A Coupled Calculation Code System for the Thorium Molten Salt Reactor

M.W. Hoogmoed
PNR-131-2009-009

September 2009
Foreword

This report presents the final thesis of the Master of Science programme in Applied Physics at the Delft University of Technology. The project has been performed at the section Physics of Nuclear Reactors of the faculty of Applied Sciences.

Thanks are going out to Jan Leen Kloosterman and Danny Lathouwers for their guidance during this project. Furthermore I would like to thank Karoly Nagy for answering lots of questions, and all the rest of the staff at PNR for making this project an interesting and pleasant experience.

Michiel Hoogmoed

Delft, September 2009
Summary

The Generation IV International Forum has selected six groups of nuclear reactor designs that share some beneficial properties relative to the current nuclear reactors. The nuclear waste they produce should last a few hundred years instead of millennia, and their energy yield should be 100-300 times higher with the same amount of fuel used in the older type reactors. Also they should have the ability to consume existing nuclear waste in the production of electricity. One of these six groups within the Generation IV reactors is the class of Molten Salt Reactor concepts.

One of the designs that have been put forward in the class of Molten Salt Reactors, is the non-moderated Thorium Molten Salt Reactor (TMSR-NM). This reactor, with an epithermal neutron spectrum, is able to use the thorium fuel cycle. Thorium is dissolved in the fuel salt and in the fertile blankets that surround the core. As this is a non-moderated reactor, no complex graphite structure is required, which simplifies the construction.

A coupled code system was required to perform simulations on this reactor model, but was not yet available at the PNR section of the Delft University of Technology. The in-house code systems DALTON and HEAT where combined to perform coupled calculations of the neutronics and the thermal hydraulics of the reactor. To achieve this, a code system had to be created to enable communication between DALTON and HEAT, and the individual programs had to be altered to be compatible with the TMSR-NM model.

When the coupled code system was established, verification has been performed by several tests. To validate the performance of the code system, simulation results have been compared with similar research on the TMSR-NM by CNRS [7]. The verification tests indicated that the simulation code system is working properly. Some of the simulation results have been validated with CNRS research, other simulation results indicated that some aspects of the reactor models used in this project and by CNRS do not correspond.

A steady state operation has been achieved at a temperature of 700°C with a power output of nearly 3GWth. This steady state was established with the fluoride salt LiF-ThF4-UO4 with a molar concentration of heavy nuclei of 29.5%. The temperature feedback of the fuel salt has been analyzed, and was found to be $\alpha \approx -6 \text{ pcm/K}$. Finally, two transient scenario’s where simulated with the code system, a failure of the heat exchanger and a pump failure. Both cases showed behavior as expected, after a swift temperature increase, power production fell down to zero because of the negative feedback coefficient.
Contents

I Thesis 9

1 Introduction 10
1.1 The Role of Nuclear Power in the Global Energy Supply and the Climate Change Problem 10
1.2 Molten Salt Reactors 11
1.3 Non-Moderated Thorium Molten Salt Reactor 12
1.3.1 Reactor design 13
1.3.2 Computational model 14
1.4 Objectives 17
1.5 Outline 17

2 Reactor neutronics 19
2.1 Introduction 19
2.2 Theory 19
2.2.1 Diffusion theory 19
2.2.2 Multigroup diffusion theory 20
2.2.3 Solving the multigroup diffusion equations: DALTON 22
2.2.4 Power scaling 22
2.2.5 Temperature feedback coefficient 22
2.3 Influence of precursors on the simulation 23
2.4 Summary 24

3 Neutron cross section processing 26
3.1 Introduction 26
3.2 Group constants 26
3.3 Processing group constants: SCALE 27
3.4 Cross section work library 28
3.5 Cross section interpolation: MIX 28
3.6 Summary 29

4 Thermal hydraulics of TMSR-NM: HEAT 30
4.1 Introduction 30
4.2 Theory 30
4.2.1 Flow field 30
4.2.2 Turbulence model ($k - e$) 31
4.2.3 Heat balance 33
4.3 Summary 35

5 Reactor material properties 36
5.1 Introduction 36
5.2 Reactor materials 36
5.3 Material properties 37
5.3.1 Number density 37
5.3.2 Density 37
5.3.3 Heat capacity 38
5.3.4 Viscosity 38
## CONTENTS

5.3.5 Thermal conductivity .............................................. 38
5.4 Summary .............................................................. 39

6 Coupling of the code systems ........................................... 40
6.1 Introduction .......................................................... 40
6.2 Code systems ......................................................... 40
6.3 Coupled code system ................................................ 41
6.4 Summary .............................................................. 41

7 Results ................................................................. 43
7.1 Introduction .......................................................... 43
7.2 Temperature feedback coefficient ................................... 43
7.3 Steady state operation ............................................... 44
    7.3.1 $k_{ef}$ and power production .................................. 44
    7.3.2 Steady state reactor properties ................................ 46
7.4 Heat transfer coefficient ............................................ 49
7.5 Fuel composition .................................................... 49
7.6 Grid resolution ....................................................... 51
7.7 Power scaling rate .................................................. 51
7.8 Convection of precursors ............................................ 53
7.9 Accident scenario's .................................................. 55

8 Conclusions ........................................................... 59

9 Recommendations ....................................................... 60

II Manual of MSR-CC ..................................................... 62

A Introduction .......................................................... 64
A.1 About MSR-CC ....................................................... 64
A.2 About the manual .................................................... 64
A.3 Outline ............................................................... 64

B MSR-CC .............................................................. 65
B.1 Structure of the MSR-CC ............................................. 65
B.2 Executing a MSR-CC simulation ................................... 65

C SCALE: Creating the cross section work library ...................... 67
C.1 Introduction ......................................................... 67
C.2 Structure of the library ............................................. 67
C.3 Routines and execution ............................................. 68
    C.3.1 makescale.f90 .................................................. 68
    C.3.2 runbatch5 ....................................................... 68
    C.3.3 quemakelibrary ............................................... 69
    C.3.4 Execution ...................................................... 69
CONTENTS

D MIX: Cross section interpolation 70
  D.1 Introduction ................................... 70
  D.2 Routines and execution .......................... 70
    D.2.1 mix ...................................... 70
    D.2.2 wax_ice ................................... 71
    D.2.3 batch5 .................................... 71
    D.2.4 Execution ................................. 72

E DALTON: neutron flux calculation 73
  E.1 Introduction .................................. 73
  E.2 Routines and execution .......................... 73
    E.2.1 makedalton.f90 ............................ 73
    E.2.2 dalton63 ................................ 73
    E.2.3 grabk.f90 .................................. 74
    E.2.4 Execution ................................ 74

F HEAT: energy and momentum 75
  F.1 Introduction .................................. 75
  F.2 Routines and execution .......................... 75
    F.2.1 HEAT ...................................... 75
    F.2.2 Execution ................................ 75

G Simulation results 76
  G.1 Log files ...................................... 76
  G.2 Simulation data for VisIt visualization .......... 76
    G.2.1 Execution ................................ 77
  G.3 Unformatted simulation data ...................... 77

III Appendix 78

H Directory structure 78

I Description of important input parameters 80
  I.1 MAININPUT ................................... 80
  I.2 Grid definition ................................ 82

J Nomenclature 85
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Technological development of nuclear reactors [5].</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>Schematic overview of the TMSR-NM reactor design.</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>Principle sketch of an intersection of the TMSR-NM [7].</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>Sketch of reactor geometry as applied in the computational model [7].</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Schematic overview of the reactor model.</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>Overview of the coupled calculation cycle of the MSR-CC code system.</td>
<td>42</td>
</tr>
<tr>
<td>7</td>
<td>Calculated $k_{\text{eff}}$ values for a set of reactor temperatures.</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>Reactor operation while establishing steady state.</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>Temperature distribution for steady state operation.</td>
<td>47</td>
</tr>
<tr>
<td>10</td>
<td>Fuel flow velocity distribution for steady state operation.</td>
<td>47</td>
</tr>
<tr>
<td>11</td>
<td>Neutron flux distribution for steady state operation.</td>
<td>48</td>
</tr>
<tr>
<td>12</td>
<td>Pressure distribution for steady state operation.</td>
<td>48</td>
</tr>
<tr>
<td>13</td>
<td>Turbulent Kinetic Energy (TKE) for steady state operation.</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>Turbulent viscosity for steady state operation.</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>Influence of the heat transfer coefficient on the simulation.</td>
<td>52</td>
</tr>
<tr>
<td>16</td>
<td>Influence of the fuel composition on the simulation.</td>
<td>52</td>
</tr>
<tr>
<td>17</td>
<td>Influence of grid resolution on reactor behavior.</td>
<td>54</td>
</tr>
<tr>
<td>18</td>
<td>Influence of the power scaling frequency on reactor behavior.</td>
<td>54</td>
</tr>
<tr>
<td>19</td>
<td>Delayed neutron source distribution without flow.</td>
<td>56</td>
</tr>
<tr>
<td>20</td>
<td>Delayed neutron source distribution with flow.</td>
<td>56</td>
</tr>
<tr>
<td>21</td>
<td>Pump failure scenario.</td>
<td>57</td>
</tr>
<tr>
<td>22</td>
<td>Heat exchanger failure scenario.</td>
<td>58</td>
</tr>
<tr>
<td>23</td>
<td>Overview of the coupled calculation cycle of the MSC-CC code system.</td>
<td>66</td>
</tr>
<tr>
<td>24</td>
<td>Schematic overview of the library structure.</td>
<td>67</td>
</tr>
<tr>
<td>25</td>
<td>Schematic overview of makescale.f90 routine.</td>
<td>68</td>
</tr>
<tr>
<td>26</td>
<td>Schematic overview of runbatch5 routine.</td>
<td>69</td>
</tr>
<tr>
<td>27</td>
<td>In- and output of the mix program.</td>
<td>70</td>
</tr>
<tr>
<td>28</td>
<td>In- and output of the wax_ice program.</td>
<td>71</td>
</tr>
<tr>
<td>29</td>
<td>In- and output of the batch5 program.</td>
<td>71</td>
</tr>
<tr>
<td>30</td>
<td>In- and output of the makedalton.f90 routine.</td>
<td>73</td>
</tr>
<tr>
<td>31</td>
<td>In- and output of the dalton63 program.</td>
<td>74</td>
</tr>
<tr>
<td>32</td>
<td>In- and output of the grabk routine.</td>
<td>74</td>
</tr>
<tr>
<td>33</td>
<td>In- and output of the HEAT program.</td>
<td>76</td>
</tr>
<tr>
<td>34</td>
<td>Screenshot of the main input file.</td>
<td>80</td>
</tr>
<tr>
<td>35</td>
<td>Horizontal grid construction.</td>
<td>82</td>
</tr>
<tr>
<td>36</td>
<td>Vertical grid construction.</td>
<td>83</td>
</tr>
<tr>
<td>37</td>
<td>The zone distribution of the reactor.</td>
<td>84</td>
</tr>
</tbody>
</table>
List of Tables

1. Reactor design specifications. .................................................. 16
2. Physical quantities involved in the neutron diffusion equation. .... 25
3. Physical quantities involved in k-ε turbulence model. ............... 32
4. Density and molar mass of the main components ....................... 39
5. Standard simulation parameters. .............................................. 43
6. Precursor decay constants and relative group yields. .................. 55
Part I

Thesis
1 Introduction

1.1 The Role of Nuclear Power in the Global Energy Supply and the Climate Change Problem

Even though there is no general consensus on when it will happen, mankind seems to agree on the idea that it will happen: somewhere in the future the world will run out of the fossil fuels that are currently being used to produce energy [1]. Our current standard in energy consumption causes us to consume enormous amounts of hydrocarbons, and it seems that this amount will even keep growing over the years. Besides this worrying development, the effect of burning hydrocarbons causes the concentration of carbon dioxide in the atmosphere to increase. Yet again, no consensus has been reached on how strong exactly this negatively effects the climate, but more and more scientific prove is suggesting it is significant [2]. For as far as both developments have not yet harmed the world already, alternative means for energy supply must be found in the near future to prevent disastrous effects on a social and/or an environmental level.

The need to reduce the worlds dependence on hydrocarbons has been recognized for some time already, so alternative methods have been established to produce energy from sustainable sources such as solar power, biofuels, wind power, or tidal power. These alternatives share the property that they do not pollute the environment (much) and their sources cannot run out. Unfortunately they also share the major difficulty that their energy production is very modest. This either results in the problem that enormous amounts of surface area are required to produce the required amount of energy, or the problem arises that the costs of producing the energy are so high that they are not commercially viable. Because of these problems, sustainable energy sources can not meet the required power production by far. Even though constantly progress is being made in developing these techniques, it seems likely that it will not be able to meet the energy demand when hydrocarbons are no longer available.

Realizing that the sustainable options for energy production are not going to be the solution in the short run, has brought back the discussion which role nuclear power could play in the supply of energy. Nuclear energy has always encountered much resistance, with its peak in the 1980’s and 1990’s, but its acceptance has slowly been increasing ever since. The resistance is based on some solid points. There is a risk of proliferation of techniques and materials that enables countries to produce nuclear weapons. Another point is the nuclear waste, which remains harmful for a time span up to tens of thousands of years [3], depending on the fuel cycle deployed. The last point is the possible dramatic consequences of an accident in a nuclear facility, in which the world unfortunately has some experience. A lot of effort is being put in solving the problems named above, but it seems that none of them
can be ruled out in the near future, and it is only possible to at least reduce their negative effects. It seems that the current situation does not allow the future energy mix to exclude nuclear power. Future progress in the development of both sustainable energy sources and nuclear energy will have to determine to what degree they can contribute to the world's daily energy needs. Meanwhile nuclear energy should not be considered a necessary evil, and its potential as a safe and a clean energy source should remain recognized.

1.2 Molten Salt Reactors

The first nuclear power plant has been established in the early 1950s, and its technology has been improving ever since. The U.S. Department of Energy defined four generations to indicate the major steps in the development of nuclear power plant technology, see figure 1. Each generation represents a higher level of technology of the nuclear reactors. The nuclear reactors that are currently in use or under construction are second- and third generation reactors (GEN-II and GEN-III). The next generation, GEN-IV, is a set of six classes of innovative nuclear reactor systems that are considered worth a thorough investigation for implementation in the future. Among these classes is the Molten Salt Reactor systems (MSR) [4].

