also evident that the simulation with a small time-step size of 1 ms provides expected results (flux is already well converged in the time variable in the 5 ms case) but the increase in computational time is fairly significant.

As next, the effects of the convergence criteria are analyzed. Recall
### 5.2 Results

#### Table 5.2.2: Impacts of various kinetics time-step sizes on the solution accuracy.

<table>
<thead>
<tr>
<th>Δt</th>
<th>50ms</th>
<th>25ms</th>
<th>10ms</th>
<th>5ms</th>
<th>1ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Power ϵ&lt;sub&gt;rel&lt;/sub&gt;</td>
<td>Power ϵ&lt;sub&gt;rel&lt;/sub&gt;</td>
<td>Power ϵ&lt;sub&gt;rel&lt;/sub&gt;</td>
<td>Power ϵ&lt;sub&gt;rel&lt;/sub&gt;</td>
<td></td>
</tr>
<tr>
<td>0.0s</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td>0.1s</td>
<td>1.314</td>
<td>0.5</td>
<td>1.311</td>
<td>0.2</td>
<td>1.309</td>
</tr>
<tr>
<td>0.2s</td>
<td>1.983</td>
<td>1.2</td>
<td>1.972</td>
<td>0.7</td>
<td>1.965</td>
</tr>
<tr>
<td>0.3s</td>
<td>2.078</td>
<td>0.0</td>
<td>2.077</td>
<td>0.0</td>
<td>2.076</td>
</tr>
<tr>
<td>0.4s</td>
<td>2.098</td>
<td>0.4</td>
<td>2.094</td>
<td>0.2</td>
<td>2.093</td>
</tr>
<tr>
<td>0.5s</td>
<td>2.115</td>
<td>0.2</td>
<td>2.112</td>
<td>0.1</td>
<td>2.111</td>
</tr>
<tr>
<td>CPU time</td>
<td>1.90s</td>
<td>2.89s</td>
<td>4.46s</td>
<td>6.68s</td>
<td>25.40s</td>
</tr>
</tbody>
</table>


5 Numerical Study

Table 5.2.3: Impact of the convergence criterion on the solution accuracy.

<table>
<thead>
<tr>
<th>Time</th>
<th>$\epsilon_{\text{feed}}$</th>
<th>$10^{-2}$</th>
<th>$10^{-3}$</th>
<th>$10^{-4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Power</td>
<td>$\epsilon_{\text{rel}}$</td>
<td>Power</td>
<td>$\epsilon_{\text{rel}}$</td>
</tr>
<tr>
<td>0.0 s</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td>0.1 s</td>
<td>1.308</td>
<td>0.0</td>
<td>1.308</td>
<td>0.0</td>
</tr>
<tr>
<td>0.2 s</td>
<td>1.961</td>
<td>0.1</td>
<td>1.961</td>
<td>0.1</td>
</tr>
<tr>
<td>0.3 s</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
</tr>
<tr>
<td>0.4 s</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5 s</td>
<td>2.111</td>
<td>0.0</td>
<td>2.111</td>
<td>0.0</td>
</tr>
<tr>
<td>CPU time</td>
<td>6.87 s</td>
<td>6.68 s</td>
<td>6.92 s</td>
<td></td>
</tr>
</tbody>
</table>

from Section 3.3 that the criteria are dependent on the user-specified convergence criterion for the outer feedback $\epsilon_{\text{feed}}$. The constant kinetics time-step size is set again to 5 ms. A somewhat surprising fact is that a loose convergence criterion does not affect the results at all (see Table 5.2.3). On the other hand, there is no reduction in execution time if a looser criterion ($10^{-2}$ in this case) is applied. A tightened criterion ($10^{-4}$ in this case) does not affect the solution (again, flux is already well converged in the $10^{-3}$ case), though, warning messages printed by the code indicate (not unexpectedly) that the number of default non-linear iterations is too low for enabling flux to converge during the first two tenth parts of seconds (i.e., during the reduction of absorption cross-section). Nonetheless, POLCA-T shows repeatedly a stable behavior with respect to the reproducibility of the results in all three simulations. Consequently, the convergence criterion is kept at $\epsilon_{\text{feed}} = 10^{-3}$ during all subsequent runs in this study.

The computation time can also be minimized by reducing the permitted (or required) number of iterations as long as it does not affect the
5.2 Results

Table 5.2.4: Impact of a varying number of allowed non-linear iterations on the solution accuracy.

<table>
<thead>
<tr>
<th></th>
<th>NONLIN</th>
<th>MINFLX</th>
<th>MAXFLX</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>10</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>0.0s</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td>1.308</td>
<td>0.0</td>
<td>1.308</td>
<td>0.0</td>
</tr>
<tr>
<td>1.961</td>
<td>0.1</td>
<td>1.961</td>
<td>0.1</td>
</tr>
<tr>
<td>2.076</td>
<td>0.0</td>
<td>2.078</td>
<td>0.0</td>
</tr>
<tr>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
</tr>
<tr>
<td>2.111</td>
<td>0.0</td>
<td>2.111</td>
<td>0.0</td>
</tr>
<tr>
<td>CPU time</td>
<td>6.68s</td>
<td>5.74s</td>
<td>6.06s</td>
</tr>
</tbody>
</table>

solution accuracy. This is especially true for the relatively expensive coupling coefficients updates (non-linear iterations). Hence, the impact of each of the iteration levels is examined in the next step. During the simulations, only one of the three iteration parameters NONLIN, MINFLX and MAXFLX is changed while the two other are kept constant. Results for different number of permitted non-linear iterations are demonstrated in Table 5.2.4.

Clearly, a decreasing number of allowed non-linear iterations does not affect the results remarkably. However, in the last two simulations (for NONLIN=5 and NONLIN=2), flux does not converge at some few time-steps and the relative error in flux at given non-linear iteration is up to 100 times higher the inner iterations criterion $\varepsilon_{\text{inners}}$ that is set to $10^{-5}$. It is thus assumed that transients with larger flux gradients in time might yield solutions with larger errors. Moreover, the decrease
in computational efforts with reduced NONLIN is not as significant as one might have expected. This can be explained by the fact that the TWIGL ramp transient does not require more than five or so non-linear iterations to achieve convergence and hence, the number of “effective” non-linear iterations is all about the same in all three simulations.

Results for different number of permitted flux inner iterations are presented in Table 5.2.5. It is obvious that the solution accuracy is hardly affected by variations in MINFLX and MAXFLX. Neither are notable variations in CPU time observed. It is thus assumed that in this particular transient problem, a proper flux convergence requires only some few inner iterations. Based on the outcome presented in Tables 5.2.4 and 5.2.5, it is now reasonable to simultaneously reduce the number of all three parameters.

The impacts of a reduced number of NONLIN, MINFLX and MAXFLX on the solution accuracy are given in Table 5.2.6. Repeated warnings regarding the deficient flux convergence occur in the last two runs (however, the relative error in flux is lower than in the case with NONLIN=2 as mentioned above). This difficulty to converge properly requires therefore more inner iterations per non-linear iteration. This does also negatively affect the computing time.

Summarizing, it can be noted that a reduction in the number of iterations does not impair the solution accuracy for transients with moderate flux gradients in time. Transients with large flux gradients in time, however, might require an increase in the number of permitted and required iterations in order to obtain sufficiently accurate results. At the same time, there is no notable gain in computing efficiency in cases with reduced NONLIN, MINFLX and MAXFLX. Consequently, it is reasonable to keep the three iteration parameters at their respective default values listed in Table 5.1.1.

The automated time-stepping algorithm that subdivides one feedback time-step into an appropriate number of kinetics time-steps (if the adaptive kinetics time-stepping procedure is activated) is evaluated next. It is started with the constant feedback time-step size of 50 ms in order to investigate if the solution presented in Table 5.2.2 can be
Table 5.2.5: Impact of a varying number of flux inner iterations on the solution accuracy.