A Molten Salt Reactor (MSR) is a type of nuclear reactor where the primary coolant is a molten salt, and in most designs the fuel is dissolved in the salt. Many variations of the reactor concept have been put forward, and a few prototypes have been built. Currently no MSRs are in use as nuclear power plants. The history of MSRs goes back to the 1960s where the concept was
used by the Oak Ridge National Laboratory during the Molten Salt Reactor Experiment (MSRE). The MSRE was a 7.4 MWth test reactor simulating the neutronic kernel of an inherently safe epithermal thorium breeder reactor [6]. Just as in the MSRE, the fuel used in molten salt reactors is often a fluoride salt. In many designs the nuclear fuel is dissolved in the salt coolant as uranium tetrafluoride, UF$_4$. MSR concepts are divided in thermal and fast reactors, both having a different setup. In a thermal MSR, the salt flows through a graphite matrix which acts as a moderator for the neutrons. In some modern thermal designs the fuel is dispersed in the graphite matrix. In the fast MSR concepts there is no graphite matrix that moderates the neutrons.

1.3 Non-Moderated Thorium Molten Salt Reactor

The MSR concept discussed in this thesis is the non-moderated Thorium Molten Salt Reactor (TMSR-NM). When Generation IV was established, France has expressed particular interest in the TMSR-NM concept. The Centre National de la Recherche Scientifique (CNRS) has initialized a research program on the Thorium fuel cycle and Molten Salt Reactors [7]. All of the reactor properties discussed in this report have been thoroughly evaluated in the CNRS research program, which enables validation of the results.

The TMSR-NM is designed to be well suited for operation with the thorium fuel cycle, which has several advantages. The advantage of the thorium fuel cycle is its capability to breed uranium ($^{233}$U). Research by Merle-Lucotte et al. [8] indicates that the MSR with a thorium fuel cycle and an epithermal neutron energy spectrum has promising features regarding safety coefficients, reprocessing requirements and breeding capabilities. One of the main advantages is the closed fuel cycle of the reactor. During operation the fuel salt is circulated through the core and the heat exchanger. An external loop is used for fuel reprocessing, which can be operated while the reactor is in operation. Therefore the reactor does not have to be shut down to load new fuel, which increases safety with respect to the fuel toxicity and proliferation. Another advantage, which is not specific for the TMSR-NM only but for most fast reactor, is the possibility to incinerate nuclear waste products through neutron induced fission. The previous MSR concepts were based on a thermal neutron spectrum, and therefore these reactors where not capable of incinerating all waste products that fast spectrum reactors could. By introducing the TMSR-NM, the capability of breeding fuel and incinerating waste products are combined.
1.3 Non-Moderated Thorium Molten Salt Reactor

1.3.1 Reactor design

Figure 2 shows a schematic overview of the TMSR-NM reactor design. The TMSR-NM has a relatively uncomplicated design, since it has no moderating graphite structure in the core. The core of the reactor is a cylinder, with the top and bottom of the cylinder covered by a neutron reflector. The inner core is surrounded from the sides by a fertile blanket. The molten fuel salt (or primary salt), enters the core through the salt injectors at the bottom. The salt flows up, and leaves the core through the salt collectors at the top. First the salt passes the helium bubbling component, and from there the salt flows down through the heat exchangers, back to the salt injectors. The primary loop is operated under ambient pressure, which is an
important safety feature. The top and bottom of the core are covered with axial neutron reflectors. The reactor vessel itself is covered with a reflector that absorbs 80% of the neutrons. The inner sides of the core are covered with a fertile blanket. Between the fertile blanket and the heat exchangers, a layer of boron carbide (B$_4$C) is applied as a neutron absorber. In total there are 32 of these channels through which the primary salt flows. Figure 2 also shows the in- and outlet of the secondary salt, which flows along the heat exchanger and from there to the reactor structure that transfers the heat to electricity. On top of the structure the in- and outlet of the reprocessing loop is shown. This loop is used to chemically reprocess the fuel salt while the reactor is in operation. Figure 3 shows an intersection of the reactor, which gives insight in the placing of the components and the directions of the fuel flow. The anticipated mean operating temperature is $T_c = 650 - 850^\circ$C, with a thermal power output of $P = 2.5 - 3$GW$_{th}$.

### 1.3.2 Computational model

Now a description will be given how the reactor is designed in the computational model. While in the previous subsection the original reactor design has been introduced, here it will be indicated which simplifications and assumptions are made to obtain a computational model that is suitable for simulation. The reactor is assumed to be a axisymmetric cylinder. Because of the azimuthal independence it is then possible to simplify the analysis to a 2D system, and describe the reactor in $(r,z)$ coordinates. Therefore it is assumed that the flow through the heat exchanger is not divided in 32 separate channels, but flows through one continuous channel. The sketch in figure 4 show how this geometry is defined. The pump is modeled by setting a fixed salt velocity of 2 m/s downwards, halfway the heat exchanger. The helium bubbling component and the layer of boron carbide are not implemented in the model. Only the flow of the primary salt is considered, going from the inner reactor core through the heat exchanger. The reprocessing loop is not taken into account in the model. The reflector covering the outer side of the reactor vessel is not implemented. Instead, the lower side of the bottom reflector is assumed to be in contact with a medium with a constant temperature of 100°C. The outer side of the breeder blanket is assumed to be in contact with air at a temperature of 25°C. The same holds for the upper side of the top reflector. Figures 4 shows how the simplified flow model is defined, and figure 5 is a sketch of the geometry applied in the reactor model, including its dimensions. Table 1 lists some of the specifications of the reactor model. The values are based on the original reactor design, and are also applied in the computational model.
1.3 Non-Moderated Thorium Molten Salt Reactor

Figure 3: Principle sketch of an intersection of the TMSR-NM [7]. The green arrows represent the fuel flow direction. The red channels flowing upwards represent the fuel reprocessing channel. These are not implemented in this reactor model.

Figure 4: Sketch of reactor geometry as applied in the computational model [7]. The figure indicates the simplified flow model, an upward flow in the inner core region and a downward flow in the heat exchanger channel.
1 INTRODUCTION

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core radius</td>
<td>1.25 m</td>
</tr>
<tr>
<td>Core height</td>
<td>2.6 m</td>
</tr>
<tr>
<td>Plenum thickness</td>
<td>2 x 0.45 m</td>
</tr>
<tr>
<td>Blanket thickness</td>
<td>0.4 m</td>
</tr>
<tr>
<td>Operating temperature</td>
<td>650 – 850°C</td>
</tr>
<tr>
<td>Power production</td>
<td>2.5-3 GW</td>
</tr>
<tr>
<td>Flow velocity through heat exchanger</td>
<td>2 m/s</td>
</tr>
</tbody>
</table>

Table 1: Reactor design specifications.

Figure 5: Schematic overview of the reactor model (radial cross section of cylinder). This figure represents the final design as implemented in the computational model. The dimensions are stated in [cm].
1.4 Objectives

**Goal of the project** To prepare research on the TMSR-NM reactor model at the research group Physics of Nuclear Reactors (PNR), a code system that is able to perform simulations of the neutronics and thermal hydraulics of this MSR concept was required. Separate in-house code systems are available that can calculate either the neutronics or the thermodynamics of a nuclear reactor model. Using these code systems, a simulation tool could be made that is able to combine the neutronics and thermodynamics of the reactor. The objective of this thesis project was therefore to develop a tool that is able to produce simulations of the non-moderated Thorium Molten Salt Reactor, with the following primary functionalities:

- Calculation of the salt flow velocity field by solving the momentum equations.
- Calculation of the energy balance, by taking into account:
  - Heat conduction, convection and radiation inside the reactor and to the surroundings
  - Heat production by nuclear fission
  - Heat extraction by the heat exchanger.
- Calculating the neutron flux of the reactor, and thus providing $k_{\text{eff}}$ and power production.

The simulation tool created in this project will be referred to as MSR-CC (Molten Salt Reactor-Coupled Calculations).

This project is carried out as a MSc graduation project at the research group Physics of Nuclear Reactors of the department Radiation, Radionuclides & Reactors of the Delft University of Technology. It is a contribution to the preparations of the EVOL project, titled 'Evaluation and Viability of Liquid Fuel Fast Reactor Systems' [10]. The project is coordinated by EURATOM, within the 'Seventh Framework Programme for Research and Technological Development (FP7)'.

1.5 Outline

This report consists of two parts. The first part is the thesis report that belongs to this graduation project. The second part is the user manual of MSR-CC. By splitting the report in this way, it will be more convenient for future users of MSR-CC to find information in it.

A theoretical background of the neutronics involved in the reactor model is provided in section 2. Section 3 concerns the processing of the neutron cross sections required for solving the multigroup diffusion equations. In section 4 the theoretical background of the thermal hydraulic properties of the reactor is introduced. A discussion about the material properties can
be found in section 5, and the coupled code system is discussed in section 6. In section 7 the results are presented. It discusses the functionalities of the simulation tool MSR-CC, shows results from test simulations, and finally the results of two reactor accident scenario’s are presented. The conclusions and recommendations can be found in sections 8 and 9 respectively. The section outline of the manual is provided in the manual itself, in section A on page 64.
2 Reactor neutronics

2.1 Introduction

In this section the most important element of nuclear reactors is discussed, the neutronics. In section 2.2 the multigroup diffusion theory is discussed. A short derivation of the multigroup diffusion theory is provided, starting from the energy-dependent diffusion theory. Also the concept of precursors and feedback mechanisms is introduced. Section 2.3 discusses a numerical model to analyze the influence of precursors on the neutron flux distribution.

2.2 Theory

2.2.1 Diffusion theory

In this subsection a short derivation will be provided of the neutron multigroup diffusion equations. The energy-dependent diffusion equation is chosen as the starting point

\[
\frac{1}{v} \frac{\partial \phi}{\partial t} - \nabla \cdot D(r, E) \nabla \phi + \Sigma_r(r, E) \phi(r, E, t) = \int_0^\infty dE' \Sigma_{s}(E' \rightarrow E) \phi(r, E', t) + S(r, E, t). \tag{2.1}
\]

Table 2 on page 25 provides a list of the physical quantities. This equation represents the neutron balance with all its loss and gain mechanisms. More information about this equation can be found in Duderstadt ([11],p.288). The neutron flux depends on time, the energy of the neutrons, and the spatial distribution. Before the multigroup diffusion equation is established, an expression for the neutron source term \( S(r, E, t) \) has to be found. This requires the introduction of the concept of precursors.

\( S(r, E, t) \) describes the mechanisms by which new neutrons can enter the neutron balance. Assuming there is no external source of neutrons, the only source of neutrons is by fission reactions. This is described by

\[
S(r, E, t) = \chi(E) \int_0^\infty dE' \nu(E') \Sigma_f(E') \phi(r, E', t). \tag{2.2}
\]

This equation describes the chance for a fission reaction to occur \( \Sigma_f \), the number of neutrons \( \nu \) that are created by the reaction and the possibility \( \chi \) that the created neutrons are born with an energy \( E \). This description works well for steady-state critical reactors, but to consider reactor time behavior the important concept of precursors must be included ([11],p.236).

Precursors In nuclear fission reactions, the major part of the neutrons will instantaneously be emitted in the form of free neutrons. These are called the prompt neutrons. However, almost 1% of the neutrons is released only after
a while, during the decay of precursor nuclei that are created during a fission reaction. These are the so-called delayed neutrons. This implies that with respect to equation (2.1), less prompt neutrons are formed, but also that another source term is introduced, by the neutron decay of the precursor nuclei. Since it would be very complicated to describe all the precursors with their individual decay rate, it is customary to divide them into six groups, each with their own characteristic yield and decay constant. The yield $\beta_i$ denotes the fraction of the total neutron production per fission in which a precursor from the group concerned is formed. The decay constant $\lambda_i$ indicates the rate by which neutron decay occurs in that specific group. Using the concentration of precursor nuclei $C_i$, the neutron source term by the decay of precursor nuclei can be described by $\lambda_i C_i$. Furthermore the prompt neutron source term must be corrected by a factor $(1 - \beta)$, with $\beta$ being the sum of the individual $\beta_i$ fractions. The total source term can now be written as

$$ S(r, E, t) = \chi_d(E) \int_0^\infty dE' (1 - \beta) \nu(E') \Sigma_f(E') \phi(r, E', t) + \sum_{i} \chi_d(E) \lambda_i C_i $$

(2.3)

where $\chi_d$ denotes the neutron energy spectrum by which neutrons are emitted. To adequately describe the precursor nuclei concentration in the reactor, a balance equation must be used to account for its loss and gain mechanisms. 

$$ \frac{\partial C_i}{\partial t} + \nabla u C_i = \int_0^\infty dE' \beta_i \nu(E') \Sigma_f(E') \phi(r, E', t) - \lambda_i C_i $$

(2.4)

On the left hand side the time rate of change of precursor nuclei is described, and the convection of precursors by media flowing with velocity $u$. The right hand side indicates the creation of precursor nuclei by fission, and the loss of precursors by neutron decay. Note that each precursor group has its own balance equation, so above six equations have been introduced.