<table>
<thead>
<tr>
<th></th>
<th>NONLIN</th>
<th>MINFLX</th>
<th>MAXFLX</th>
<th>Time</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25</td>
<td>5</td>
<td>25</td>
<td>0.0s</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>2</td>
<td>10</td>
<td>0.1s</td>
<td>1.308</td>
<td>0.0</td>
<td>1.308</td>
<td>0.0</td>
<td>1.308</td>
<td>0.0</td>
<td>1.308</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>10</td>
<td>15</td>
<td>0.2s</td>
<td>1.961</td>
<td>0.1</td>
<td>1.961</td>
<td>0.1</td>
<td>1.961</td>
<td>0.1</td>
<td>1.961</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>2</td>
<td>5</td>
<td>0.3s</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>10</td>
<td>5</td>
<td>0.4s</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>15</td>
<td>5</td>
<td>0.5s</td>
<td>2.111</td>
<td>0.0</td>
<td>2.111</td>
<td>0.0</td>
<td>2.111</td>
<td>0.0</td>
<td>2.111</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CPU time</td>
<td>6.68s</td>
<td>7.49s</td>
<td>7.51s</td>
<td>7.30s</td>
<td>7.28s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5 Numerical Study

Table 5.2.6: Impact of a reduced number of iterations on the solution accuracy.

<table>
<thead>
<tr>
<th></th>
<th>NONLIN</th>
<th>MINFLX</th>
<th>MAXFLX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Power</td>
<td>$\epsilon_{rel}$</td>
<td>Power</td>
</tr>
<tr>
<td>Time</td>
<td>0.0 s</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.1 s</td>
<td>1.308</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.2 s</td>
<td>1.961</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.3 s</td>
<td>2.078</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.4 s</td>
<td>2.093</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>0.5 s</td>
<td>2.111</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>CPU time</td>
<td>5.72 s</td>
<td>6.46 s</td>
</tr>
</tbody>
</table>

64
Table 5.2.7: Impacts of variable kinetics time-steps on the solution accuracy.

<table>
<thead>
<tr>
<th>$\Delta T$</th>
<th>$\Delta t$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50 ms</td>
<td></td>
<td></td>
<td>5 ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONST.</td>
<td>ADAPT.</td>
<td>CONST.</td>
<td>ADAPT.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td>Power</td>
<td>$\epsilon_{rel}$</td>
<td>Power</td>
<td>$\epsilon_{rel}$</td>
<td>Power</td>
<td>$\epsilon_{rel}$</td>
<td>Power</td>
<td>$\epsilon_{rel}$</td>
</tr>
<tr>
<td>0.0 s</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td>0.1 s</td>
<td>1.314</td>
<td>0.5</td>
<td>1.310</td>
<td>0.2</td>
<td>1.308</td>
<td>0.0</td>
<td>1.308</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.2 s</td>
<td>1.983</td>
<td>1.2</td>
<td>1.964</td>
<td>0.3</td>
<td>1.961</td>
<td>0.1</td>
<td>1.961</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.3 s</td>
<td>2.078</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.4 s</td>
<td>2.098</td>
<td>0.4</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>0.5 s</td>
<td>2.115</td>
<td>0.2</td>
<td>2.112</td>
<td>0.1</td>
<td>2.111</td>
<td>0.0</td>
<td>2.111</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>CPU time</td>
<td>1.90 s</td>
<td></td>
<td>2.97 s</td>
<td>6.68 s</td>
<td></td>
<td>6.29 s</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Improved by means of this algorithm. The results are presented in Table 5.2.7 (where $\Delta T = \Delta T_{\text{min}} = \Delta T_{\text{max}}$). Obviously, the adaptive time-stepping option improves the results significantly for $\Delta T = 50 \text{ ms}$. The obtained differences in comparison to the reference solution are reduced by up to a factor 4 (from 1.2% to 0.3% at 0.2 s). Though, this is achieved at the costs of a higher execution time. For $\Delta T = 5 \text{ ms}$, no deviations from the solution with $\Delta t = 5 \text{ ms}$ are observed with the adaptive time-stepping. This is probably due to a 5 ms kinetics time-step being quite sufficient to capture the transient behavior so that the adaptive time-step algorithm does not actually induce finer kinetics time-step sizes. At the same time, the simulations with $\Delta T = 5 \text{ ms}$ require about double the effort of the adaptive time-step case with $\Delta T = 50 \text{ ms}$. One can thus conclude that the automated time-stepping algorithm is most efficient for large $\Delta T$ or in the case of rapid temporal changes in flux.

Effects of user-specified, maximum permitted relative flux change dur-
ing a kinetics time-step $\Delta_{\Phi}^{\text{kin}}$ cannot be evaluated as straightforward as the other parameter. The reason for this is that this parameter is used in conjunction with the temporal truncation tolerance to determine the kinetics time-step size. The truncation tolerance is steered via the user input parameter $\varepsilon_{\text{feed}}$. The time-step size is chosen according to whichever of these two tolerances restrict the kinetics time-step the most. Despite this complication, one can evaluate the impact of $\Delta_{\Phi}^{\text{kin}}$ to some extent by varying its value, and this was done for a constant truncation tolerance corresponding to $\varepsilon_{\text{trunc}} = 10^{-5}$.

Table 5.2.8 displays different solutions obtained with 50% (it is coded as the upper limit of $\Delta_{\Phi}^{\text{kin}}$), 10% and 1% maximum permitted relative changes in flux, respectively. Two scenarios are considered: one in which the feedback time-step size is 5 ms and the other of which it is 50 ms. One expects the impact of $\Delta_{\Phi}^{\text{kin}}$ to be seen mostly in the 50 ms case since the previous results already indicated that a kinetics time-steps size of 5 ms is probably close to a minimum needed for high accuracy. Clearly, the both simulations with $\Delta_{\Phi}^{\text{kin}} = 50\%$ do not affect the results at all. It is assumed that in these cases, the kinetics time-step size is regulated by the temporal truncation tolerance and is therefore in the range of the reference solution. The computation with $\Delta_{\Phi}^{\text{kin}} = 1\%$ and $\Delta T = 5\,$ ms produces also expected results, however, with a considerably higher computation time than for $\Delta_{\Phi}^{\text{kin}} = 10\%$ as the total number of kinetics time-steps now increases by a factor of about 1.3.

The real effect of the $\Delta_{\Phi}^{\text{kin}}$-algorithm is revealed in the last simulation, where a constant feedback time-step size of 50 ms is used in combination with the automated time-stepping algorithm and $\Delta_{\Phi}^{\text{kin}}$ set to 1%. Here, the very rapid change in flux during the important part of the power escalation results in $\Delta_{\Phi}^{\text{kin}}$ determining the kinetics time-step size. The end result is that a tighter $\Delta_{\Phi}^{\text{kin}}$ produces a denser kinetics time-stepping during the important phase of the transient which resembles that of the 5 ms case. During subsequent slower periods of the transient, larger time-steps are allowed. This results in a reduction of the CPU time relative to that of the 5 ms case while retaining the same
5.2 Results

Table 5.2.8: Impacts of maximum permitted relative flux changes over a kinetics time-step.

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Power ε_{rel}</th>
<th>Power ε_{rel}</th>
<th>Power ε_{rel}</th>
<th>Power ε_{rel}</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 s</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>7.09 s</td>
</tr>
<tr>
<td>0.1 s</td>
<td>1.308</td>
<td>0.1</td>
<td>1.309</td>
<td>0.1</td>
<td>6.68 s</td>
</tr>
<tr>
<td>0.2 s</td>
<td>1.961</td>
<td>0.1</td>
<td>1.960</td>
<td>0.1</td>
<td>9.30 s</td>
</tr>
<tr>
<td>0.3 s</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td>3.27 s</td>
</tr>
<tr>
<td>0.4 s</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
<td>2.97 s</td>
</tr>
</tbody>
</table>

**Table 5.2.8 continued...**
5 Numerical Study

accuracy as that of the 5 ms case. It can be concluded that $\Delta_{kin}^{\Phi}$ and $\varepsilon_{trunc}$ are good complements in regulating the kinetics time-step size. Since the result with $\Delta_{kin}^{\Phi} = 10\%$ is found to be acceptable, this is the value that will be used in the rest of this work.