### 2.2.2 Multigroup diffusion theory

The equations (2.1), (2.3) and (2.4) constitute a complete description of the energy-dependent neutron balance and the precursor concentrations. To solve the diffusion equation numerically, the continuous energy dependence of these equations is transformed to the multigroup diffusion equation. This implies that the neutron energy spectrum is discretized to a number of $G$ energy ranges. This creates a set of $G$ neutron balance equations, for which the multigroup fluxes $\phi_g$ must be solved. The multigroup fluxes are defined as the integral of the energy-dependent flux over their energy range, so they represent the total neutron flux within that energy range. Equation (2.1) shows that not only the neutron flux has to be discretized, but also the
2.2 Theory

quantities $D$, $\Sigma^t$, $\Sigma^s$, $\chi$, $\nu$ and $\Sigma^f$. These quantities are known as the group constants. A discussion about how group constants are processed will be postponed until the next section, since first the multigroup equations will be established. Transforming the diffusion equation with continuous energy to the multigroup diffusion equation involves taking the integral of the equation over the energy domain of each group. A detailed discussion about this work can be found in Duderstadt ([11],p.288). Here only the result will be introduced, the multigroup diffusion equation[12],

$$\frac{1}{v_g} \frac{\partial \Phi_g}{\partial t} - \nabla \cdot D_g \nabla \Phi_g + \Sigma^r_g \Phi_g = \sum_{g' \neq g} \Sigma^s_{g' \rightarrow g} \Phi_{g'} + \chi_p \sum_{g'} (1 - \beta) \nu \Sigma^f_{g'} \Phi_{g'} + \sum_i \lambda_i \chi_d C_i \quad (2.5)$$

and the multigroup precursor concentration equations

$$\frac{\partial C_i}{\partial t} = \sum_{g'} \beta_i \nu \Sigma^f_{g'} \Phi_{g'} - \lambda_i C_i - \nabla u C_i \quad (2.6)$$

To summarize the mechanisms in the equations above, below a brief description is given of each term:

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{v_g} \frac{\partial \Phi_g}{\partial t}$</td>
<td>Time rate of change of neutrons in group g</td>
</tr>
<tr>
<td>$\nabla \cdot D_g \nabla \Phi_g$</td>
<td>Change due to leakage</td>
</tr>
<tr>
<td>$\Sigma^r_g \Phi_g$</td>
<td>Neutrons leaving group g by a collision, either by absorption or out-group scattering</td>
</tr>
<tr>
<td>$\sum_{g' \neq g} \Sigma^s_{g' \rightarrow g} \Phi_{g'}$</td>
<td>Neutrons scattering into energy group g</td>
</tr>
<tr>
<td>$\chi_p \sum_{g'} G \nu \Sigma^f_{g'} \Phi_{g'}$</td>
<td>Source neutrons appearing in energy group g</td>
</tr>
<tr>
<td>$\sum_i \lambda_i \chi_d C_i$</td>
<td>Precursor decay neutrons appearing in energy group g</td>
</tr>
<tr>
<td>$\frac{\partial C_i}{\partial t}$</td>
<td>Time rate of change of the precursor concentration</td>
</tr>
<tr>
<td>$\sum_{g'} \beta_i \nu \Sigma^f_{g'} \Phi_{g'}$</td>
<td>Gain of precursors by fission reactions</td>
</tr>
<tr>
<td>$\lambda_i C_i$</td>
<td>Loss of precursors by decay</td>
</tr>
<tr>
<td>$\nabla u C_i$</td>
<td>Convection of precursors</td>
</tr>
</tbody>
</table>
2.2.3 Solving the multigroup diffusion equations: DALTON

At this point a set of equations has been established that enables calculation of the neutron flux profile and $k_{\text{eff}}$. Solving the multigroup diffusion equation numerically is done by finding a solution for the following matrix system

$$A\Phi_g = \frac{1}{k}F\Phi_g$$

(2.7)

where $A$ represents the diffusion terms of the diffusion equation, $F$ represents the fission matrix and $k$ is the multiplication factor. More information about how this numerical method is applied and solved can be found in Duderstadt ([11], p.216). It should be mentioned that although the shape of the neutron flux profile can be calculated, no information can be obtained about its magnitude. Therefore if $\Phi$ is a solution of the diffusion equations, then any multiple of $\Phi$ is also a solution ([11], p.206). In this project the numerical calculations are performed by DALTON. This code system requires a library of group constants as input. In section 3 it is explained how this library is processed. Furthermore it requires information about the reactor geometry, and various numerical solver properties can be adjusted. More information about this code system can be found in the manual of MSR-CC, which is part II of this document.

2.2.4 Power scaling

With respect to the thermal hydraulics calculations, which will be discussed in section 4, the quantity of interest is the power production in the reactor, since this quantity can directly be applied in the energy balance equation. DALTON however calculates the neutron flux profile and $k_{\text{eff}}$. Also it has been shown that the solution for the flux profile can be scaled by any constant, and thus the same holds for the power output. Therefore DALTON automatically scales the calculated flux profile to satisfy a power production that is provided in its input. For the simulation to represent a realistic reactor model, a relation between the power production and the $k_{\text{eff}}$ value of the reactor must be implemented. Therefore a short routine is created that linearly scales the power production with $k_{\text{eff}}$, every time the multigroup diffusion equation is calculated,

$$P_{\text{new}} = P_{\text{old}}k_{\text{eff}}.$$  

(2.8)

All simulations have been started with an initial thermal power production of $P_{\text{init}} = 2.0\text{GWth}$. This way a relation is established between $k_{\text{eff}}$ of the reactor and its power production.

2.2.5 Temperature feedback coefficient

A very important element in the control of nuclear reactors is the relation between changes in the reactor operating conditions and its effect on $k_{\text{eff}}$,
called feedback mechanisms. Often these feedback mechanisms are described as the relation between the reactor operating conditions and the reactivity. Reactor reactivity is defined as ([11], p. 221)

\[ \rho \equiv \frac{k - 1}{k}. \]  

(2.9)

One of the most important feedback mechanisms is the relation between temperature and reactivity. It has already been mentioned that the neutron cross sections depend on temperature, so a change in temperature would affect the cross sections and thus \( k_{\text{eff}} \) and \( \rho \). However this effect has not yet been quantified. The temperature feedback coefficient is given by ([11], p. 556)

\[ \alpha \equiv \frac{\partial \rho}{\partial T} = \frac{1}{k^2} \frac{dk}{dT}. \]  

(2.10)

This is an important parameter in nuclear reactor design, since it provides essential information about reactor behavior. For example, if the sign of \( \alpha \) is positive, that would imply that when the temperature of the reactor increases, the reactivity also increases. Therefore a negative temperature feedback coefficient is desired. Since DALTON can perform calculations of \( k_{\text{eff}} \) for any provided temperature, it is possible to calculate the \( \frac{dk}{dT} \) gradient and make an estimation of the temperature feedback coefficient.

2.3 Influence of precursors on the simulation

In this section the governing equations of the multigroup diffusion theory have been introduced. DALTON is able to numerically solve these equations, except for one part: the convection of the precursors. This implies that the term \( \nabla uC_i \) from equation (2.6) is not implemented in the code. Attempts have been made to use an alternative version of DALTON, which was created with the very purpose of being able to take fuel flow into account. Unfortunately also this version of DALTON appeared to be incompatible with the reactor system considered. Resolving this problem would involve too much work to fit within this project. To be able to at least visualize the consequence on the neutron flux of not considering the convection of precursors, a numerical model was created that calculates the precursor concentration of a highly simplified version of the reactor model. This model is described in the next paragraph.

Numerical model In 2.2.2 the precursor concentration equation (2.6) was introduced. This equation is used to obtain a numerical model of a simplified version of the reactor geometry that does include convection. The fuel flow field will be assumed to be one-dimensional, and independent of \( z \). The average vertical components of the steady state flow velocity provide the velocity values. Furthermore a steady state precursor concentration will
be assumed, $\partial C/\partial t = 0$. Taking these assumptions together the set of six precursor group equations becomes

$$v \frac{\partial}{\partial z} C_i = \beta_i \sum_g \nu_g \Sigma^f_g \Phi_g - \lambda_i C_i.$$  \hspace{1cm} (2.11)

From here the index $i$ of the precursor group will be omitted, in favor of the x- and y grid coordinates $i,j$. Then this equation can be transformed to a finite-volume numerical equation,

$$C_{i,j} = \frac{1}{V_{i,j}} \frac{\Delta z_j \beta \sum_g \nu_g \Sigma^f_g \Phi_{i,j,g} + v_i \frac{V_{i,j+1}}{V_{i,j}} C_{i,j-1}}{v_i + \lambda \Delta z_j}$$ \hspace{1cm} (2.12)

where $C_{i,j}$ represents the spatial distribution of the precursor concentration. This numerical model can be solved by using values for $\Phi_g$ and $\nu_g \Sigma^f_g$ from a steady state solution, and setting an arbitrary initial value for $C$ at the bottom of the core. Then this system can be calculated until the solution converges. Once the precursor concentration has been calculated for all six groups, the total contribution of precursors as a neutron source can be calculated by

$$\sum_i \lambda_i C_i.$$ \hspace{1cm} (2.13)

This quantity appears as a source of neutrons in the multigroup diffusion equation (2.5). The results of this numerical model are discussed in section 7.8.

2.4 Summary

To calculate the neutron flux profile and $k_{\text{eff}}$ of the reactor, the multigroup diffusion theory is numerically solved by the DALTON code system. The multigroup diffusion equation contains a number of quantities, known as the group constants, that should be processed before performing the calculation. A linear relation between the $k_{\text{eff}}$ and the resulting power production is introduced. Finally a numerical model is derived to visualize the effect of precursors on the neutron flux in a flowing fuel system. This analysis is performed since it was not possible to implement precursor flow in the model itself.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>neutron velocity</td>
</tr>
<tr>
<td>$C$</td>
<td>precursor concentration</td>
</tr>
<tr>
<td>$D$</td>
<td>neutron diffusion coefficient</td>
</tr>
<tr>
<td>$E$</td>
<td>energy of the neutron</td>
</tr>
<tr>
<td>$G$</td>
<td>total number of energy groups</td>
</tr>
<tr>
<td>$I$</td>
<td>total number of precursor groups</td>
</tr>
<tr>
<td>$S$</td>
<td>neutron source</td>
</tr>
<tr>
<td>$\beta$</td>
<td>delayed neutron fraction</td>
</tr>
<tr>
<td>$\Sigma_f$</td>
<td>fission cross section</td>
</tr>
<tr>
<td>$\Sigma_r$</td>
<td>removal cross section, i.e. by out-group scattering or absorption</td>
</tr>
<tr>
<td>$\Sigma_s$</td>
<td>scattering cross section</td>
</tr>
<tr>
<td>$\Sigma_t$</td>
<td>total cross section</td>
</tr>
<tr>
<td>$\chi$</td>
<td>neutron spectrum</td>
</tr>
<tr>
<td>$\phi$</td>
<td>energy dependent neutron flux</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>multigroup neutron flux</td>
</tr>
<tr>
<td>$\nu$</td>
<td>average number of neutrons created during one fission event</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>precursor decay constant</td>
</tr>
<tr>
<td>$s$</td>
<td>neutron energy group</td>
</tr>
<tr>
<td>$i$</td>
<td>precursor group</td>
</tr>
</tbody>
</table>

Table 2: Physical quantities involved in the neutron diffusion equation.
3 Neutron cross section processing

3.1 Introduction

In section 2 the multigroup diffusion theory has been introduced. This theory showed the importance of the energy-dependent behavior of the cross sections involved. This section concerns the process of creating the library of cross sections that is required to perform the numerical calculation of the multigroup diffusion equations. In 3.2 the concept of group constants is introduced, and a calculation method to find them is discussed. Section 3.3 introduces the SCALE-5 code system, which is used to numerically find the group constants. Section 3.4 describes the construction of the cross section work library, and 3.5 discusses the MIX code system, which provides an interpolates set of cross section corresponding to the reactor temperature.

3.2 Group constants

In section 2.2.2 it has been shown that transforming the energy-dependent diffusion equation to the multigroup diffusion equation involves a thorough evaluation of the physical quantities that depend on energy. For example, the energy-dependent fission cross section $\Sigma_f(E)$ must be replaced by the constant value $\Sigma_g$ in energy group $g$ of the multigroup diffusion equation. These constants in the multigroup diffusion equation are called the group constants, as they are constant within the energy group. Multigroup diffusion theory can be an adequate approach to solve the neutron balance, provided that accurate group constants are applied. This section will describe how these group constants are obtained. To do so, the example of the fission cross section will be continued here. The formal description of the neutron flux in energy group $g$ is given by ([11],p.289)

$$\Phi_g(r,t) \equiv \int_{E_g}^{E_{g-1}} dE \phi(r,E,t) \quad (3.1)$$

and the fission cross section group constant is given by

$$v_g \Sigma_f^g \equiv \frac{1}{\Phi_g} \int_{E_{g'}}^{E_{g'-1}} dE' \nu(E') \Sigma_f^g(E') \phi(r,E',t). \quad (3.2)$$

Even though at this moment an explicit expression for $\Sigma_f^g$ is obtained, it is apparent that in order to be able to calculate this group constant, the flux $\phi(r,E',t)$ must be known. But the flux is exactly the quantity one ultimately wants to solve. The next paragraph describes how this problem is solved.
Calculation method for group constants  The strategy to obtain the group constants is to guess or approximate the intragroup fluxes $\phi(r, E, t) \approx \phi_{\text{approx}}(r, E, t)$ and use these fluxes to find the group constant expressions. The most common method to generate group constants is by performing two multigroup calculations:

1. First the spatial and time dependence of the flux is ignored (or simplified), and a very finely structured multigroup calculation is performed to calculate the intragroup fluxes. The group constants used for the fine group calculation are tabulated cross section data averaged over the fine groups.
2. Then the calculated intragroup fluxes are used to calculate the group constants for the desired multigroup distribution.

In the next subsection it will be explained how these group constants are processed in this project.

3.3 Processing group constants: SCALE

In the previous section the multigroup diffusion theory has been discussed, which is numerically solved with the DALTON code. Furthermore the concept and importance of group constants in the multigroup diffusion equations has been introduced. DALTON does not calculate the group constants, but requires them as input to be able to calculate the neutron flux. To generate the group constants, the SCALE-5 code system is used. The SCALE code system performs the calculation of the group constants by the procedure just described.

Energy groups  First, a choice must be made for the energy ranges used. In this model, three energy groups have been used, with the following domains.

- $E_1 = [2.0 \cdot 10^7 - 9.5 \cdot 10^2 \text{ eV}]$
- $E_2 = [9.5 \cdot 10^2 - 6.25 \cdot 10^{-1} \text{ eV}]$
- $E_3 = [6.25 \cdot 10^{-1} - 1.0 \cdot 10^{-5} \text{ eV}]$

These groups represent roughly the energy of neutrons created by fission in $E_1$, the energy domain in which the neutrons are moderated in $E_2$, and thermal to cold neutrons in $E_3$.

Reactor material  In the diffusion theory, the group constants are properties of the material present in the domain of which the flux distribution is calculated. Therefore, SCALE requires input about the composition of the materials, and if multiple materials are used, their distribution in the reactor. More information about the composition of the materials can be found in section 5.
Reactor geometry  Since an approximated flux distribution must be calculated, information about the geometry of the reactor is required as input for SCALE. In this model, the group constants for each material have been calculated separately, as if they were infinite media. Considering the goal of this project, which is to establish a first version of a coupled code system, this approximation is justified. For more accurate processing of the group constants, reactor geometry can be included in the model.

Reactor temperature  Finally, also the temperature of the reactor must be taken into account, since the cross section spectra depend on the thermal motion of the nuclei. This plays a role for example in the concept of doppler broadening, where the neutron absorbance by resonances increases with increasing temperature ([11], p.49). Therefore, also the temperature of the materials must be provided as input for SCALE.

3.4 Cross section work library

In the previous subsection it has been described how the group constants can be processed with the SCALE code system. One of the input arguments for SCALE is the reactor temperature. Since in this project the thermal hydraulics of a nuclear reactor is simulated, the reactor temperature will vary during the simulation, which would require a new calculation of the group constants. This would be a very time consuming process. Instead, before starting with the simulation, a library of group constants will be processed for a discrete set of temperatures. This library will be referred to as the cross section work library. To establish this, the temperature limits must be chosen, and a distribution of temperatures within these limits. In this project the following temperature set is used

\[ 300/600/900/1200/1500/1800/2100\text{K} \]

The only step that remains is to use this cross section work library to find group constants for reactor temperatures that lie in between this set of temperatures. This will be discussed in the next subsection.

3.5 Cross section interpolation: MIX

In the previous subsection it has been described how a cross section work library is established with group constants for a set of temperatures. During the simulation it should be possible to generate group constants for the current reactor temperature. In other words, the group constants should be updated during the simulation. This functionality is provided in the in-house code system MIX. By providing the reactor temperature as input for MIX, it will calculate the coefficients that are required to make an interpolation between two sets of groups constants from the library. From
these coefficients it creates an input file for a module of the SCALE code system, to ultimately provide a set of group constants corresponding to that temperature. So MIX does not perform the actual calculation of the group constants, but it does write instructions for the SCALE code system, which performs the actual interpolation. Having introduced the MIX code, it is now possible to generate the group constants for each material in the reactor, at any given temperature.

3.6 Summary

In this section the concept of group constants, arising from the multigroup diffusion theory, has been discussed. A general method for calculating the group constants is introduced. The numerical processing of the group constants is performed by the SCALE code system. SCALE creates a cross-section work library which contains group constants data for a set of temperatures. This work library is created before the simulation is started. During the simulation, the MIX code system repeatedly performs an interpolation between the library data to create a set of group constants that belongs to the current reactor temperature.
4 Thermal hydraulics of TMSR-NM: HEAT

4.1 Introduction

In the previous sections the neutronics of the reactor model has been described. This section concerns the thermal hydraulic properties of the reactor model. One part of these calculations is solving the energy balance, to find the reactor temperature. An important parameter in this calculation is the power production, which is provided by DALTON. Furthermore the fuel flow field is calculated. Also a turbulence model is implemented in the code system, the $k - \epsilon$ model. This model calculates the turbulent kinetic energy of the flow and the dissipation of this energy. From these results the turbulent viscosity is calculated, which is an important parameter for the momentum equations. Also the pressure distribution is calculated. The main effort in this project has been to adjust the HEAT code system as used in [14] to be compatible with the physical model for this reactor. In 4.2 the theoretical background of the HEAT code system is provided.

4.2 Theory

4.2.1 Flow field

**Reynolds number estimation** The dimensionless Reynolds number indicates whether flow is in the laminar or turbulent regime. So depending on the Reynolds number of the flow considered in this project, it can be verified whether a turbulence model is justified. The Reynolds number is given by ([15],p.81)

\[ \text{Re} = \frac{\rho v_x D}{\mu_l}. \]  

(4.1)

The following typical values have been used: a fuel flow velocity of $v = 2 \text{ m.s}^{-1}$, a laminar viscosity of $\mu_l = 4 \cdot 10^{-3} \text{ Pa.s}$ and a density of $\rho = 3000 \text{ kg.m}^{-3}$. Two domains have been considered, the inner core region with a diameter of $D_{\text{core}} = 2.5 \text{ m}$, and the heat exchanger channel with a diameter of $D_{\text{channel}} = 0.25 \text{ m}$. From this it was found that in the heat exchanger channel $\text{Re} \sim 10^5$, and in the reactor core $\text{Re} \sim 10^6$. Both Reynolds numbers are in the turbulent regime ([15],p.82), so a turbulence model is justified for this reactor design.

**Continuity equation** The mass conservation law states that the rate of change of mass in a volume equals the rate of mass production in that volume [16]. In vector form this law reads

\[ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0. \]  

(4.2)
The flow in this reactor is considered to be incompressible, so \( \rho(\mathbf{r}, t) = \rho \). Then equation (4.2) becomes

\[
\nabla \cdot \mathbf{v} = 0 \tag{4.3}
\]

where \( u \) and \( v \) are the velocities in the horizontal and vertical direction respectively, and the subscripts \( x \) and \( y \) indicate partial derivatives.

**Flow equations**  The salt can be considered to be an incompressible, viscous fluid and hence can be described by the Navier Stokes equations [15]. In vector form it reads

\[
\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p - \nabla \cdot \mathbf{T} + \mathbf{f} \tag{4.4}
\]

The terms on the left hand side describe the unsteady acceleration and the convective acceleration respectively. The right hand side describes the divergence of stress and other body forces. The only body force \( \mathbf{f} \) on the fuel salt is gravity, so \( \mathbf{f} = \rho g \). For an incompressible, newtonian fluid the divergence of stress can be written as [17]

\[
\mathbf{T} = 2\mu_{\text{eff}}(\mathbf{E} - \frac{1}{3}\Delta \mathbb{I}) \tag{4.5}
\]

with the rate-of-strain tensor

\[
\mathbf{E} = \frac{1}{2}(\nabla \mathbf{v}) + \frac{1}{2}(\nabla \mathbf{v})^T \tag{4.6}
\]

the rate of expansion of the flow

\[
\Delta = \nabla \cdot \mathbf{v} \tag{4.7}
\]

and \( \mathbb{I} \) the unit matrix. The effective dynamic viscosity is the sum of the laminar and the turbulent dynamic viscosity

\[
\mu_{\text{eff}} = \mu_l + \mu_t. \tag{4.8}
\]

**4.2.2 Turbulence model \((k - \epsilon)\)**

The \( k - \epsilon \) model is often used to describe turbulent flow. The model is described by two equations, one for the turbulent kinetic energy \( k \), and one for the dissipation rate of turbulent kinetic energy, \( \epsilon \). The turbulent kinetic energy (TKE) is the mean kinetic energy per unit mass associated with eddies in turbulent flow. The turbulent kinetic energy is dissipated because of the work done by the fluctuating viscous stresses in resisting deformation of the fluid material by the fluctuating strain rates [18]. The two equations
are given by [19]

\[ \rho \left( \frac{\partial k}{\partial t} + \nabla \cdot (v k) \right) = \nabla \cdot \left( \left[ \mu_l + \frac{\mu_t}{\sigma_k} \right] \nabla k \right) - \rho \epsilon + G_k \]  
(4.9)

\[ \rho \left( \frac{\partial \epsilon}{\partial t} + \nabla \cdot (v \epsilon) \right) = \nabla \cdot \left( \left[ \mu_l + \frac{\mu_t}{\sigma_\epsilon} \right] \nabla \epsilon \right) + C_1 \epsilon G_k - C_2 \epsilon^2 \]  
(4.10)

In these equations the production of turbulent kinetic energy by buoyancy is omitted. $G_k$ is a production term due to mean shear. The constants used in this model are listed in Table 3.

<table>
<thead>
<tr>
<th>$k$</th>
<th>turbulent kinetic energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon$</td>
<td>dissipation of turbulent kinetic energy</td>
</tr>
<tr>
<td>$\rho$</td>
<td>density of the salt</td>
</tr>
<tr>
<td>$v$</td>
<td>salt velocity</td>
</tr>
<tr>
<td>$\mu_l$</td>
<td>laminar viscosity</td>
</tr>
<tr>
<td>$\mu_t$</td>
<td>turbulent viscosity</td>
</tr>
<tr>
<td>$\mu_{\text{eff}}$</td>
<td>effective viscosity</td>
</tr>
<tr>
<td>$\sigma_k = 1.0$</td>
<td>turbulence model constants</td>
</tr>
<tr>
<td>$\sigma_\epsilon = 1.3$</td>
<td></td>
</tr>
<tr>
<td>$C_{1\epsilon} = 1.44$</td>
<td></td>
</tr>
<tr>
<td>$C_{2\epsilon} = 1.92$</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Physical quantities involved in $k$-$\epsilon$ turbulence model. The constants named in the table are obtained from [19].

**Production due to shear stress**  
$G_k$ represents the production of turbulent kinetic energy. It is is given by

\[ G_k = \mu_t S^2 \]  
(4.11)

with $S$ the modulus of the mean rate-of-strain tensor,

\[ S \equiv \sqrt{2\overline{EE}} \]  
(4.12)

**Turbulent viscosity**  
Once $k$ and $\epsilon$ are calculated, the resulting turbulent viscosity is calculated from

\[ \mu_t = \rho C_\nu \frac{k^2}{\epsilon} \]  
(4.13)

**Law of the wall**  
Turbulent flow in the vicinity of a solid wall is described by the 'law of the wall'. This law states that states that the average velocity of a turbulent flow at a certain point is proportional to the logarithm of the
4.2 Theory

distance from that point to the wall. This proportionality is described by [20]

\[ u^+ = \frac{1}{\kappa} \ln(y^+) \]  

(4.14)

where \( u^+ \) is the dimensionless velocity, given by

\[ u^+ = \frac{u}{u_\tau} \]  

(4.15)

with \( u \) the velocity parallel to the wall, and \( u_\tau \) the friction velocity. \( \kappa \) is the von Karman constant, given by \( \kappa = 0.41 \). The dimensionless quantity \( y^+ \) denotes the wall coordinate, and is given by

\[ y^+ = y \frac{u_\tau \rho}{\mu_l} \]  

(4.16)

where \( y \) is the distance to the wall. In the k-\( \epsilon \) model the friction velocity is given by ([21],p.136)

\[ u_\tau = k^{1/2} C_\mu^{1/4}. \]  

(4.17)

A proper choice must be made for the region of this viscous sublayer for which the law of the wall applies, since the law is not valid for flow that is too close to the wall. The criteria is set by the wall coordinate, which states that if \( y^+ > 11 \), the law of the wall should be chosen [20].

4.2.3 Heat balance

The heat production and its distribution over the reactor is the main variable of interest when considering power (efficiency) and safety. The heat balance will be analyzed in three domains: the reactor core, the bottom and top reflectors, and the fertile blanket. The starting point is the energy equation [15]

\[ \rho c_p \frac{DT}{Dt} = -\nabla \cdot \mathbf{q}^{\text{total}} + S + L \]  

(4.18)

or, when writing out the total derivative

\[ \rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T \right) = -\nabla \cdot \mathbf{q}^{\text{total}} + S + L. \]  

(4.19)

Here \( c_p \) is the specific heat capacity at constant pressure, \( \mathbf{q}^{\text{total}} \) the sum of all contributions to the heat flux density, and \( S \) and \( L \) are the source and loss terms per unit volume, respectively. From here it will be discussed how the terms in equation (4.19) are defined for each of the domains. First a description is given of the heat balance for the inside of the domains, and secondly it is discussed how the energy balance is defined at the domain interfaces.
**Reactor core**  The heat flux density within the reactor core consist only of heat conductivity, which is given by

\[ q^{cd} = -\lambda_{\text{fuel}} \nabla T \]  

(4.20)

and appears as \( q^{\text{total}} = q^{cd} \) in equation (4.19). The energy source can simply be stated as \( S = P \), which is the fission power density obtained directly from calculations by DALTON. The only contribution to the loss term \( L \) is by the heat exchanger. The energy extracted from the system by the heat exchanger is set to be

\[ L = h(T - T_0) \]  

(4.21)

with \( h \) the heat transfer coefficient, and \( T_0 = 298 \text{K} \). Note that the loss term by the heat exchanger is only applied in the channel, and not in the inner core region.

**Reflector**  In the reflector domain there is no flow field present, so in equation (4.19)

\[ \mathbf{v} \cdot \nabla T = 0. \]  

(4.22)

There is no source of loss term present, and the heat flux by conductivity is given by

\[ q^{cd} = -\lambda_{\text{reflector}} \nabla T \]  

(4.23)

with \( q^{\text{total}} = q^{cd} \).

**Fertile blanket**  The heat balance for the fertile blanket region is almost the same as for the reflector. No flow field and no loss term are present. The only difference is that the source term is \( S = P \) instead of zero, since the blanket contains fertile material. And since the conductivity of the fertile blanket and the reflector is not equal, here the conductivity heat flux is given by

\[ q^{cd} = -\lambda_{\text{blanket}} \nabla T \]  

(4.24)

with \( q^{\text{total}} = q^{cd} \).

**Interfaces**  At this point the energy equation is given for the inside of each of the domains. At the interfaces of these domains other mechanisms might need to be applied. These mechanism will be described below.

**Radiation**  The top of the reactor, the outer side of the reflector, is considered to be surrounded by air. The same holds for the outer side of the fertile blanket at the side. Therefore at these interfaces, heat is transferred by radiation. The heat flux density for radiation is given by

\[ q^R \bigg|_{\text{sides}} = 4\epsilon \sigma T^4. \]  

(4.25)
with $\epsilon = 0.8$ the emissivity and the Stefan Boltzmann constant, $\sigma_b = 5.67 \times 10^{-8} \text{ W.m}^{-2}\text{.K}^{-4}$.

**Free convection** In the same region where radiation occurs, heat can also transferred by free convection. The heat transfer for this process is defined as

$$h = \frac{\text{Nu}\lambda_{\text{air}}}{D}$$

with $D$ a characteristic length and Nu the dimensionless Nusselt number. The expression for the Nusselt number depends on the geometry of the surface that is considered, and can be found in Mudde ([15], p.152). The heat flux density for free convection can thus be written as

$$q_{\text{convection}} \bigg|_{\text{sides}} = \frac{\text{Nu}\lambda_{\text{air}}}{D} (T - T_0)$$

with $T_0 = 298\text{K}$.

### 4.3 Summary

In this section the theoretical background of the thermal hydraulics of the reactor model has been introduced. The Navier Stokes equations have been applied on a viscous, incompressible fluid to obtain the fuel flow equations. A Reynolds number calculation indicated that the flow field considered is in the turbulent regime, with $\text{Re} \sim 10^6$. Therefore the $k - \epsilon$ turbulence model is applied, of which the governing equations have been presented. Finally an overview has been given of the heat balance in the reactor. All physical quantities described in this section are numerically calculated by the HEAT code.
5 Reactor material properties

5.1 Introduction

In the previous sections it has been shown that many of the calculations involve material properties. A very important example is the material composition of the fuel salt, since the amount of fissile material in the fuel salt strongly affects the neutron cross sections. Material properties are also required to calculate the flow field and the energy balance of the reactor. In 5.2 the materials used in this reactor model are described. In 5.3 the physical properties of these materials are discussed.

5.2 Reactor materials

Three different materials can be distinguished in the reactor model: the breeder, the reflector and the fuel salt. Figure 5 on page 16 shows where these materials are located. The composition of these materials is based on materials used in the TMSR-NM concept as described by Merle-Lucotte et al. [7].

**Reflector**  The axial reflectors consist of NiWCr with respective molar fractions of 7%, 6% and 87%. This reflector material is chosen for its capability to avoid thermalization of the neutrons [7].

**Fertile blanket**  The fertile blanket is the binary fluoride salt LiF-ThF$_4$ with a molar fraction of 28% of ThF$_4$.