The examination of different theta values is carried out by studying effects of the explicit Euler ($\Theta = 0$), Crank-Nicolson ($\Theta = 0.5$) and fully implicit ($\Theta = 1.0$) integration methods. The calculations reveal that difficulties arise only with explicit Euler method where negative fluxes are computed (independent on the time-step size employed) and the computation is stopped. The Crank-Nicolson method, however, shows reliable results as long as the temporal mesh employed is sufficiently small. Then, the solution accuracy is as high as in the reference case. For larger time-step sizes the numerical round-off errors become more pronounced, especially in single-precision calculations [23] and the method becomes susceptible to stability problems. Nonetheless, the computation with $\Theta = 0.5$ and a constant kinetics time-step size of 5 ms is not less accurate than the CUBBOX solution, which is presented in the next chapter. Hence, the method is positively capable of computing transients if so required for some reason, even though it necessitates a higher computational effort. The results with $\Theta = 0.5$ for two different time-step sizes are shown in Table 5.2.9.

In contrast to the conditionally stable Crank-Nicolson method, the fully implicit method is unconditionally stable for any time-step size [28] and is therefore a certainly preferred method.

5.3 Conclusions

The TWIGL sensitivity study demonstrates clearly that the POLCA-T neutronics model possesses a high numerical stability for a wide range of varying numerical settings. The kinetics time-step size can be increased from 5 ms to 25 ms without observing any numerical instabilities or significant impacts (under 1% in core power level) on the solution accuracy. The default values for the maximum allowed (or minimum required) number of iterations can be decreased without affecting the
5.3 Conclusions

Table 5.2.9: Impacts of different theta values on the solution accuracy.

<table>
<thead>
<tr>
<th>Θ</th>
<th>ΔT</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
<th>Power</th>
<th>$\epsilon_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>5 ms</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td>0.5</td>
<td>0.25 ms</td>
<td>1.319</td>
<td>0.8</td>
<td>1.309</td>
<td>0.1</td>
<td>1.308</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>5 ms</td>
<td>1.984</td>
<td>1.3</td>
<td>1.961</td>
<td>0.1</td>
<td>1.961</td>
<td>0.1</td>
</tr>
<tr>
<td>0.3</td>
<td>5 ms</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
<td>2.076</td>
<td>0.0</td>
</tr>
<tr>
<td>0.4</td>
<td>5 ms</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
<td>2.093</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5</td>
<td>5 ms</td>
<td>2.111</td>
<td>0.0</td>
<td>2.110</td>
<td>0.0</td>
<td>2.111</td>
<td>0.0</td>
</tr>
</tbody>
</table>

CPU time | 8.69 s | 120.00 s | 6.68 s |

results but still yielding a converged flux. Neither does a looser convergence criterion $\epsilon_{feed}$ impair the solution accuracy. By this means, the high reproducibility of accurate results can definitely be exploited to decrease the computational efforts but it also indicates that the solution is rather insensitive to significant variations in input.

The computational efficiency can be additionally improved by employing the automated time-stepping algorithm, especially in combination with large feedback time-step sizes. In transients with large flux changes in time, the combination of a tightened $\Delta^\text{kin}$-criterion and a coarse temporal mesh is another effective tool that is capable to decrease both the solution error and the computing costs.

Based on the results of this numerical study, some indication for appropriate values of the mentioned numerical parameters that are to be used in the subsequent benchmark calculations can be derived. Though, it should be pointed out that these values, are alterable depending on the problem considered but also on options employed in reference solutions.

- Time-stepping: The feedback and kinetics time-step sizes are set
5 Numerical Study

depending on the problem simulated. The size is also adjusted such that the comparability with reference solutions is facilitated.

- **Number of iterations**: It is shown that even though some CPU time can be saved by reducing the allowed number of non-linear iterations, it is reasonable to keep NONLIN, MINFLX and MAXFLX at their default values that are 25, 5 and 25, respectively. These values are used in the subsequent TWIGL and LMW benchmark simulations. The more difficult LRA transient with its extremely large flux gradients requires, however, a higher number of non-linear and flux inner iterations.

- **The convergence criterion**: A looser $\varepsilon_{\text{feed}}$ does not impair the initially obtained results in this particular problem. Though, it is assumed that in transients with larger flux gradients in time the results can be affected. Consequently, $\varepsilon_{\text{feed}}$ is set 0.001 unless it is shown that the solution accuracy is not affected for a looser $\varepsilon_{\text{feed}}$.

- **Delta flux**: It is reasonable to apply $\Delta_{\Phi}^{\text{kin}} = 0.1$ (see the results presented in the previous section).

- **Theta value**: With regard to the implicitness of the integration method, the most stable, precise and less extensive solutions are obtained with the fully implicit method.
6 Benchmarking Results and Discussion

In this chapter, solutions of different kinetics problems obtained with POLCA-T are presented. The results are compared with other, well-established spatial kinetics codes. The discussion starts with the analysis of the two-dimensional TWIGL reactor kinetics problem and continues subsequently with the three-dimensional LWR operational transient LMW and the BWR kinetics benchmark problem LRA.

6.1 TWIGL Reactor Kinetics Problem

In the previous chapters, the TWIGL reactor kinetics problem was introduced and the ramp perturbation transient was used as a model for the sensitivity study of POLCA-T neutronics. In this section, the initial static eigenvalue results are detailed and both the step and ramp perturbation transients extensively analyzed. Recall from Chapter 4 that while in the ramp perturbation transient the reactivity is inserted by linearly decreasing the absorption cross-sections during a certain period of time, the same amount of reactivity is inserted suddenly in the step case.

Static eigenvalue results computed by POLCA7 for initial conditions are shown in Table 6.1.1 where the solution is compared with other codes. For the POLCA7 simulations, meshes with dimensions of 8x8 cm$^2$ are employed. This mesh structure yields thus 400 equal-sized nodes in total for the full core model. Accordingly, codes with 400 nodes employ the same mesh structure as POLCA7.\footnote{In literature, one can find several problem solutions with coarser meshes. However,}
are quoted from [26, 28, 29].

The computed initial $k_{eff}$ matches exactly most of the published results including the fine-mesh eigenvalue solution of SPANDEX. Thus, the initial static solution can be considered to be completely acceptable. Subsequently, a one second zero-transient is executed in POLCA-T prior to starting the actual transient. Throughout this one second period the zero-transient yields exactly the same result as the preceding steady-state calculation. This verifies that the transient equations are correctly implemented.

### 6.1.1 Ramp Perturbation Transient

The results for the ramp perturbation transient are compiled in Table 6.1.2. All codes employ a constant kinetics time-step size of 5 ms. Furthermore, POLCA-T employs optimized numerical settings as those determined in Chapter 5, and which are basically values listed in Table 5.1.1. As concerns the spatial mesh, the same node size (8x8 cm$^2$) is used in QUANDRY, AETNA [29] and POLCA-T. CUBBOX [15] uses a somewhat coarser mesh with nodes of $h_x = h_y = 16$ cm for $24 \leq x, y \leq 136$ cm and $h_x = h_y = 12$ cm elsewhere.

The presented results repeatedly confirm the high accuracy of the solution computed with POLCA-T. Recall from the previous chapter that the maximum difference in relative power between POLCA-T and QUANDRY is 0.1%. Comparisons of CUBBOX and AETNA with QUANDRY show maximum differences of 1.3% for CUBBOX and 0.6% for AETNA. The relative differences between POLCA-T and QUANDRY are at several time-points equal to or less than the differences between CUBBOX and AETNA in comparison to QUANDRY (i.e., taking QUANDRY as the reference for comparison). This is to be

---

closer meshes require horizontally varying node sizes, a feature that cannot be modeled in POLCA7.

2Unfortunately, no reference solutions for the assembly power densities are available for the mesh employed. Still, the simplicity of the problem in combination with the very good agreement in eigenvalue results does not challenge the correctness of the TWIGL core model.
Table 6.1.1: TWIGL static eigenvalue results for initial conditions.