**Fuel salt**  Two types of material have proven to be suitable to be used as fuel for the TMSR-NM, fuel based on uranium and fuel based on plutonium together with minor actinides. Both fuels have their own advantages and will have different effects on the reactor behavior, but since the goal of this project is not directly related to the used materials, only the uranium based fuel is evaluated here. Below a short description of both fuels is given.

**Uranium**  The uranium based fuel of the TMSR-NM is the fluoride salt LiF-ThF$_4$-UF$_4$. In this fuel, thorium and uranium are the fertile and fissile materials respectively. The molar fraction of these two elements in the fuel may vary from 7.5% to 27.5%, which will result in an epithermal (for low %) to a fast neutron spectrum (for high %). The inventory of uranium and thorium for a molar fraction of 27.5% is m(\textsuperscript{233}U)=6180kg and m(\textsuperscript{233}Th)=52190kg.
Transuranic Elements The reactor fuel can also be a fluoride salt composed of thorium together with a mix of transuranic elements (TRU). The chemical composition is LiF-ThF$_4$-(Pu-MA)F$_3$, where MA stands for Minor Actinides. Conveniently, the TRU mix of used fuel of water moderated reactors can be used as this kind of fuel. The concentration of heavy nuclei in this salt also ranges from 7.5% to 27.5%.

5.3 Material properties

It is necessary to know certain material properties to calculate the energy and momentum balance of the reactor. Since it was not always possible to find exact data of material properties, approximations have been made which are well suited for the simulations in this stage and will cause only minor deviations. These approximations are indicated throughout the section.

5.3.1 Number density

The SCALE library contains information about cross sections per nuclide. DALTON however requires neutron cross section data per material used, and not for individual nuclides. Therefore SCALE is used to produce cross section data for materials as a composition. SCALE requires the number densities of the nuclides that constitute the material as input. In the next paragraph it is explained how the number densities can be calculated when the molar fractions of the nuclides in a material are known.

Calculating number density To calculate the number density $N_i$ of a nuclide within a material, its molar mass $M_i$, molar fraction $x_i$, and density $\rho_i$ must be known. First the molar volume of the material as a whole is calculated by \[ V = \sum_{i \text{ nuclides}} x_i \frac{M_i}{\rho_i}. \] (5.1)

Then the number densities of the individual nuclides can be calculated by \[ N_i = x_i N_A \frac{1}{V}. \] (5.2)

5.3.2 Density

The temperature dependent densities of the components of which the core and the breeder salt are constituted, is shown in Table 4. These densities are used when creating the cross-section library. In HEAT the salt is assumed incompressible, so then a constant density is used. The densities of the fuel salt and the fertile blanket are almost equal, and are taken to be $\rho =$
3290 kg.m$^{-3}$, which corresponds to a temperature around 700K [24]. The density of the reflector is $\rho = 8860$ kg.m$^{-3}$ [25].

5.3.3 Heat capacity

According to Khoklov [26] the heat capacity of LiF-ThF$_4$(78-22) is $c_p = 1.205 \cdot 10^3$ J.kg$^{-1}$.K$^{-1}$. Though the concentration in the fertile blanket material is slightly different, this value will be used as a sufficient approximation. Also from Khoklov the value $c_p = 1.565 \cdot 10^3$ J.kg$^{-1}$.K$^{-1}$ is used as the heat capacity of the fuel salt, despite the fact that this value belongs to a LiF-BeF$_2$-ThF$_4$ mixture. To estimate the heat capacity of the reflector material, the commercially available Hastelloy N [25] has been used, which has a similar composition. This material has a heat capacity of $c_p = 5.00 \cdot 10^2$ J.kg$^{-1}$.K$^{-1}$

5.3.4 Viscosity

The MOST review of the chemical aspects of reactor fuels [27] gives a laminar viscosity of $\mu_{lam} = 4 \cdot 10^{-3}$ Pa.s for a LiF-ThF$_4$ mixture. This value will be used for the fuel salt. The turbulent viscosity will be discussed at the governing equations chapter.

5.3.5 Thermal conductivity

Experimental values for the thermal conductivity for the fuel salt and the fertile blanket were not available, so instead an empirical relation is applied that yields reasonable values for most fluoride salts [26]

$$\lambda = -0.34 + 0.5 \cdot 10^{-3} T[K] + \frac{32.0}{M},$$

with M the molar mass of the salt and T the temperature of the material. This relation will be used both for the fertile blanket and the fuel salt. The thermal conductivity of the reflector is $\lambda = 20$ W.m$^{-1}$.K$^{-1}$ [25], temperature dependence is not taken into account. In turbulent flow, the total conductivity is given by

$$\lambda_{tot} = \lambda + \lambda_t$$

(5.3)

with the turbulent conductivity given by [28]

$$\lambda_t = \frac{c_p \mu_t}{Pr_t}$$

(5.4)

with $Pr_t$, the dimensionless Prandtl number, which approximates the ratio of momentum diffusivity and thermal diffusivity.
Table 4: Density and molar mass of the main components of the fuel salt and the fertile blanket [25], [27]. These material properties are required when calculating number densities, which is used as input for SCALE to generate the cross section library.

5.4 Summary

This section contains information about the materials used in this reactor design. Three materials are applied, the fuel salt, the fertile blanket and the reflector. It is discussed what the composition is of each of the materials, and on what grounds they are chosen. Furthermore all material properties are introduced that are required the perform the calculations that have been discussed in the previous sections.
6 Coupling of the code systems

6.1 Introduction

In the previous sections the calculation methods for many different physical quantities have been introduced. In these sections it was already briefly mentioned which code systems perform the calculations. In this section the code systems will be described in more detail and it will be explained how they are coupled. Section 6.2 summarizes which code systems are used and what they will calculate. Section 6.3 describes how these codes are connected to obtain a coupled code system.

6.2 Code systems

The code systems used in this project were already available at the PNR research group, or had to be created. The concept of a coupled code system is that the programs depend on each other’s input, and that the calculations are repeated in a cycle. This has been the main effort of this project: adjusting the involved code systems in such a way that they perform their calculations on the same reactor model, and to make communication between the code systems possible. Below it is briefly described which code systems are used, what they calculate and in which order they operate.

SCALE In section 3.3 it has been described how the code system SCALE \citep{29} is able to generate a library of group constants, the constants required to solve the multigroup diffusion equations. The library consists of a set of group constants for each material used in the model, for a discrete range of temperatures.

MIX The MIX code has been introduced in section 3.5. It is used during the simulation to calculate the set of group constants that corresponds to the current reactor temperature, and thus has to be updated within the cycle.

DALTON Section 2 provided an overview of the neutronics involved in this reactor model. The in-house code system DALTON\textsuperscript{1} is capable of calculating the multigroup neutron diffusion equations to find the neutron flux distribution and the $k_{\text{eff}}$ value. It uses the set of group constants generated by the MIX code as input. From the neutron flux distribution DALTON calculates the power production density.

\textsuperscript{1}Developed by D. Lathouwers, d.lathouwers@tudelft.nl.
6.3 Coupled code system

HEAT The thermal hydraulic aspects of the reactor model have been introduced in section 4. The most important quantities are the fuel flow field and the reactor temperature. The thermal hydraulic quantities of the reactor are calculated by the in-house code system HEAT\(^2\).

Power scaling The power scaling is a short routine that establishes a relation between the \(k_{\text{eff}}\) value and the evolution of the power production in the routine. The concept of power scaling is discussed in section 2.2.4.

6.3 Coupled code system

At this point enough information has been gathered to construct the final coupled calculation system which enables simulation of the TMSR-NM. The procedure is started by SCALE generating a cross section work library. Information about the material composition and reactor geometry must be provided as input. Then the first step within the coupled calculation cycle is performed, the interpolation of the group constants by the MIX code. MIX performs an interpolation of the group constants to obtain the values corresponding to the current reactor temperature. Now DALTON can use these group constants as input to calculate the power production and the \(k_{\text{eff}}\) value. Before DALTON is started, the required power output is corrected for the previous value of \(k_{\text{eff}}\) by the power scaling. Then HEAT is able to calculate the thermal hydraulic properties of the reactor, using the power production as input for the heat balance. After this calculation, a new reactor temperature has been established, so MIX should update the group constant values for the new temperature. From this point on the coupled calculation cycle can be started again. For future reference, this simulation system is called MSR-CC (MSR - Coupled Calculations). Figure 6 shows how the routines just described communicate with each other. It also indicates what the input and the output is for every part of the cycle.

6.4 Summary

The code systems that have been described in the previous sections have been summarized. The construction of the coupled calculation system is described, by indicating the order in which the code systems are executed, and what information they communicate. This system of coupled calculations will be referred to as MSR-CC for future reference. At this point a complete description is provided about the coupled code system of the TMSR-NM. The theoretical background is provided, and the approach of solving this set of equations numerically has been described. The next section will present the results of the simulations.

\(^2\)Developed by D. Lathouwers, d.lathouwers@tudelft.nl.
Figure 6: Overview of the coupled calculation cycle of the MSC-CC code system. The simulation consists of two parts, first the initialization phase, and then the coupled calculation cycle. Within the boxes it is indicated which code system performs the calculation. In between the boxes it is indicated what physical quantity is communicated.
7 Results

7.1 Introduction

In this section the results of the simulations performed by MSR-CC are presented. The process of constructing and programming the simulation tool MSR-CC is not discussed here. In this section only results are discussed that are acquired by the final version (at least within this project) of MSR-CC. First the calculation of the temperature feedback coefficient is presented in section 7.2. To introduce the features of the simulation tool, in section 7.3 the results are discussed from a single simulation that establishes steady state operation. Section 7.4 discusses the influence of adjusting the amount of heat removal by the heat exchanger. Several fuel compositions have been simulated in section 7.5. In section 7.6 the importance of the grid resolution of the numerical model has been evaluated. The consequences of changing the frequency of power scaling is discussed in 7.7. A brief analysis of the effect of precursor flow on the neutron flux is performed in 7.8. In section 7.9 the results of two transient scenario’s are briefly discussed: a heat exchanger failure and a pump crash.

Simulation parameters Throughout this section, simulation results are presented. Unless stated otherwise, the following parameters have been used in the simulations:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cycles</td>
<td>1000</td>
</tr>
<tr>
<td>Number of HEAT timesteps per cycle</td>
<td>1</td>
</tr>
<tr>
<td>Time step size in HEAT</td>
<td>0.5 s</td>
</tr>
<tr>
<td>Total grid resolution (ni:nj)</td>
<td>25:40</td>
</tr>
<tr>
<td>Initial power production</td>
<td>2000 MW</td>
</tr>
<tr>
<td>Heat transfer coefficient</td>
<td>(900 \times 10^4 , \text{W.m}^{-3}.\text{K}^{-1})</td>
</tr>
<tr>
<td>Initial core temperature</td>
<td>700°C</td>
</tr>
<tr>
<td>Initial reflector / outer fertile blanket temp</td>
<td>100°C</td>
</tr>
<tr>
<td>Initial inner fertile blanket temp</td>
<td>500°C</td>
</tr>
<tr>
<td>Molar fraction of heavy nuclei in the fuel</td>
<td>29.5%</td>
</tr>
</tbody>
</table>

Table 5: Standard simulation parameters. Unless stated otherwise, these parameters have been used to perform the simulation.

7.2 Temperature feedback coefficient

In section 2.2.5 the concept of feedback has been discussed. Using the code systems DALTON and MIX it is possible to calculate \(k_{\text{eff}}\) for a set of reactor temperatures, without having to run a full simulation. From these results
it is then possible to obtain the gradient $dk/dT$ by linear fitting. Figure 7 shows the result of these calculations. The linear fit of the data returned the result $dk/dT = -6.0783 \times 10^5 \text{ K}^{-1}$. So the temperature feedback coefficient is approximately $\alpha \approx -6 \text{ pcm}/\text{K}$. This is comparable with the results obtained by Merle-Lucotte et al. [13], who found $\alpha \approx -5 \text{ pcm}/\text{K}$ in similar conditions. This rather large negative relation between $k_{\text{eff}}$ and the reactor temperature is a great advantage in reactor control, since a sudden increase in temperature will be followed by a swift decrease in reactivity.

![Graph showing calculated $k_{\text{eff}}$ values for a set of reactor temperatures. The red line indicates a linear fit of the data, the gradient of this fit provides the value of $dk/dT$.](image)

**Figure 7**: Calculated $k_{\text{eff}}$ values for a set of reactor temperatures. The red line indicates a linear fit of the data, the gradient of this fit provides the value of $dk/dT$.

### 7.3 Steady state operation

In this subsection the results are discussed of a simulation where steady state operation of the reactor is achieved. The goal of this subsection is to provide an overview of the features of MSR-CC, i.e. which simulations results it is able to produce. It was not within the scope of this project to fully investigate all simulation results. Therefore some of the results will only briefly be discussed.

First the results are presented of the simulation reaching its steady state, i.e. the $k_{\text{eff}}$ and the power production in the reactor. Then the properties of the reactor at the steady state are discussed, e.g. the temperature distribution and fuel velocity field.

#### 7.3.1 $k_{\text{eff}}$ and power production

Figure 8 shows four graphs with data of the simulation: the power that is produced in the reactor by fission, $P_{\text{prod}}$, the power that is extracted from
7.3 Steady state operation

the system by the heat exchanger, $P_{extr}$, the maximum reactor temperature, $T_{max}$ and k-effective, $k_{eff}$.

First it is observed that at $t = 0$, $k_{eff} > 1$, so the reactor is supercritical. The power production will thus increase. Despite the increasing power production, the temperature of the reactor is decreasing. This implies that the heat exchanger extracts more power from the reactor than the amount of power that is produced by fission.

As long as $P_{extr} > P_{prod}$, the temperature will decrease. While the temperature decreases, $k_{eff}$ rises because of the negative temperature feedback coefficient. After a while the increase of the power production has caused the power production to exceed the power extraction. Now the temperature will start to rise, which in turn causes $k_{eff}$ to decrease. This procedure continues until the reactor goes from supercritical to subcritical. Then all effects described above will occur in the opposite direction.

A steady state is established when the reactor is at a temperature that yields a critical system. At that point, the power production has been scaled in such a way that it matches the power extraction by the heat exchanger at that steady state temperature. From this point, all parameters in the system have reached a steady state, except for the temperature of the reflector and breeder that surround the core, since these are not cooled and thus will slowly start to heat up.

Figure 8 shows that the power production at steady state is around 3 GW, and that the steady state temperature is 700 °C.

![Figure 8: Reactor operation while establishing steady state. The legend indicates that the fuel contains 29.5% heavy nuclei.](image-url)
7.3.2 Steady state reactor properties

This subsection discusses the properties of the reactor once the steady state operation is established that has been described in the previous subsection. All results discussed below are calculated by HEAT, except for the neutron flux profile, which is calculated by DALTON.

**Temperature** Figure 9 shows the temperature distribution of the reactor at steady state. The core has a temperature of approximately 850°C. One can see the difference in temperature between the bottom and the top of the core, caused by heat generated by fission in the core, and the heat extracted by the heat exchanger in the outer loop. The effect of the fuel flow on the temperature is clearly visible, for example at the lower side of the fertile blanket it is observed that the salt has a relatively high temperature due to the low velocity at that point.

**Velocity** Figure 10 shows the fuel flow field of the reactor. As described in section 1.3.2, the pump is modeled by setting the salt flow velocity at $v = 2 \text{ m.s}^{-1}$ downwards in the heat exchanger. It can be seen that flow is present throughout the whole core, except at a few locations where the velocity is nearly zero. For example at the left side of the breeder the velocity is quite low, which results in a higher temperature, as seen in figure 9.

**Flux** Figure 11 shows the neutron flux distribution of the reactor. Since the temperature distribution is approximately homogeneous in the reactor core, and the reactor is axisymmetric, the symmetric flux profile is as expected. It can be seen that the flux is considerably lower in the fertile blanket and the heat exchanger. Note once again that the convection of precursors has not been taken in consideration when calculating the flux distribution.

**Pressure** Figure 12 shows the pressure distribution in the core. Since the expansion of the salt is neglected in the model, the pressure should be approximately equal to the static pressure exerted by the salt that lies above the point where the pressure is measured. To verify whether this is correct, the static pressure at the bottom of the reactor from the simulation is compared with a simple calculation of the static pressure in a similar situation. The static pressure is calculated by

$$P_{\text{static}} = \rho gh$$  \hspace{1cm} (7.1)

which gives $P_{\text{static}} = 0.84\text{bar}$ for this reactor. The pressure as obtained from the simulation is around $P_{\text{static}} = 0.90\text{bar}$, which agrees fairly well with the theoretically calculated value for the static pressure. In figure 12 the linear increase of the static pressure from the top downwards can be observed.
7.3 Steady state operation

Figure 9: Temperature distribution for steady state operation. The temperature in the reactor core is nearly homogeneous, only a small temperature gradient is visible from the bottom, where the cooled fuel salt flows out of the heat exchanger, to the top of the core.

Figure 10: Fuel flow velocity distribution for steady state operation. The low flow velocity on the left side of the inner fertile blanket causes the temperature to be higher there than in the rest of the core region, since this fuel is transported to the heat exchanger slower than the rest of the fuel.
7 RESULTS

Figure 11: Neutron flux distribution for steady state operation. The convection of precursors has not been taken in consideration when calculating the flux distribution.

Figure 12: Pressure distribution for steady state operation. The approximately linear increase in pressure is a consequence of the increase of static pressure.
7.4 Heat transfer coefficient

Turbulent Kinetic Energy  Figure 13 shows the turbulent kinetic energy distribution of the fuel salt. The highest levels of TKE are found at the entrance of the heat exchanger and at the bottom part of the breeder. A large gradient in the fuel flow velocity creates a great contribution to the shear stress production term $G_k$ in the $k-\epsilon$ equations (4.9) and (4.10). With the figures 10 and 13 one can compare this effect by identifying the regions with large velocity gradients.

Turbulent viscosity Figure 14 shows the turbulent viscosity of the fuel salt. Note the difference in magnitude with the laminar viscosity, by recalling that the effective viscosity was given by $\mu_{\text{eff}} = \mu_l + \mu_t$, and the laminar viscosity of the fuel salt is $\mu_{\text{lam}} = 4 \times 10^{-3}$ Pa.s.

7.4 Heat transfer coefficient

In the previous subsection it has been observed that a steady state is established when the reactor reaches its steady state temperature, and the power production equals the power extraction. In this subsection it is evaluated how the reactor operation is influenced by the amount of heat that is removed by the heat exchanger. Figure 15 shows the simulation results for three different values of the heat transfer coefficient. At $t = 0$ the parameters $P_{\text{prod}}$, $k_{\text{eff}}$, $T_{\text{max}}$ are equal for each of the three simulations, except of course for the power extraction by the heat exchanger, $P_{\text{extr}}$. In case of the lowest heat exchanger strength, $P_{\text{prod}} > P_{\text{extr}}$, which causes the temperature to increase. For the case of the strongest heat exchanger strength, the opposite occurs. For each case it is observed that eventually a steady state is obtained in the same way as described in section 7.3.1. Comparing the steady state results, it is shown that the power production adapts to the fixed heat exchanger strength, and causes only a minor change in the operating temperature.

7.5 Fuel composition

In this subsection the influence of the fuel composition on reactor behavior is evaluated. In section 5 it has been discussed that a fluoride salt with uranium-233 and thorium is used as fuel. The results in figure 16 show what the effect is of changing the amount of heavy nuclei in this fuel. At $t = 0$ the parameters $P_{\text{prod}}$, $P_{\text{extr}}$, $T_{\text{max}}$ are equal for each of the three simulations, while $k_{\text{eff}}$ has different values. The differences in $k_{\text{eff}}$ at a given temperature is caused by the difference in neutron cross sections of the selected fuels. For all cases, the simulation eventually reaches a steady state. The graph shows that each fuel composition corresponds to its own steady state temperature. This temperature depends strongly on the fuel composition, e.g. a change in heavy nuclei fraction of 2.5% (from 27.5% to
Figure 13: Turbulent Kinetic Energy (TKE) for steady state operation. Regions with high TKE values can be identified as regions with large gradients in the fuel velocity.

Figure 14: Turbulent viscosity for steady state operation. Note the difference in magnitude of laminar and turbulent viscosity. The laminar viscosity is $\mu_{\text{lam}} = 4.10^{-3}$ Pa.s, many orders of magnitude lower than the turbulent viscosity.
30.0%) results in a steady state temperature difference of roughly 650°C. Merle-Lucotte et. al. [30] found steady state operation at a temperature of 630°C, with a heavy nuclei fraction in the fuel of 22% and a power production of 2500MW$_{th}$. Figure 16 shows that in this model an operating temperature of 630°C is established with a heavy nuclei fraction of roughly 29%. Whether this difference is caused by the fuel composition used or the reactor geometry should be investigated.

### 7.6 Grid resolution

In this subsection the influence of the grid resolution on the calculation results is investigated. Theoretically, a grid with an infinitely high resolution would yield the exact solution of the problem. Therefore it is expected that the results of the calculation will converge for increasing resolution. Figure 17 shows that the simulation results do seem to converge for increasing grid resolution, but extra simulations with higher grid resolutions are desirable to confirm the results. Unfortunately MSR-CC is not able to perform calculations with a grid resolution higher than 45:70, due to an unresolved problem in the code. Therefore this analyses should be proceeded when this code problem is resolved.

### 7.7 Power scaling rate

In a broad sense, the coupled calculation is a matter of repeating DALTON and the HEAT calculations in a cycle. DALTON performs a time independent calculation of the multigroup diffusion equation, while HEAT calculates the time evolution of thermal hydraulic properties. Therefore the time span must be specified for which HEAT calculates the momentum and energy balance, until it 'exits' and the cycle will proceed. This time span defines the rate at which the power is scaled, since this is executed every time the cycle is repeated. The advantage of a long time span for the HEAT calculation is that it speeds up the calculations, since for every cycle the set of cross sections must be updated and DALTON must be executed. Therefore it is of interest to evaluate the influence of the power update rate on the simulation. Figure 18 shows the results of three simulations, each with a different power update rate. The rate is expressed as the number of time steps that is calculated in one HEAT calculation. The rates used in the simulations are 1, 2, and 5 time steps. All simulations have calculated 1000 time steps in total, which means that 1000 (1000*1), 500 (500*2) and 200 (200*5) cycles were performed.

At t = 0 all parameters of the simulation are equal, which is to be expected since only the power scaling rate changes, which is a time dependent effect. The time evolution of the simulation shows that for the highest frequency, the oscillations in the parameters are the strongest. The case of the low
Figure 15: Influence of the heat transfer coefficient on the simulation. All simulations establish steady state operation, with only a minor change in the reactor operating temperature. The heat removal, and thus power production, are strongly influenced by changing the heat transfer coefficient.

Figure 16: Influence of the fuel composition on the simulation. The percentages indicated in the legend represent the molar fraction of heavy nuclei in the fuel.
frequency shows only modest oscillations in the parameters. This effect is caused by the time that it takes for the amount of heat removal of the heat exchanger to respond to the changes in power production. When HEAT is calculated for a long time span, the fuel salt in the core has enough time to flow to the heat exchanger and 'communicate' its temperature to the heat exchanger, which adapts its heat removal to this temperature. In this way steady state operation is reached gradually, since power production and heat removal are more or less balanced. When HEAT is calculated for a short time span, the fuel salt that enters the heat exchanger has a temperature that corresponds to a power production of a few cycles earlier in the simulation, because of the time required to flow to that region. Thus the heat exchanger adapts its heat removal to a power production that corresponds to a few cycles back in the simulation. This results in the 'overshoots' observed in figure 18. Despite the large differences in the first phase of the simulation, the steady state situations of all cases are equal. This is a desirable result, since the variable that is changed here should have no influence on the final steady state operation.

7.8 Convection of precursors

In this subsection the results are discussed from the numerical model that was introduced in section 2.3. Two scenario’s have been calculated, the case where there is no fuel flow, and the case where there is fuel flow. The calculations have been performed with data for $\Phi_g$ and $\nu\Sigma_f^g$ from a steady state operation of the simulation as described in section 7.3. Data for $\beta$ and $\lambda$ have been obtained from the Oak Ridge MSRE experiment [24], which was performed with similar materials. In Table 6 the decay constants and the relative precursor yields can be found. The total precursor yield is $\beta_{tot} = 0.0029$, so the yield per precursor group is $\beta_i = \beta_{tot}\beta_{rel}$. From the numerical model, precursor concentrations have been obtained for all precursor groups separately. To indicate the contribution of the precursors to the neutron flux, the total delayed neutron source has been calculated. This is the $\sum_\lambda\lambda_iC_i$ term in the diffusion equation (2.5). Figures 19 and 20 show the delayed neutron source for the situations without flow and with flow respectively.

The results show that the delayed neutron source is distributed differently when the convection of the precursors is taken into account. If no convection of the precursors is present, the delayed neutrons will appear at the same place as where the precursor nuclei was created by a fission reaction. The neutron source distribution of the delayed neutrons is therefore equal to the neutron flux distribution. If convection is taken into account, the precursor nuclei that is created will begin to move along with the flow profile. The location where the precursor will decay, and thus a neutron will appear, depends on the decay constant of the precursor and the flow veloc-
7 RESULTS

Figure 17: Influence of grid resolution on reactor behavior. The simulation results appear to be converging for increasing grid resolution. Unfortunately is it not possible to perform the simulations with a grid resolution higher than 45:70 due to an unresolved problem in the code.

Figure 18: Influence of the power scaling frequency on reactor behavior. A ratio of 500:2 means that 500 cycles are performed, with HEAT calculating 2 time steps per cycle. The time step size is $\Delta t = 0.5s$
ity. Considering this, and recalling the flow profile in figure 10 and the flux
distribution in figure 11, one expects that the precursors that are created in
the reactor core will flow upwards and to the heat exchanger. Consequently,
the precursor nuclei will decay in a region outside the inner core, as shown
in figure 20.

<table>
<thead>
<tr>
<th>Precursor group $i$</th>
<th>Group decay constant $\lambda_i$</th>
<th>Relative group yield $\beta_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0126</td>
<td>0.0821</td>
</tr>
<tr>
<td>2</td>
<td>0.0337</td>
<td>0.2963</td>
</tr>
<tr>
<td>3</td>
<td>0.1390</td>
<td>0.2485</td>
</tr>
<tr>
<td>4</td>
<td>0.3250</td>
<td>0.2838</td>
</tr>
<tr>
<td>5</td>
<td>1.1300</td>
<td>0.0546</td>
</tr>
<tr>
<td>6</td>
<td>2.5000</td>
<td>0.0347</td>
</tr>
</tbody>
</table>

Table 6: Precursor decay constants and relative group yields. Data obtained from
the Oak Ridge MSRE experiment [24].

### 7.9 Accident scenario’s

**Pump crash scenario** In this subsection a pump crash scenario is sim-
ulated. The simulation is performed by first bringing the reactor operation
to a steady state as described in section 7.3, and then from $t=0$ starting a
velocity transient of the pump. The transient is defined as

$$v_{\text{pump}} = v_0 e^{-\frac{t}{\tau}}$$  \hspace{1cm} (7.2)

with $v_0 = 2 \text{ m.s}^{-1}$ the normal operating velocity of the pump and $\tau = 40s$
the time constant for the transient of the pump. With this transient the
pump velocity falls down to zero in roughly a minute. The results in figure
21 show that the temperature increases strongly, since the heat generation
in the core by power production is not transported to the heat exchanger by
fuel flow. Because of the negative feedback coefficient, a strong temperature
increase causes the reactivity to decreases rapidly. This causes the power
production to decrease, and as a consequence also the power extraction
decreases. The fuel in the core has reached a high temperature, and since
active cooling is not possible without circulation, the fuel salt temperature
remains high.

**Heat exchanger failure scenario** This subsection considers the scenario
that the heat exchanger fails to operate. After the simulation is brought to
steady state operation, a transient in the heat transfer coefficient of the heat
exchanger is started. The transient is defined as

$$h = h_0 e^{-\frac{t}{\tau}}$$  \hspace{1cm} (7.3)
Figure 19: Delayed neutron source distribution when no fuel flow is present. The left and right side represent a simplified model of the inner side of the core and the heat exchanger channel, respectively.

Figure 20: Delayed neutron source distribution when fuel flow is present. The left and right side represent a simplified model of the inner side of the core and the heat exchanger channel, respectively.
7.9 Accident scenario’s

Figure 21: Pump failure scenario. At t=0 the reactor was operating at a steady state, as described in section 7.3. From t=0 the velocity transient is started, as shown in the top right graph.

with \( h_0 = 900 \times 10^3 \text{W.m}^{-3}\text{.K}^{-1} \) the normal operating heat transfer coefficient and \( \tau = 40\text{s} \) the time constant for the transient of the pump. With this transient the operation of the heat exchanger stops in roughly a minute. In this situation the heat generated by power production is not extracted by the heat exchanger while it is pumped around, but the salt is delivered back to the reactor core with the same temperature as it entered. So the fuel salt temperature starts to increase, followed by a decrease in the reactivity due to the negative feedback coefficient. After the system has become subcritical, the power production approaches zero. Finally, a state is achieved where the fuel salt circulates with an increased temperature around the system, with no power production and no cooling.
Figure 22: Heat exchanger failure scenario. At t=0 the reactor was operating at a steady state, as described in section 7.3. From t=0 the heat transfer coefficient transient is started, as can been seen in the bottom right graph.
8 Conclusions

- The neutronics and the thermal hydraulic properties of the non moderated Thorium Molten Salt Reactor (TMSR-NM) have been analyzed. A coupled code system has been established that combines DALTON and HEAT calculations for the TMSR-NM, named MSR-CC.