<table>
<thead>
<tr>
<th>Code</th>
<th>POLCA7</th>
<th>QUANDRY</th>
<th>AETNA</th>
<th>CONQUEST</th>
<th>SPANDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>400 (8x8 cm²)</td>
<td>400 (8x8 cm²)</td>
<td>400 (8x8 cm²)</td>
<td>144</td>
<td>1600 (4x4 cm²)</td>
</tr>
<tr>
<td></td>
<td>0 ≤ x, y ≤ 24 cm</td>
<td>24 ≤ x, y ≤ 56 cm</td>
<td>56 ≤ x, y ≤ 104 cm</td>
<td>12 cm</td>
<td>16 cm</td>
</tr>
<tr>
<td></td>
<td>104 ≤ x, y ≤ 136 cm</td>
<td>136 ≤ x, y ≤ 160 cm</td>
<td>16 cm</td>
<td>12 cm</td>
<td></td>
</tr>
<tr>
<td>Eigenvalue $k_{eff}$</td>
<td>0.91321</td>
<td>0.91321</td>
<td>0.91321</td>
<td>0.91312</td>
<td>0.91321</td>
</tr>
</tbody>
</table>
Table 6.1.2: Relative power versus time for the TWIGL ramp perturbation problem in comparison with other codes (5 ms step size).

<table>
<thead>
<tr>
<th>Code</th>
<th>CUBBOX</th>
<th>AETNA</th>
<th>QUANDRY</th>
<th>POLCA-T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0 s</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>0.1 s</td>
<td>1.321</td>
<td>1.306</td>
<td>1.308</td>
<td>1.308</td>
</tr>
<tr>
<td>0.2 s</td>
<td>1.985</td>
<td>1.947</td>
<td>1.959</td>
<td>1.961</td>
</tr>
<tr>
<td>0.3 s</td>
<td>2.074</td>
<td>2.068</td>
<td>2.077</td>
<td>2.076</td>
</tr>
<tr>
<td>0.4 s</td>
<td>2.092</td>
<td>2.088</td>
<td>2.090</td>
<td>2.093</td>
</tr>
<tr>
<td>0.5 s</td>
<td>2.109</td>
<td>2.105</td>
<td>2.110</td>
<td>2.111</td>
</tr>
</tbody>
</table>

expected since POLCA-T and QUANDRY are based on very similar methods.

Graphically, the ramp perturbation transient is shown in Figure 6.1.1 where it is compared with the QUANDRY reference solution again. Observe that the transient starts at \( t = 1.0 \) s as the first second is used for the zero-transient. The power rises to more than double the initial level prior to becoming almost constant at the end of the transient. The reactivity progression is illustrated in Figure 6.1.2 with the CONQUEST code as the reference \([9]\).\(^3\) The total amount of reactivity inserted is 384 pcm or 0.5 \( \$ \) (resulting thus in a delayed super-critical transient). The linear increase in reactivity as well as the good agreement with the CONQUEST results confirm again the correctness of the core modeling and perturbation specification.

\(^3\)The reactivity profile was published for the CONQUEST code only.
6.1 TWIGL Reactor Kinetics Problem

Figure 6.1.1: Power versus time for the TWIGL ramp perturbation transient.

Figure 6.1.2: Reactivity profile for the TWIGL ramp perturbation transient.
6 Benchmarking Results and Discussion

6.1.2 Step Perturbation Transient

In this section, the results for the step perturbation transient are described. The transient is induced by a sudden decrease in thermal absorption cross-section in the fuel region “1” (see also Section 4.1). Figures 6.1.3 and 6.1.4 show relative power and reactivity as function of time, respectively. The amount of reactivity inserted is equal to that in the ramp transient (384 pcm) but its sudden initiation induces a much stronger rise in power than it is the case in the ramp transient. Still, the final value for power is almost the same for both transients since the power depends on the final reactor condition solely.

During the computation a constant kinetics time-step size of 10 ms (being the common time-step size for the most published solutions, of which only some are presented here) is employed. The spatial mesh is the same as in the ramp perturbation case.

The results of POLCA-T are compared with those of QUANDRY, AETNA and SPANDEX in Table 6.1.3. Reference values are quoted from [28] and [29]. The maximum difference in relative power between POLCA-T and QUANDRY is less than 0.1% at any time point. It should also be noted that even though POLCA-T employs a larger temporal mesh than some of the other codes, it shows a strong consistency with other solutions (and in particular with the fine-mesh SPANDEX solution) and a stable numerical behavior. In this context, it is noteworthy that Tamitani et al. [29] reported that power oscillations (in time) for AETNA calculations with a time-step size of 10 ms were observed during the first tenth parts of seconds. Such an unstable behavior has not been observed in POLCA-T simulations even for larger kinetics time-step sizes of up to 50 ms.

The TWIGL benchmark solutions reveal that POLCA-T is fully capable to solve two-dimensional spatial kinetics problems. It is thus reasonable to proceed with the more realistic, three-dimensional problems that exhibit larger spatial and temporal flux tilts.

\(^4\)In both figures the transient starts at 1 second as the first second is used to run the zero-transient.
6.1 TWIGL Reactor Kinetics Problem

**Figure 6.1.3:** Power versus time for the TWIGL step perturbation transient.

**Figure 6.1.4:** Reactivity profile for the TWIGL step perturbation transient.
6 Benchmarking Results and Discussion

Table 6.1.3: Relative power versus time for the TWIGL step perturbation problem in comparison with different codes.

<table>
<thead>
<tr>
<th>Code</th>
<th>POLCA-T</th>
<th>QUANDRY</th>
<th>AETNA</th>
<th>SPANDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes*</td>
<td>400</td>
<td>400</td>
<td>400</td>
<td>1600</td>
</tr>
<tr>
<td>Time-step size (ms)</td>
<td>10</td>
<td>2.5</td>
<td>10</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Relative power</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0 s</td>
<td>1.000</td>
</tr>
<tr>
<td>0.1 s</td>
<td>2.061</td>
</tr>
<tr>
<td>0.2 s</td>
<td>2.080</td>
</tr>
<tr>
<td>0.3 s</td>
<td>2.097</td>
</tr>
<tr>
<td>0.4 s</td>
<td>2.114</td>
</tr>
<tr>
<td>0.5 s</td>
<td>2.132</td>
</tr>
<tr>
<td>CPU time</td>
<td>3.88 s</td>
</tr>
</tbody>
</table>

*The corresponding node sizes are also listed in Table 6.1.1.*
6.2 LMW Operational Transient

The LMW Operational Transient is extensively described in Chapter 4. This benchmark problem involves movements of two different control rod banks in a highly simplified PWR reactor. Recall that the thermal-hydraulic feedback is not considered in the present work.

6.2.1 Static Results

The static POLCA7 calculations for the initial core condition are performed with equilateral nodes of 20 cm. The results that are listed in Table 6.2.1 reveals that the static eigenvalue solution is in a very good agreement with other solutions. The difference between POLCA7 and QUANDRY that employed the same node size as well as between POLCA7 and the fine-mesh eigenvalue solution of SPANDEX is 6 pcm only. This difference is within the spread of differences between the various codes, which indicates that the POLCA7 solution is fully acceptable. For the static calculation of the final core state there are no reference static solutions available for the above mentioned codes. However, Ohlsson [24] computed the final conditions with a fine-mesh (4x4x4 cm$^3$) solution of the code MGRAC that also uses the Analytic Nodal Method. The difference between the eigenvalues computed by POLCA7 and MGRAC is 6 pcm as well.

Recall from Chapter 4 that in the LMW core model, the control rod material contains fissile material and hence, the upper axial reflector layer is modeled as part of fuel in POLCA7. This yields a total fuel assembly length (active core height) of 180 cm and an active core volume of 5.544 m$^3$ (recall from Chapter 4 that the active core volume in the reference solutions is 4.928 m$^3$). The core power density computed by POLCA7 becomes thus 133.3 W/cc (instead of 150.0 W/cc as used in the references).

The circumstance has the side effect that the assemblies that do not contain any control rods in their upper part (or, in other words, do not

---

$^5$ All reference values are taken from [24, 26, 28, 29].
6 Benchmarking Results and Discussion

Table 6.2.1: LMW benchmark: Eigenvalue results for both the initial and the final core state.

<table>
<thead>
<tr>
<th>Code</th>
<th>POLCA7</th>
<th>QUANDRY</th>
<th>AETNA</th>
<th>SPANDEX</th>
<th>MGRAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node size*</td>
<td>20x20x20</td>
<td>20x20x20</td>
<td>10x10x5</td>
<td>5x5x2.5</td>
<td>4x4x4</td>
</tr>
<tr>
<td>Initial state</td>
<td>0.99970</td>
<td>0.99974</td>
<td>0.99971</td>
<td>0.99964</td>
<td>0.99964</td>
</tr>
<tr>
<td>Final state</td>
<td>0.99672</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.99666</td>
</tr>
</tbody>
</table>

*The node size (or node volume) is defined as $h_x \times h_y \times h_z$ where $h$ is the node width in the respective direction.

produce any power in their upper part, which is normally composed of water reflector) exhibit too low power densities as their “active” volume $V_i$ includes non-power generating nodes as well.\(^6\) Consequently, the normalized fuel assembly power densities cannot be directly compared between POLCA7 and reference solutions since the latter model the upper reflector layer (including the control rods) as the reflector region exclusively, thereby implying an active core height of 160 cm. (However, the total core power accounts even for the power generated in the rodded reflector regions as becomes evident from simply re-scaling the core power density calculated by POLCA7 by the factor of 180/160 that yields then a core power density of 150.0 W/cc again).

Therefore, the POLCA7 power density distribution has to be recalculated in order to ensure the comparability with other solutions. The recalculation is done in two steps:

1. The computed normalized fuel assembly densities $\bar{P}^n_i$, where $i$

\(^6\)See also Chapter 4, where the normalization procedure in POLCA7 is outlined.
denotes the \(i\)-th fuel assembly, have to be adjusted for the correct length:

\[
\bar{P}_i^* = \frac{P_{n,i}l_{sol}}{l_{ref}}
\]

where \(l_{sol}\) denotes the fuel bundle length in POLCA7 (180 cm) and, correspondingly, \(l_{ref}\) is the fuel bundle length in reference solutions (160 cm). Clearly, this recalculation is applied only to fuel assemblies that do not contain control rods in their upper part. Fuel assemblies with control rods are not affected by this recalculation since their power densities are in fact correct.

2. The so obtained new mean reactor power density, which is defined to be (recall from Chapter 4 that the normalization in POLCA7 is done such that the mean reactor power density is 1000):

\[
\frac{\sum_i \bar{P}_i^*}{N} = 1000
\]

yields now incorrect results (the core volume includes the rodded fuel bundles with the “wrong” active length of 180 cm). Hence, all fuel assembly power densities have to be corrected for the difference by multiplying with a factor \(k\) computed as

\[
k = \frac{N \cdot 1000}{\sum_i \bar{P}_i^*} \rightarrow k\bar{P}_i^* = P_{i,sol}^n
\]

with \(P_{i,sol}^n\) being the wanted normalized assembly power density for the \(i\)-th assembly with the corrected length. \(N\) is the total number of fuel bundles in the core.

The recalculated normalized assembly power densities for the initial state are illustrated in Figure 6.2.1, where they are compared with the fine-mesh solution of MGRAC. The latter employed nodes with dimensions of 4x4x4 cm\(^3\) [24]. Following the proposition from Lawrence [16], namely that the assembly-averaged power densities should be computed to within 2% of the reference solution in order to be sufficiently accurate, one can conclude that this POLCA7 solution is in an exceptionally good conformance with the reference. The maximum difference in assembly power density is 0.6% only.
6 Benchmarking Results and Discussion

Figure 6.2.1: LMW benchmark, initial condition: Normalized assembly power densities in comparison with MGRAC (octal symmetry is applied).

Analogously, the assembly power distribution map for the final condition is depicted in Figure 6.2.2. Even here, the maximum difference between POLCA7 and MGRAC (for MGRAC results it is referred to [24]) is 0.5% at most. Thus, the accuracy of both the initial and final static solutions computed by POLCA7 is positively confirmed.

6.2.2 Transient Results

The transient progression is illustrated in Figure 6.2.3 where the transient starts at 1 second as the first second is used to run the zero-transient. The POLCA-T simulation employs a 20x20x20 cm³ mesh and a constant kinetics time-step size of 250 ms. Following the extraction of control rods, the power rises and peaks at more than 170% of the initial power about 20 seconds into the transient. The increasing CR fraction in the core due to the insertion of the second CR bank forces the relative power to drop to values around 38%. The reactivity shape that so far was only published for the CONQUEST code [9] is
6.2 LMW Operational Transient

Figure 6.2.2: LMW benchmark, final condition: Normalized assembly power densities in comparison with MGRAC (octal symmetry is applied).

plotted in Figure 6.2.4. CONQUEST employed the same spatial and temporal mesh as POLCA-T. (Note that the minor differences in reactivity between POLCA-T and CONQUEST are traced back to the fact that CONQUEST values were read off a plot with a limited resolution resulting in some error in determination of individual values).

As already pointed out by Smith and other authors, the so-called control rod cusping effect is especially pronounced in the LMW transient. That is, the error in determination of volume-weighted, homogenized cross-sections for partially rodded nodes can become significant. The error results then in an unphysical behavior of the total reactivity, whose profile shows graphically a cusp-like shape at those time-points where the control rod crosses node boundaries. This behavior leads consequently to an underestimation of the total reactivity and core power.

In real-life calculations, errors in cross-section homogenization are also caused by the presence of spacers, burnable absorbers, etc. In POLCA7,
6 Benchmarking Results and Discussion

Figure 6.2.3: Power versus time for the LMW operational transient.

Figure 6.2.4: Reactivity versus time for the LMW operational transient.
a special scheme (called AXHOM) is employed in order to minimize errors caused by these heterogeneities. The rod cusping effect during the POLCA-T calculations becomes also evident in Figure 6.2.5. Here, two reactivity profiles are plotted with focus on the the power peak region (between 6 and 28 seconds into the transient): one without any correction for the cusping effect and the other with activated AXHOM procedure. The cusp-like behavior is clearly seen at indicated time-points where the two CR banks traverse the node boundaries.

The impact of the cusping effect on the power progression is demonstrated in Figure 6.2.6. The “uncorrected” power profile shows a clear underestimation of power, especially in the aforementioned power peak region. Therefore, it is of particular interest to scrutinize the POLCA7 axial homogenization procedure in more detail while solving the LMW benchmark problem. This procedure is extensively described in [18] and hence, the scheme is outlined here only briefly. Given exact geometry data, node-averaged fluxes and material properties, the procedure does:
6 Benchmarking Results and Discussion

Figure 6.2.6: LMW transient results with and without cusp correction method AXHOM.

- Divide nodes axially into subnodes where heterogeneities appear. That is, in the present case, the partially rodded nodes are split into two axial subnodes: one consisting of rodded material and the other of unrodded material.
- Solve analytically the 1D-diffusion equation in axial direction for the entire affected assembly.
- Compute new flux-weighted cross-sections that are homogeneous within each partially rodded node. Also compute axial flux discontinuity factors of the axial sides of the homogenized node(s).

Table 6.2.2 presents solutions of simulations with and without activation of the axial homogenization procedure AXHOM. Moreover, the node size in axial direction is halved in order to reduce the rod cusp effect additionally. The kinetics time-step size is set constant to 250 ms. All results are compared with the reference solution that was obtained.
6.2 LMW Operational Transient

by Richardson extrapolation of several CUBBOX solutions and presented by Smith [26]. The choice of the reference solution is motivated by the fact that CUBBOX does not require any special procedure for treating the rod cusping effect since the cross-sections do not need to be homogeneous inside each node. Note, however, that the reference solution is not considered to be more accurate than 1% [9, 26].

The comparison between POLCA-T and the extrapolated CUBBOX solution indicates obviously that the axial homogenization procedure reduces the relative differences in between, in particular in cases with larger axial nodes. Here, the differences are cut by more than half so that the solution with an axial node size of 20 cm and activated AXHOM yields fairly acceptable results already. Halving the node size in axial direction also enhances the accuracy of the results so that the largest differences are minimized from 5.4% to 1.8%. However, the latter is probably not a justifiable approach considering the large increase in computational time.

A comparison of POLCA-T results with other codes is given in Table 6.2.3. All the presented solutions are considered to be well converged in the time variable [28]. QUANDRY, AETNA and SPANDEX are reported to have used their respective cusping correction methods. For POLCA-T, two solutions are given: one for a constant kinetics time-step size of 250 ms and the other one for variable kinetics time-steps. For the “variable” simulation, the maximum allowed feedback time-step size is 1.0 s. Other codes, with except of CUBBOX and SPANDEX, used a constant kinetics time-step size of 250 ms as well. The CUBBOX solution was obtained by using constant 125 ms time-steps, while SPANDEX employed a variable time-step size algorithm as described in [3, 28]. Note that the radial spatial mesh used in POLCA-T calculations corresponds to fuel assembly dimensions as they are prevalent in PWR reactors, reflecting so highly realistic conditions.