- In this project the TMSR-NM operation with uranium based fluoride salt fuel has been considered, and simulations have been performed with several compositions of the fuel. A fuel salt with a molar fraction of 29.5% of heavy nuclei resulted in a power output of nearly 3GW at a reactor temperature of 700 °C.

- The temperature feedback coefficient of fuel salt with a molar fraction of 29.5% of heavy nuclei was found to be $\alpha \approx -6$ pcm/K. This is an important property with respect to reactor safety, and the result is in agreement with research by Merle-Lucotte et. al. [13].

- The heat removal determines the power output of the reactor. Since no information was available about the physical properties of the heat exchanger, the heat transfer coefficient could be chosen freely, and thus the power output could reach any value. A more sophisticated physical model of the heat exchanger is required to determine its physical limits.

- It has been shown that for an increasing grid resolution, the simulation results are converging. Simulations with even higher grid resolutions are desirable to confirm the convergence, but an unresolved error in the MSR-CC code did not allow simulations with a grid resolution higher than $n_i:n_j=45:70$.

- It has been shown that changing the power scaling rate, i.e. the time span over which HEAT performs its calculations per cycle, strongly affects the reactor behavior during unsteady operation, but the final results are all equal for steady state operation.

- The physical model as used by DALTON to compute the neutron flux is not fully complete, since the convection of precursor nuclides is not implemented. Attempts have been made to implement an alternative version of DALTON that provides these functionalities, but this turned out not to be feasible since the reactor model it uses was not compatible with the model in this project. To evaluate the effect of the convection of precursors on the neutron flux distribution, external numerical calculations have been performed. This indicated that the fuel flow causes the neutron flux to be distributed differently, with a higher neutron source due to precursors decay at the top of the reactor core and in the heat exchanger channel.
9 Recommendations

About the TMSR-NM

- The fuel material composition used in this model was an approximation of the fuel used by CNRS. Future simulations should be performed with a more accurate reconstruction of this fuel, to make validation of the results more reliable. The relevance of this has become clear in the simulation results, where it was shown that the steady state operation is strongly affected by the fuel composition.

- The TMSR-NM allows for two types of fuel, one based on uranium and one based on plutonium and minor actinides. This project has only evaluated the uranium fuel, leaving the plutonium fuel available for future research.

- The reactor materials should be thoroughly investigated. Some of the material properties used in this project were estimations, found by using properties of similar materials. Accurate information about the density, viscosity and heat conduction of the materials would improve the model.

- The cooling of the breeder blanket, that surrounds the core, should be investigated. Since this blanket is surrounded from all sides by the high temperatures of the salt, it heats up quickly to the temperature of the reactor core. It should be investigated how this is planned in the theoretical model of the TMSR-NM.

- A better physical model for the heat exchanger should be implemented. In this project there was no insight in the physical capabilities of the heat exchanger, so no realistic estimation could be made about the power production.

About the code system MSR-CC

- For long term simulations, fuel burn-up evaluation should be implemented. This could be especially interesting to evaluate the breeder capabilities.

- The convection of precursors in the neutron diffusion equation should be implemented to make the neutron model complete.
• Due to an unresolved error, the maximum grid resolution possible is $ni:nj=45:70$. This error should be resolved, also since it might affect the simulations at the lower grid resolutions.

• In the code system no error- or warning messages have been implemented. Adding these to the code system can avoid tedious situations for future users.
Part II
Manual of MSR-CC
Summary

This document is the manual of the MSR-CC (Molten Salt Reactor - Coupled Calculations) code system. The code system is a coupling of neutronics calculations and thermohydrodynamic calculations, performed by DALTON and HEAT respectively. The system applies its calculations on the non-moderated Thorium Molten Salt Reactor (TMRS-NM). This manual provides all information necessary for a user to control MSR-CC.

Some notes

- Filenames and command lines are stated in typewriter fonts.
- When a command line is stated, this command should be run from the home directory of MSR-CC (./), unless stated otherwise.
- In the manual, filenames are indicated without their corresponding directory. For an overview of the directories and the files, see chapter H of the appendix.
- All filenames typed after the qsub command are send to the que of the hpc11.tudelft.net cluster. These files should therefore be que instruction files.
- Fortran files are first compiled and then executed:
  - ifort filename.f90
  - ./a.out
- Input files for SCALE are executed with the command
  - batch5 -m scaleinputfile
A Introduction

A.1 About MSR-CC
The MSR-CC (Molten Salt Reactor - Coupled Calculations) code system is developed at the research group Physics of Nuclear Reactors, part of the Radiation, Radionuclides and Reactors department of the Delft University of Technology. The code system is a coupling of neutronics calculations and thermohydrodynamic calculations, performed by DALTON and HEAT respectively. The system applies its calculations on the non-moderated Thorium Molten Salt Reactor (TMRS-NM).

A.2 About the manual
This manual describes how MSR-CC works and how it can be controlled by the user. The in- and output of routines is described, and the contents of important files are explained. First the initialization part is described, everything that is done before the coupled calculations start. Then the cycle of the coupled calculations within the simulation is discussed. For each step it is described which routines are used and which parameters can be adjusted by the user. The appendix of this document can be used as a quick reference to most input parameters and grid construction. This manual does not describe the physical background of the nuclear reactor, or the theoretical model that is used in the code systems. For information about these topics, please refer to the first part of this document.

This first version of MSR-CC is developed as a MSc thesis project. Please note that this version does not contain automated warnings or error messages. Therefore it is important to be very careful when modifying anything in MSR-CC.

A.3 Outline
First an outline is provided of the MSR-CC in B, indicating which code systems are used and how they are connected. In section C the procedure to create a cross section work library with SCALE is explained. Section 3.5 discusses the interpolation of cross sections by MIX. In the sections E and F information is provided about DALTON and HEAT respectively. Finally, section G presents what output is delivered by MSR-CC and how some of the data can be plotted.
B MSR-CC

B.1 Structure of the MSR-CC

The section describes which code systems are involved in the MSR-CC and how they are communicating. More detailed information about the individual code systems is provided in the succeeding sections. Below is list is provided with the steps taken in a simulation by the MSR-CC.

1. The SCALE code system makes a cross section work library for the used materials of the reactor.
2. MIX calculates a specific set of cross sections, corresponding to the current reactor temperature.
3. DALTON calculates the neutron flux profile and $k_{\text{eff}}$ of the reactor, from that calculates the power distribution.
4. HEAT calculates the fuel flow field and the temperature of the reactor. Also it determines how much power the heat exchanger extracts from the system. The temperature is used as input for MIX, and the cycle is started over from step 2.
5. When MSR-CC has performed the full simulation, VisIT is used to visualize the results.

In figure 23 the cycle described above is schematically shown.

B.2 Executing a MSR-CC simulation

In this section the most important commands that are used to operate MSR-CC are discussed. In the succeeding sections information will be provided about every step that is performed within the system, but executing a MSR-CC simulation is done by only a few commands that combine many of these minor steps. Basically the execution consists of three steps, which are described below:

Creating the cross section library

The cross section library is created with the command

- `qsub quemakelibrary` (executed from directory `./SCALE/`).

Executing the simulation

The simulation is started the command

- `qsub quejob`.

This command sends the executable file `run` to the que. `run` is a Perl script that combines all commands necessary to execute the simulation cycle.
Plotting the results  A part of the results can be visualized directly by running VisIt

```
visit -cli -s ./plots/visitscript
```

Other output files can be visualized or analyzed manually by the user.

Figure 23: Overview of the coupled calculation cycle of the MSC-CC code system. The simulation consists of two parts, first the initialization phase, and then the coupled calculation cycle. Within the boxes it is indicated which code system performs the calculation. In between the boxes it is indicated what physical quantity is communicated.
C SCALE: Creating the cross section work library

C.1 Introduction

This section describes the initialization of the simulation, before starting the coupled calculations. The goal of the initialization is to create a work library with the nuclear cross sections of all the materials used in the reactor. Not only are the cross sections different for the various materials, but the cross sections depend also on the temperature. A work library is therefore required that contains cross section data for the whole temperature range in the simulation. See section 3 for more information about this procedure.

C.2 Structure of the library

The library is a package of \( n \) files, each file representing the set of cross sections for one specific temperature. The user specifies the array of \( (n) \) temperatures for which the library is made. See figure 24 for an overview. The files are named \texttt{library.1.n}, with \( n \) being the number specifying the temperature (1 is the lowest temperature).

![Figure 24: Schematic overview of the library structure](image-url)
C.3  Routines and execution

C.3.1  makescale.f90

This fortran routine creates the input files for the SCALE code system, denoted as libraryinput.1.n\(^3\). The routine reads input from the MAININPUT file, but also requires input in the routine itself. Here it is summarized which input it reads and thus is adjustable by the user. Figure 25 shows a schematic overview of the in- and output of the routine.

**Input**  The following information is read from the MAININPUT file: the library name (30)\(^4\), the no. of temperatures in the library (32) and the corresponding temperatures (33+). Within the routine itself, input must be provided about the material properties, like molar masses and densities of the used elements. The most important parameter that can be adjusted within the file makescale.f90 is the composition of the materials.

**Output**  The output consists of n SCALE input files, named libraryinput.1.n, and an executable file named runbatch5.

![Schematic overview of makescale.f90 routine.](image)

C.3.2  runbatch5

This PERL script executes the SCALE code system for every library input file that has been created with makescale.f90. The output of this

---

\(^{3}\)The name of the library can be adjusted in MAININPUT. In the manual ‘library’ is used as the default name.

\(^{4}\)The number represents the line number of MAININPUT where the input can be found.
script is the final set of cross section library files created by SCALE, named `library.1.n`. Figure 26 shows the structure.

![Diagram of runbatch5 routine](image)

**Figure 26**: Schematic overview of runbatch5 routine.

### C.3.3 quemakelibrary

This PERL script is the execution file for making the library. It sends the following commands to the que:

- `ifort makescale.f90`
- `./a.out`
- `runbatch5`

### C.3.4 Execution

When the input is stated correctly in `MAININPUT` and within `makescale.f90`, the following command executes all steps described within this section, and thus automatically creates the library:

- `qsub quemakelibrary` (executed from the `./SCALE/` directory!)
D MIX: Cross section interpolation

D.1 Introduction

The work library that has been made in the initialization phase consists of cross sections for the reactor materials, at a discrete set of temperatures. To obtain the cross sections that correspond to any given temperature, the MIX code system\(^5\) interpolates the cross sections of the work library with the neighboring temperatures. More information on this code system can be found in section 3.5.

D.2 Routines and execution

D.2.1 mix

The mix executable reads the reactor zone temperatures from the file fort.223, and calculates which coefficients are needed to interpolate the cross sections from the work library. From the MAININPUT file it reads information about the work library, like the name and the temperatures it contains. The output consists of the file fort.12, in which the coefficients are written, the library file numbers that should be used and the ID’s of the interpolated nuclides. Also a log file is created, named logMIX.

\(^5\)Created by B. Boer, b.boer@tudelft.nl.
D.2.2 wax_ice

This PERL script reads the interpolation coefficients from fort.12 and puts them into an input file for the SCALE module named WAX. Then the script writes input commands for the module ICE to produce one set of cross sections. Finally it creates the SCALE input file library.scale5.inp. A log file with the name logWAX is created.

![Diagram of wax_ice](image1)

Figure 28: In- and output of the wax_ice program

D.2.3 batch5

Batch5 is the routine used to execute the input file created by wax_ice. The output is the final work library with the interpolated cross sections, library.xs. A log file with the name library.scale5.output is created.

![Diagram of batch5](image2)

Figure 29: In- and output of the batch5 program

---

Read documentation about SCALE for information about the WAX and ICE modules.
D.2.4 Execution

MIX is one of the steps in the calculation cycle. It is executed every time HEAT has calculated the new reactor temperature. MIX then uses this temperature to 'update' the cross sections, which is then used as input for DALTON. One run of MIX consists of the following commands:

- mix
- wax.ice
- batch5 -m library.scale5.inp
E   DALTON: neutron flux calculation

E.1   Introduction

The in-house code system DALTON\(^7\) calculates the neutron flux profile and \(k_{\text{eff}}\). From this the power production in the reactor is obtained. For more information on DALTON, see section 2.

E.2   Routines and execution

E.2.1   makedalton.f90

This routine writes the input file for DALTON. It uses input variables from MAININPUT. The output file is named inputdalton. A description of the contents of this file is provided by comments in the file itself. The power scaling, as discussed in section 2.2.4, is also performed within this routine. This is done by reading the values of \(k_{\text{eff}}\) and power production from the files \texttt{keff} and \texttt{Ptot} respectively. After the power production is updated, it is used as input for DALTON in the file inputdalton, and also written back to \texttt{Ptot}.

![Figure 30: In- and output of the makedalton.f90 routine](image)

E.2.2   dalton63

This executable calculates \(k_{\text{eff}}\) and the neutron flux distribution. The program uses the file \texttt{fort.1} as input, which is linked to the input file that was just created, inputdalton, and gives instructions on the geometry and the grid. The program uses the file \texttt{library.xs} as the library for cross sections. The calculated power production and neutron flux profile are written to \texttt{fort.33} and \texttt{fort.88} respectively. A log file of the calculation is written to logDALTON, in which also the calculated \(k_{\text{eff}}\) is stored.

\(^7\)Created by D. Lathouwers, d.lathouwers@tudelft.nl.
E.2.3  grabk.f90

This short fortran routine extracts the $k_{\text{eff}}$ value from the log file logDALTON and stores it in the file keff. This routine has been implemented to make it easier to call k-effective as a variable from other routines.