As for the constant 250 ms solution, one would initially expect that the results obtained by POLCA-T are close to those of QUANDRY,
Table 6.2.2: LMW operational transient: Examination of the axial homogenization procedure.

<table>
<thead>
<tr>
<th>Number of axial layers/axial node size</th>
<th>AXHOM active</th>
<th>CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/20</td>
<td>Yes</td>
<td>60.0</td>
</tr>
<tr>
<td>18/10</td>
<td>No</td>
<td>38.6</td>
</tr>
<tr>
<td>30.24</td>
<td>Yes</td>
<td>50.3</td>
</tr>
<tr>
<td>34.60</td>
<td>No</td>
<td>40.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Power (%)</th>
<th>$\epsilon_{rel}$</th>
<th>Power (%)</th>
<th>$\epsilon_{rel}$</th>
<th>Power (%)</th>
<th>$\epsilon_{rel}$</th>
<th>Power (%)</th>
<th>$\epsilon_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5.0</td>
<td>112.8</td>
<td>0.1</td>
<td>112.2</td>
<td>0.0</td>
<td>112.4</td>
<td>0.0</td>
<td>112.9</td>
<td>0.0</td>
</tr>
<tr>
<td>10.0</td>
<td>134.1</td>
<td>0.4</td>
<td>132.4</td>
<td>0.0</td>
<td>134.5</td>
<td>0.1</td>
<td>134.1</td>
<td>0.4</td>
</tr>
<tr>
<td>20.0</td>
<td>171.3</td>
<td>1.4</td>
<td>166.5</td>
<td>2.1</td>
<td>171.8</td>
<td>2.1</td>
<td>170.4</td>
<td>1.9</td>
</tr>
<tr>
<td>30.0</td>
<td>137.0</td>
<td>2.1</td>
<td>132.4</td>
<td>5.4</td>
<td>137.4</td>
<td>2.1</td>
<td>137.0</td>
<td>2.0</td>
</tr>
<tr>
<td>40.0</td>
<td>80.9</td>
<td>1.4</td>
<td>78.8</td>
<td>4.6</td>
<td>81.1</td>
<td>0.1</td>
<td>80.9</td>
<td>1.0</td>
</tr>
<tr>
<td>50.0</td>
<td>50.3</td>
<td>1.4</td>
<td>49.2</td>
<td>3.5</td>
<td>50.4</td>
<td>1.2</td>
<td>50.3</td>
<td>1.4</td>
</tr>
<tr>
<td>60.0</td>
<td>38.6</td>
<td>1.2</td>
<td>37.8</td>
<td>3.2</td>
<td>38.7</td>
<td>0.9</td>
<td>38.6</td>
<td>1.2</td>
</tr>
</tbody>
</table>

CPU time (s): 34.60, 30.24, 70.24, 62.33

Table 6.2.2: LMW operational transient: Examination of the axial homogenization procedure.
### 6.2 LMW Operational Transient

#### Table 6.2.3: LMW operational transient: Comparison between different codes.

<table>
<thead>
<tr>
<th>Code</th>
<th>Node size [cm$^3$]</th>
<th>Number of kinetics time-steps</th>
<th>Relative power (%)</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLCA-T</td>
<td>20x20x20</td>
<td>77*</td>
<td>100.0</td>
<td>1</td>
</tr>
<tr>
<td>CUBBOX</td>
<td>20x20x20</td>
<td>480</td>
<td>100.0</td>
<td>-</td>
</tr>
<tr>
<td>QUANDRY</td>
<td>20x20x10</td>
<td>240</td>
<td>100.0</td>
<td>-</td>
</tr>
<tr>
<td>AETNA</td>
<td>10x10x5</td>
<td>240</td>
<td>100.0</td>
<td>-</td>
</tr>
<tr>
<td>PANTHER</td>
<td>5x5x2.5</td>
<td>160*</td>
<td>100.0</td>
<td>-</td>
</tr>
</tbody>
</table>

- Node size: [cm$^3$]
- Number of kinetics time-steps
- Relative power (%)
- CPU time

*Variable kinetics time-step sizes were employed.*
which is most similar to POLCA-T in methods and modeling. However, POLCA-T agrees better with other codes where a much refined spatial mesh is used. The maximum difference between POLCA-T and PANTHER or between POLCA-T and SPANDEX is 0.9% only. On the contrary, the difference between POLCA-T and QUANDRY is 2.5% at most. These discrepancies arise probably due to the differences between their respective cusping correction methods. In QUANDRY, Smith implemented a model, where the axial flux shape in the partially rodded node is approximated by a quadratic polynomial that takes into account the average flux in the node in question as well as in the both adjacent nodes. The polynomial is then used to determine flux-volume weighted, homogenized cross-sections in the affected (i.e., partially rodded) node [26]. However, Smith pointed out [26] that this cusping correction model still leaves room for further improvement. In SPANDEX, a slightly more sophisticated approach is used to estimate the subnode flux in rodded nodes. However, none of these methods solves the heterogeneous problem for the partially rodded node. But POLCA7 does. Consequently, the cusping correction method employed in POLCA7 is probably better than that used in QUANDRY and SPANDEX, a fact that also explains the very good agreement of POLCA-T results with fine-mesh solutions.

The high conformance with other similar codes is also shown for the “variable” POLCA-T solution, which with 77 variable kinetics time-steps employs the coarsest temporal mesh. The results are not considerably affected by reduced number of kinetics time-steps. Here, the maximum difference in comparison of POLCA-T to PANTHER and SPANDEX is 1.3%. At the same time, the computational costs are 1.9 times lower than those of the simulation with constant kinetics time-steps. This fact does additionally demonstrate a sound performance of the automated time-stepping algorithm utilized in POLCA-T. Based on the results and the discussion above, one can conclude that the presented POLCA-T solution for the LMW transient is fully acceptable. The LMW benchmark confirms repeatedly the correctness of the neutron kinetics model in POLCA-T.
6.3 LRA BWR Kinetics Problem

The LRA BWR Kinetics Problem is described in detail in Chapter 4. In the four-rod case, four transient control rods are fully ejected from the core during a time period of two seconds. The single-rod transient considers a rapid withdrawal of one transient control rod at the same speed. In both cases, the reactor is initially in low power conditions.

6.3.1 Static Results

Static calculations are performed both for the initial core configuration (i.e., the transient control rods are still inserted in the core) and for the final core configuration (with the control rod(s) being removed). Both initial and final conditions are at the initial fuel temperature of 300K. The final state is in addition computed for both the four-rod and the single-rod versions. Table 6.3.1 provides the POLCA7 eigenvalue results, which are compared with QUANDRY, CUBBOX and IQSBOX [2, 26]. CUBBOX and IQSBOX apply 15x15 cm\(^2\) mesh on radial plane and an axial node spacing of 30cm, whereas CUBBOX uses 15cm in the reflector regions (the numbers in brackets indicate the axial node spacing in the reflector area). In the four-rod case, QUANDRY uses a finer radial mesh than POLCA7 but a comparatively coarse axial one (25cm in the active core and 15cm in the reflector area). CUBBOX and IQSBOX solutions for the final core state of the single-rod case are not available.

For the initial and the four-rod final core state one can observe a very good agreement between POLCA7 and QUANDRY results, which differ not more than 2pcm only. Even in the single-rod final core state, where the flux perturbations are as largest (compared to other two conditions), the difference is 9pcm only. The differences between POLCA7 and CUBBOX and IQSBOX, which are 15 and 17pcm, respectively, are still moderate in the initial state. In the final state these discrepancies increase to 33pcm at most. It is assumed that in conditions, where such large spatial flux gradients are prevalent, the differences between various spatial kinetics methods become much more pronounced.
## Benchmarking Results and Discussion

Table 6.3.1: LRA benchmark problem: Eigenvalues for initial and final core configurations.

<table>
<thead>
<tr>
<th>Code</th>
<th>Node size (cm³)</th>
<th>Core configuration</th>
<th>Initial state</th>
<th>Final state, four-rod case</th>
<th>Final state, single-rod case</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLCA7</td>
<td>15x15x25 (15)</td>
<td>N/A</td>
<td>0.99641</td>
<td>1.01550</td>
<td>1.01485</td>
</tr>
<tr>
<td>QUANDRY</td>
<td>15x15x30 (15)</td>
<td>N/A</td>
<td>0.99639</td>
<td>1.01549</td>
<td>1.01449*</td>
</tr>
<tr>
<td>CUBBOX</td>
<td>15x15x30 (15)</td>
<td>N/A</td>
<td>0.99626</td>
<td>1.01517</td>
<td>1.0150</td>
</tr>
<tr>
<td>IQSBOX</td>
<td>7.5x7.5x25 (15)</td>
<td>N/A</td>
<td>0.99641</td>
<td>1.01518</td>
<td>0.99641</td>
</tr>
</tbody>
</table>

*For the calculation of the single-rod final core state QUANDRY employed another spatial mesh, which is 15x15x25 (15) cm³.*
6.3 LRA BWR Kinetics Problem

**Figure 6.3.1:** LRA benchmark problem: Normalized fuel assembly power densities for the initial configuration (octal symmetry is applied).

The normalized fuel assembly power densities are illustrated for the initial conditions in Figure 6.3.1. Hereby, they are compared with the fine-mesh solution of QUANDRY, where nodes with square dimensions of 7.5x7.5 cm² were employed. The node height in the QUANDRY reference solution was set to 25 cm in the active core and to 15 cm in the axial reflector region.

Similarly, the final power density map for the four-rod case is shown in Figure 6.3.2, where QUANDRY calculations with the same mesh as in the initial state are used as the reference. In both cases, the maximum differences do not exceed 0.5% (which is well below the “Lawrence-criterion” of 2% being mentioned in Section 6.2) and that is comparable to the differences between various QUANDRY solutions (coarse mesh vs. fine-mesh) that amount to 0.45% at most [26].
6 Benchmarking Results and Discussion

<table>
<thead>
<tr>
<th>POLCA7</th>
<th>QUANDRY (Ref.)</th>
<th>ε (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>359</td>
<td>385</td>
<td>470</td>
</tr>
<tr>
<td>0,0</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>569</td>
<td>564</td>
<td>670</td>
</tr>
<tr>
<td>0,0</td>
<td>0,2</td>
<td>0,0</td>
</tr>
<tr>
<td>630</td>
<td>499</td>
<td>557</td>
</tr>
<tr>
<td>0,0</td>
<td>-0,5</td>
<td>0,0</td>
</tr>
<tr>
<td>523</td>
<td>406</td>
<td>453</td>
</tr>
<tr>
<td>0,0</td>
<td>0,2</td>
<td>0,1</td>
</tr>
<tr>
<td>300</td>
<td>292</td>
<td>352</td>
</tr>
<tr>
<td>0,0</td>
<td>0,1</td>
<td>0,0</td>
</tr>
<tr>
<td>193</td>
<td>209</td>
<td>265</td>
</tr>
<tr>
<td>0,0</td>
<td>0,3</td>
<td>0,0</td>
</tr>
<tr>
<td>146</td>
<td>147</td>
<td>249</td>
</tr>
<tr>
<td>0,0</td>
<td>0,0</td>
<td>0,0</td>
</tr>
<tr>
<td>193</td>
<td>152</td>
<td>168</td>
</tr>
<tr>
<td>0,0</td>
<td>0,0</td>
<td>0,0</td>
</tr>
</tbody>
</table>

Figure 6.3.2: LRA four-rod case: Normalized fuel assembly power densities for the final configuration.

Finally, the normalized assembly power densities for the final configuration of the single-rod version are presented in Figure 6.3.3. Here, the POLCA7 results shown in the first row are compared with the fine-mesh (5x5x5 cm³) solution of MGRAC [24] and the relative differences are found in the second row. The maximum difference is 1.5%, while the absolute error for the assembly in question is - owing to a very low power in this assembly - fairly low. This compliance with the "Lawrence-criterion" repeatedly indicates that the static calculations have converged. The assembly power density map after the transient CR withdrawal is illustrated in Figure 6.3.4 where the large flux tilts become more apparent.

---

8QUANDRY used a coarser mesh for the full core final state calculation and hence, the fine-mesh solution of MGRAC is considered to be more suitable for a comparison with POLCA7.
Figure 6.3.3: LRA single-rod case: Normalized fuel assembly power density for the final configuration.
Figure 6.3.4: Assembly power density map for the final conditions in the single-rod case. The normalized power densities are plotted on the z-axis.
Table 6.3.2: LRA benchmark problem: Temporal mesh (feedback time-step sizes) employed during the transient calculations.

<table>
<thead>
<tr>
<th>Time interval*</th>
<th>Time-step size $(\Delta T_{max} = \Delta T_{min})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.0 &lt; t \leq 1.5 \text{ s}$</td>
<td>25 ms</td>
</tr>
<tr>
<td>$1.5 &lt; t \leq 1.6 \text{ s}$</td>
<td>10 ms</td>
</tr>
<tr>
<td>$1.6 &lt; t \leq 1.7 \text{ s}$</td>
<td>2.5 ms</td>
</tr>
<tr>
<td>$1.7 &lt; t \leq 1.8 \text{ s}$</td>
<td>1.25 ms</td>
</tr>
<tr>
<td>$1.8 &lt; t \leq 1.95 \text{ s}$</td>
<td>1.0 ms</td>
</tr>
<tr>
<td>$1.95 &lt; t \leq 2.0 \text{ s}$</td>
<td>2.5 ms</td>
</tr>
<tr>
<td>$2.0 &lt; t \leq 3.0 \text{ s}$</td>
<td>20 ms</td>
</tr>
<tr>
<td>$3.0 &lt; t \leq 4.0 \text{ s}$</td>
<td>25 ms</td>
</tr>
</tbody>
</table>

*Note that the transient starts at 1.0 s, as the first second is used to run the zero-transient.

6.3.2 LRA Four-Rod Transient

The LRA four-rod transient, initiated by a rapid withdrawal of four peripheral control rods, is solved with the same spatial mesh as it was used in the static calculations. As already mentioned above, large changes of flux in time require utilization of more stringent options in POLCA-T than the previous problems. So, the numbers of NONLIN, MINFLX and MAXFLX are set to 50, 20 and 100, respectively. The convergence criterion $\varepsilon_{feed}$ is kept at $10^{-3}$. Variable kinetics time-steps are employed with the feedback time-step sizes $\Delta T_{max}$ and $\Delta T_{min}$ adjusted to the temporal mesh utilized in QUANDRY (and many other) calculations [26, 28, 29]. Hence, a better comparability between the results is provided. The feedback time-step sizes are stated in Table 6.3.2.

The power progression and the average fuel temperature are shown in
Figures 6.3.5 and 6.3.6, respectively. The rapid drop of the control rods causes the power to rise by more than ten orders of magnitude. The increase in power induce an almost stepped jump in fuel temperature that, on its turn, forces the power to reverse at about 0.9 s into the transient. Afterwards, the fuel temperature continues to increase moderately until it reaches approximately 1000 K at the end of the transient. The power shape shows a second, lower, power peak at about 1.5 s into the transient before it stabilizes at values slightly below 100 W/cc.

A comparison of POLCA-T results with other solutions that employed nearly the same spatial mesh is provided in Table 6.3.3. All values are quoted from [2, 9, 29]. Even though the difficulty of the problem makes a direct comparison between the individual results rather difficult (one can clearly see the wide spreading of different values), it is obvious that POLCA-T results are in line with other codes for any computed parameter. The spreading of data is of the same order of magnitude in cases where the problem is solved with the same code but with varying
temporal and spatial meshes. In QUANDRY calculations, for instance, values for the first power peak vary from 5532 W/cc to 6549 W/cc, depending on spatial and temporal meshes employed [6, 26].