E.2.4  Execution

DALTON is the part of the simulation cycle where the $k_{\text{eff}}$ and the neutron flux distribution is calculated. This part consists of the following commands:

- ifort makedalton.f90
- ./a.out
- dalton63
- ifort grabk.f90
- ./a.out
F HEAT: energy and momentum

F.1 Introduction

The in-house code HEAT calculates the temperature- and the velocity field from the energy and momentum equations, respectively. It also calculates a number of other reactor properties, which are not used as input for the other routines:

- Pressure
- Viscosity
- Turbulent kinetic energy (TKE)
- Dissipation rate of TKE

When HEAT is finished, a new temperature field has been calculated. From here the cycle can start over; MIX uses the new temperatures to update the cross section work library, which DALTON in turn uses to calculate the new $k_{\text{eff}}$ and neutron flux distribution. Then HEAT will calculate what the new temperature becomes. Etc. etc..

F.2 Routines and execution

F.2.1 HEAT

The input for the HEAT code system comes mainly from the file MAININPUT, except for information about the power production, which is read from fort.33. HEAT produces the following output files:

- fort.223, the average temperature per zone, used as input for MIX
- relchanges, the relative changes of reactor properties w.r.t. the previous time step, this might be useful when monitoring the stability of the simulation
- statistics, information about the power production and power extraction

F.2.2 Execution

This part of the simulation is executed by the command

- HEAT
G Simulation results

This section describes which output files are created containing the simulation results, and how they can be visualized. MSR-CC creates three types of output:

- Log files of various steps of the simulation, e.g. to check errors
- Simulation data in .tec format, for visualization with VisIt\(^8\)
- Unformatted simulation data, which the user can manually plot or evaluate

G.1 Log files

The following log files are created during or at the end of the simulation:

- logBATCH5
- logDALTON
- logHEAT
- logMIX
- logWAX

The filenames indicate from which subroutine it is a log file.

G.2 Simulation data for VisIt visualization

VisIt is a linux based visualization tool that is able to output graphs through a X11 interface. For convenience, a script is written that automatically plots a number of graphs in VisIt. The script is called visitscript, and outputs 8 graphs. Below the names of the plotted .tec files are listed.

- visitvelocity.tec, fuel velocity field

\(^8\)For more information about this program, see https://wci.llnl.gov/codes/visit/.
G.3 Unformatted simulation data

- visittemp.tec, reactor temperature
- visitpressure.tec, core pressure
- visitheatexchanger.tec, power extracted by heat exchanger
- visitflux.tec, neutron flux field
- visiteps.tec, dissipation of turbulent kinetic energy
- visitkturb.tec, turbulent kinetic energy
- visitmu.tec, dynamic viscosity coefficient

For more information about the physical properties named above, and a brief discussion of some previous results, please refer to part I of this document.

G.2.1 Execution

To run the visit script, type from the main directory

```
visit -cli -s ./plots/visitscript
```

G.3 Unformatted simulation data

The following files contain unformatted simulation data, and can be used as data for plotting or evaluation directly.

- hxarray, power extracted by the heat exchanger
- keffarray, $k_{eff}$ of the reactor
- Ptotarray, power created by fission
- Tmaxarray, maximum (zone-average) temperature in the reactor

All above files have the same properties: the length of the array is equal to the no. of cycles calculated\textsuperscript{9}, and represent the value at the moment a cycle is ended.

\textsuperscript{9}Line 27 of MAININPUT.
Part III
Appendix

H Directory structure

This section provides a list of all the files that have been discussed in this manual. Since in the manual the path names of the files are not indicated, this list can be used to trace back the files when using MSR-CC.

- ./MAININPUT
- ./run
- ./quejob
- ./SCALE/makescale.f90
- ./SCALE/runbatch5
- ./SCALE/quemakelibrary
- ./SCALE/libraryinput.1.n
- ./SCALE/library.1.n
- ./mixer/mix
- ./mixer/wax_ice
- ./mixer/fort.12
- ./mixer/fort.223
- ./mixer/library.scale5.inp
- ./mixer/library.scale5.output
- ./mixer/library.xs
- ./mixer/Tmaxarray
- ./mixer/hxarray
- ./Dalton/dalton63
- ./Dalton/inputdalton
- ./Dalton/makedalton.f90
- ./Dalton/Ptotarray
- ./Dalton/keffarray
- ./Dalton/grabk
- ./Dalton/fort.33
- ./Dalton/daltonvisit.f90

- ./HEAT/HEAT

- ./plots/visitscript
- ./plots/visitvelocity.tec
- ./plots/visittemp.tec
- ./plots/visitpressure.tec
- ./plots/visitheatexchanger.tec
- ./plots/visitflux.tec
- ./plots/visiteps.tec
• ./plots/visitkturb.tec
• ./plots/visitmu.tec

• ./logs/logBATCH5
• ./logs/logDALTON
• ./logs/logHEAT
• ./logs/logMIX
• ./logs/logWAX
• ./logs/hxarray
• ./logs/keffarray
• ./logs/Ptotarray
• ./logs/Tmaxarray
I  Description of important input parameters

I.1  MAININPUT

Figure 34 shows a screenshot of the main input file of MSR-CC, MAININPUT. The line numbers are labelled at the left side. According to these numbers, the input at every line is described below. The sizes are given in [m] and the velocity in [m/s]. All temperatures are in Kelvin. See figures 35 and 36 for a schematic overview of the parameters.

```plaintext
1 .T.          ! Restart HEAT
2 .T.          ! Cylindrical HEAT
3 40           ! ni
4 60           ! nj
5 5            ! ni_refl
6 5            ! nj_refl
7 20           ! ral
8 30           ! ra3
9 20           ! rb1
10 40          ! rb3
11 0.d0        ! xmain
12 1.90d0      ! xmax
13 0.d0        ! ymin
14 2.6d0       ! ymax
15 0.5d0       ! refl thickness
16 1.25d0      ! xmain
17 1.65d0      ! xymax
18 0.4d0       ! ybmin
19 2.2d0       ! ybmax
20 1           ! No of timesteps in HEAT
21 5           ! No of inner iterations HEAT
22 0.5d0       ! dt
23 -2.0d0      ! Outlet velocity (vнос)
24 2000.d6     ! Reference Power
25 900.d3      ! Heat transfer coef. h
26 500.d0      ! Temp bottom reflector
27 500.        ! No of cycles
28 353.d0      ! Initial reflector temperature
29 923.d0      ! Initial core temperature
30 library     ! Name of xs lib files (max. 15 char)
31 1           ! No of variables (Tc)
32 7           ! No of temperatures in xs lib
33 300.d0      ! Temperatures
34 600.d0      ! No. r intervals (not used)
35 900.d0      ! No. of reactor zones defined
```

Figure 34: Screenshot of the main input file
1. HEAT starts from the last state of the previous run (.T.) or with the initial conditions (.F.)
2. Grid is set in cylindrical (.T.) or cartesian coordinates (.F.)
3. No. of horizontal grid points in core
4. No. of vertical grid points in core
5. No. of horizontal grid points in reflector
6. No. of vertical grid points in breeder
7. Horizontal grid point at left outside of breeder
8. Horizontal grid point at right inside of breeder
9. Vertical grid point under downside of breeder
10. Vertical grid point upper inside of breeder
11. Horizontal coordinate of core centre (default = 0.d0)
12. Radius of the core
13. Vertical coordinate of core centre (default = 0.d0)
14. Height of the core
15. Thickness of reflector
16. Horizontal left coordinate of breeder
17. Horizontal right coordinate of breeder
18. Vertical bottom coordinate of breeder
19. Vertical top coordinate of breeder
20. No. of time steps HEAT makes during one cycle
21. No. of inner iterations HEAT makes during one time step
22. Time step size
23. Pump velocity
24. Initial power output of DALTON
25. Heat transfer coefficient of heat exchanger in HEAT
26. Fixed temperature of the bottom of reactor
27. No. of cycles in the simulation
28. Initial temperature of reflector area
29. Initial temperature of core area
30. Name of the cross section library files
31. No. of variables to determine cross section interpolation (default = 1)
32. No. of temperatures in cross section library
33. Temperatures of the cross section library
34. 
35. 
36. 
37. 
38. 
39. 
40. No. of r intervals (not used)
41. No. of reactor zones to interpolate
I.2 Grid definition

The main parameters of the reactor grid and size are illustrated in figures 35 and 36. Figure 37 shows how the reactor is divided in zones. These zones are used by DALTON to calculate the average power produced per zone.

Figure 35: Horizontal grid construction. Note the difference between the quantities that indicate lengths and the ones that indicate grid quantities.
I.2 Grid definition

Figure 36: Vertical grid construction. Note the difference between the quantities that indicate lengths and the ones that indicate grid quantities.
Figure 37: The zones distribution of the reactor. The zones distribution is used when transferring temperature and power information, i.e. these quantities are averaged over the zone and then transferred.
## J Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>$W.m^{-3}.K^{-1}$, heat transfer coefficient</td>
</tr>
<tr>
<td>$h$</td>
<td>m, height</td>
</tr>
<tr>
<td>$k$</td>
<td>- multiplication factor</td>
</tr>
<tr>
<td>$k$</td>
<td>$m^2.s^{-2}$, turbulent kinetic energy</td>
</tr>
<tr>
<td>$u$</td>
<td>$m.s^{-1}$, horizontal velocity component</td>
</tr>
<tr>
<td>$v$</td>
<td>$m.s^{-1}$, vertical velocity component / total scalar velocity</td>
</tr>
<tr>
<td>$x$</td>
<td>- molar fraction</td>
</tr>
<tr>
<td>$x$</td>
<td>m, horizontal coordinate</td>
</tr>
<tr>
<td>$v$</td>
<td>$m.s^{-1}$, velocity vector</td>
</tr>
<tr>
<td>$f$</td>
<td>Pa.m$^{-1}$, body forces</td>
</tr>
<tr>
<td>$r$</td>
<td>m, location vector</td>
</tr>
<tr>
<td>$p$</td>
<td>Pa, pressure</td>
</tr>
<tr>
<td>$c$</td>
<td>$J.kg^{-1}.K^{-1}$, specific heat capacity</td>
</tr>
<tr>
<td>$A$</td>
<td>$m^2$, area</td>
</tr>
<tr>
<td>$C$</td>
<td>$m^{-3}$, precursor neutron concentration</td>
</tr>
<tr>
<td>$D$</td>
<td>m, diameter / characteristic length / neutron diffusion coefficient</td>
</tr>
<tr>
<td>$E$</td>
<td>J, energy</td>
</tr>
<tr>
<td>$L$</td>
<td>- loss term</td>
</tr>
<tr>
<td>$M$</td>
<td>g.mol$^{-1}$, molar mass</td>
</tr>
<tr>
<td>$N$</td>
<td>$m^{-3}$, number density</td>
</tr>
<tr>
<td>$I$</td>
<td>- total number of precursor groups</td>
</tr>
<tr>
<td>$Nu$</td>
<td>- dimensionless Nusselt number</td>
</tr>
<tr>
<td>$P$</td>
<td>W, power production</td>
</tr>
<tr>
<td>$P$</td>
<td>N m$^{-2}$, pressure</td>
</tr>
<tr>
<td>$Pr$</td>
<td>- dimensionless Prandtl number</td>
</tr>
<tr>
<td>$Re$</td>
<td>- dimensionless Reynolds number</td>
</tr>
<tr>
<td>$S$</td>
<td>- source term</td>
</tr>
<tr>
<td>$T$</td>
<td>K, temperature</td>
</tr>
<tr>
<td>$V$</td>
<td>$m^3$.mol$^{-1}$, molar volume</td>
</tr>
<tr>
<td>$V$</td>
<td>$m^3$, volume</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>- temperature feedback coefficient</td>
</tr>
<tr>
<td>$\beta$</td>
<td>- delayed neutron yield</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>$m^2.s^{-3}$, dissipation of turbulent kinetic energy</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>- radiation emissivity</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>$W.m^{-1}.K^{-1}$, heat conduction</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>$s^{-1}$, precursor decay constant</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Pa, shear stress</td>
</tr>
<tr>
<td>$\nu$</td>
<td>- number of neutrons created per fission</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>$m^{-1}$, macroscopic neutron cross section</td>
</tr>
<tr>
<td>$\phi$</td>
<td>$m^{-2}.s^{-2}$, neutron flux</td>
</tr>
<tr>
<td>$\Phi$</td>
<td>$m^{-2}.s^{-2}$, multigroup neutron flux</td>
</tr>
<tr>
<td>$\rho$</td>
<td>kg.m$^{-3}$, density</td>
</tr>
<tr>
<td>$\rho$</td>
<td>- reactivity</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Pa.s, dynamic viscosity coefficient</td>
</tr>
<tr>
<td>$\chi$</td>
<td>- neutron fission spectrum</td>
</tr>
<tr>
<td>$\Xi$</td>
<td>$s^{-1}$, rate of strain tensor</td>
</tr>
<tr>
<td>$\nabla$</td>
<td>Pa, divergence of stress</td>
</tr>
<tr>
<td>Subscripts</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>$t$</td>
<td>turbulent / total</td>
</tr>
<tr>
<td>$l$</td>
<td>laminar</td>
</tr>
<tr>
<td>$c$</td>
<td>core</td>
</tr>
<tr>
<td>$r$</td>
<td>reflector</td>
</tr>
<tr>
<td>$i$</td>
<td>identifier</td>
</tr>
<tr>
<td>$p$</td>
<td>constant pressure</td>
</tr>
<tr>
<td>$tot$</td>
<td>total</td>
</tr>
<tr>
<td>$g$</td>
<td>neutron energy group</td>
</tr>
<tr>
<td>$d$</td>
<td>delayed neutrons</td>
</tr>
<tr>
<td>$p$</td>
<td>prompt neutrons</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Superscripts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f$</td>
<td>fission</td>
</tr>
<tr>
<td>$s$</td>
<td>scattering</td>
</tr>
<tr>
<td>$R$</td>
<td>radiation</td>
</tr>
<tr>
<td>$C$</td>
<td>convection</td>
</tr>
<tr>
<td>$cd$</td>
<td>conduction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constants</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\kappa$</td>
<td>0.41</td>
<td>von Karman constant</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>0.8</td>
<td>emissivity coefficient for radiation</td>
</tr>
<tr>
<td>$g$</td>
<td>9.81 m.s$^{-1}$</td>
<td>gravitational acceleration</td>
</tr>
<tr>
<td>$N_A$</td>
<td>$6.022 \cdot 10^{23}$ mole$^{-1}$</td>
<td>Avogadro’s number</td>
</tr>
<tr>
<td>$\sigma_b$</td>
<td>$5.67 \cdot 10^{-8}$ W.m$^{-2}$.K$^{-4}$</td>
<td>Stefan Boltzmann’s constant</td>
</tr>
</tbody>
</table>
References


[12] Kophazi, J., Lathouwers, D., Three-dimensional space and time-dependent analysis of molten salt reactors, Delft University of Technology


REFERENCES