As concerns the potential control rod cusping effect, Smith [26] found out that this is not especially pronounced in the LRA problem. He simulated the transient by employing two different axial node sizes without observing any notable differences between the two solutions. This might also be explained by the fact that the cusping effect depends on the time it takes for a control rod to pass through a node. By short CR dropping times (in the LRA problem it takes only 0.1 seconds for a control rod to pass through a node) the dips in the reactivity profile become less extended as compared to transients with slow CR moving times (in the LMW problem it takes 6.7 seconds for a control rod to pass through a node). Hence, the reactivity (and power) profile approximates the “exact” solution in the case of fast transients.
### Table 6.3.3: LRA four-rod transient: Comparison of POLCA-T results with other codes. Power densities are given in W/cc.

<table>
<thead>
<tr>
<th>Node size</th>
<th>Code</th>
<th>Node size [cm$^3$]</th>
<th>Number of kinetic time-steps</th>
<th>Time to first peak (s)</th>
<th>Power at first peak (s)</th>
<th>Time to second peak (s)</th>
<th>Power at second peak (s)</th>
<th>Power at $t = 3.0$ s (K)</th>
<th>Average fuel temperature at $t = 3.0$ s (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15x15x30</td>
<td>POLCA-T</td>
<td>15x15x30</td>
<td>1200</td>
<td>0.900</td>
<td>5754</td>
<td>0.900</td>
<td>368</td>
<td>1011</td>
<td>60.5</td>
</tr>
<tr>
<td>15x15x30</td>
<td>CUBBOX</td>
<td>15x15x30</td>
<td>1373</td>
<td>0.906</td>
<td>6268</td>
<td>0.993</td>
<td>38.2</td>
<td>1011</td>
<td>60.5</td>
</tr>
<tr>
<td>15x15x30</td>
<td>CONQUEST</td>
<td>15x15x30</td>
<td>2708</td>
<td>0.914</td>
<td>5532</td>
<td>1.00</td>
<td>128.2</td>
<td>1018</td>
<td>60.5</td>
</tr>
<tr>
<td>15x15x30</td>
<td>IQSBOX</td>
<td>15x15x30</td>
<td>706</td>
<td>0.894</td>
<td>5798</td>
<td>1.00</td>
<td>125.0</td>
<td>1018</td>
<td>60.5</td>
</tr>
<tr>
<td>15x15x30</td>
<td>AETNA</td>
<td>15x15x30</td>
<td>410</td>
<td>0.905</td>
<td>5390</td>
<td>1.00</td>
<td>125.0</td>
<td>1018</td>
<td>60.5</td>
</tr>
<tr>
<td>15x15x30</td>
<td>POLCA-T</td>
<td>15x15x30</td>
<td>1200</td>
<td>0.900</td>
<td>5754</td>
<td>0.900</td>
<td>368</td>
<td>1011</td>
<td>60.5</td>
</tr>
</tbody>
</table>

Table 6.3.3: LRA four-rod transient: Comparison of POLCA-T results with other codes. Power densities are given in W/cc.
6.3 LRA BWR Kinetics Problem

6.3.3 LRA Single-Rod Transient

The single-rod case is not as widely used as its four-rod counterpart and the amount of published results is limited. This problem causes even more severe flux gradients across the core (compared to the four-rod case) due to the asymmetry of the core configuration. On the other hand, the power density peak is with approximately 1500 W/cc somewhat lower than in the previous case. Otherwise, the power progression, which is plotted in Figure 6.3.7 is very similar to the four-rod transient. During the calculations, POLCA-T employed the same numerical values and temporal mesh as those in the four-rod case. However, the computation time is about five times higher than it is in the four-rod version. Likewise, the single-rod version requires more kinetics time-steps.

Table 6.3.4 summarizes the transient results that can be found in the public domain so far [28]. POLCA-T, QUANDRY and PANTHER
employed the same temporal mesh as in the four-rod case (see also Table 6.3.2). An automated time-stepping algorithm was used in the SPANDEX solution. Similarly to the quarter-core results, the spread in the data is relatively large but even here, POLCA-T show acceptable agreement with other solutions. In other words, all computed parameters are well within the presented spectrum of results. It can be concluded that despite of the complexity of the problem, POLCA-T computed the results with sufficient accuracy and numerical stability.
Table 6.3.4: LRA single-rod transient: Comparison of POLCA-T results with other codes. Power densities are given in W/cc.

<table>
<thead>
<tr>
<th>Code</th>
<th>POLCA-T</th>
<th>SPANDEX</th>
<th>QUANDRY</th>
<th>PANTHER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node size [cm³]</td>
<td>15x15x15 (30)</td>
<td>15x15x30 (15)</td>
<td>See below.</td>
<td></td>
</tr>
<tr>
<td>Number of kinetics time-steps</td>
<td>1419</td>
<td>1430</td>
<td>820</td>
<td>820</td>
</tr>
<tr>
<td>Power at first peak</td>
<td>1506</td>
<td>1478</td>
<td>1435</td>
<td>1514</td>
</tr>
<tr>
<td>Time to first peak (s)</td>
<td>0.945</td>
<td>0.956</td>
<td>0.950</td>
<td>0.950</td>
</tr>
<tr>
<td>Power at first minimum</td>
<td>33.1</td>
<td>36.4</td>
<td>20.7</td>
<td>36.7</td>
</tr>
<tr>
<td>Time to first minimum (s)</td>
<td>1.02</td>
<td>1.06</td>
<td>1.08</td>
<td>1.08</td>
</tr>
<tr>
<td>Power at second peak</td>
<td>106</td>
<td>109</td>
<td>141</td>
<td>170</td>
</tr>
<tr>
<td>Time to second peak (s)</td>
<td>1.52</td>
<td>1.62</td>
<td>1.57</td>
<td>1.52</td>
</tr>
<tr>
<td>Power at t = 3.0 s</td>
<td>21.9</td>
<td>22.3</td>
<td>22.6</td>
<td>23.3</td>
</tr>
</tbody>
</table>

Mesh intervals for QUANDRY and PANTHER (based on a quarter-core representation):

- $0 < x, y < 15$ and $105 < x, y < 135$: 15 cm
- $15 < x, y < 105$ and $135 < x, y < 165$: 30 cm
- $0 < z < 30$ and $330 < z < 360$: 30 cm
- $30 < z < 330$: 37.5 cm
7 Summary and Conclusions

The objective of this thesis was to perform benchmarking of the three-dimensional, multigroup neutron kinetics model implemented in the transient core simulator POLCA-T. The work was done by simulating some two- and three-dimensional kinetics benchmark problems and comparing the obtained results to those of other codes. Additionally, impacts of different numerical parameters on the numerical stability of the model were tested by carrying out a numerical study.

In Chapter 2, some conventional methods for the solution of the time-dependent, multidimensional neutron diffusion equation were presented. The discussion included both the spatial discretization techniques and the time-integration methods. The main focus was put on those methods, which are employed in codes that were used as references in this work.

The utilization of the Analytic Nodal Method and the Theta Method in the neutron kinetics model of POLCA-T was briefly explained in Chapter 3. The considered benchmark problems as well as the benchmarking procedure were extensively described in Chapter 4.

The results of the numerical study that was performed by simulating the TWIGL ramp perturbation transient were presented in Chapter 5. This study has examined the sensitivity of the solution accuracy to variations in different numerical settings. These settings included, inter alia, the kinetics time-step size, the maximum allowed number of various iteration levels and the convergence criteria. It was shown that even large variations in these options did hardly affect the computed solution.

This study demonstrated clearly that the POLCA-T neutronics provide highly stable results that are rather insensitive to significant variations
in input, for example. Additionally, the possibility to employ a coarser
temporal mesh, a lower number of iterations and a looser convergence
criteria without sacrificing the solution accuracy leads to a notable re-
duction in computing time. The numerical study has also provided
preferable options to be used during the subsequent benchmarking cal-
culations, whose results were addressed in Chapter 6.
The computed solutions for all benchmark problems were found to be
spatially and temporally well converged. These results were fairly con-
sistent with the corresponding solutions of other codes. Most notably,
the agreement between the results was substantial in comparisons to
those solutions, where finer spatial and temporal meshes were employed
than in POLCA-T.
Based on the project outcome, one can therefore conclude that the
POLCA-T neutron kinetics is modeled and implemented correctly. It
is a numerically very stable, accurate and reliable computational tool
that is fully capable to solve problems the code is confronted with.
Bibliography


[10] L. A. Hageman and J. B. Yasinsky. Comparison of Alternating-Direction Time-Differencing Methods with Other Implicit Meth-
Bibliography


