Simulation of a fast molten salt reactor

R.J.S. van ’t Eind
July 2011
PNR-131-2011-009
Abstract

One of the generation four reactor concepts is the molten salt reactor. The coupled code system described and used in this thesis simulates a non moderated molten salt reactor. This non moderated version does not contain graphite and is able to incinerate waste from other reactors due to its fast spectrum. The fuel used in the reactor examined consists of LiF-ThF$_4$-UF$_4$.

A simulation of this kind of reactor was already partially developed, but the flow of precursors was not yet implemented. In this work, this was implemented together with some other improvements. The simulation can now run while using a finer mesh than before. Also the simulation is now structured more intuitively. For example, the transients can be used and adapted without re-compiling large sections of the code.

In the simulations, no material evolution is calculated. Also, the heat exchanger is modeled by using a constant transfer coefficient.

While simulating the reactor with the pump forcing the fuel around at 2[m/s] the reactor reached an equilibrium temperature of a little over 1050[K]. The temperature feedback coefficient of this reactor was estimated at approximately -6[pcm/K] and it produced around 1.8 GWth.
Contents

1 Introduction 7
  1.1 Introduction .............................................. 7
  1.2 Nuclear power generation ................................. 7
    1.2.1 Fission .............................................. 8
    1.2.2 Safety .............................................. 9
    1.2.3 Precursors .......................................... 9
  1.3 Molten salt reactor, MSR ................................. 10
    1.3.1 Molten Salt Fast Reactor, the MSFR ................... 11
  1.4 Overview of the project .................................... 12
  1.5 General description of the reactor ....................... 12
    1.5.1 Computational model ................................ 12

2 Theory 15
  2.1 Neutronics ................................................ 15
    2.1.1 General theory ..................................... 15
    2.1.2 Delayed neutrons ................................... 16
    2.1.3 The multigroup approximation ......................... 17
  2.2 Cross-sections and other group constants ................. 18
  2.3 Thermal Hydraulics ....................................... 19
    2.3.1 Temperature ........................................ 19
    2.3.2 Velocity field ...................................... 21
    2.3.3 Turbulence model: \( k-\epsilon \) ...................... 22
    2.3.4 Laws of the wall .................................... 22
  2.4 Interaction between neutronics and thermal hydraulics ... 23

3 Simulating a MSR 25
  3.1 Coupling ................................................. 25
    3.1.1 Introduction ........................................ 25
    3.1.2 Velocity ............................................ 25
    3.1.3 Temperature ........................................ 26
3.1.4 Cross-sections and other group constants . . . . . . . . . 27
3.1.5 Power production . . . . . . . . . . . . . . . . . . . . . . . 28
3.2 Execution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
3.2.1 Initialisation . . . . . . . . . . . . . . . . . . . . . . . . . 28
3.2.2 Looping . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

4 Results 29
4.1 Properties of the reactor . . . . . . . . . . . . . . . . . . . . . . 29
4.1.1 Temperature and velocity fields . . . . . . . . . . . . . . . . 29
4.1.2 Normal operating condition . . . . . . . . . . . . . . . . . . 31
4.1.3 Time step sizes . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.1.4 Reactivity loss due to flow . . . . . . . . . . . . . . . . . . . 35
4.1.5 Power production change due to flow . . . . . . . . . . . . . 37
4.1.6 Temperature feedback coefficient . . . . . . . . . . . . . . . 37
4.2 Transients . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.2.1 Pump failure . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.2.2 Pump startup . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.2.3 Partial pump failure . . . . . . . . . . . . . . . . . . . . . . 40
4.2.4 Heat exchanger failure . . . . . . . . . . . . . . . . . . . . . 40
4.2.5 Partial heat exchanger failure . . . . . . . . . . . . . . . . . 44
4.2.6 Over cooling . . . . . . . . . . . . . . . . . . . . . . . . . . 44
4.2.7 Total failure . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

5 Conclusions and recommendations 51

A Introduction 53

B Code systems used 55
B.1 SCALE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
B.2 Mix . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
B.2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . 56
B.2.2 The in and output . . . . . . . . . . . . . . . . . . . . . . . 56
B.2.3 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . 56
B.3 DALTON . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
B.3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 56
B.3.2 The input and output . . . . . . . . . . . . . . . . . . . . . . 56
B.3.3 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . 58
B.4 HEAT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
B.4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
B.4.2 The input and output . . . . . . . . . . . . . . . . . . . . . . 59
B.4.3 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . 60
C. coupling of the codes
C.1. Introduction .............................................. 63
C.2. Methodology .............................................. 63
Chapter 1

Introduction

1.1 Introduction

With the recent focus on the clean production of energy, alternatives to burning fossil fuels are needed. Whether it is acceptably safe is debatable, but nuclear power production is an option.

The amount of power that can be generated is large compared to other forms of energy generation and there is no production of greenhouse gases. On the other side, the transport of fuel to and from the reactor poses both proliferation and environmental risks. The same is true for the reactor itself, if an accident occurs. The waste from the reactor still contains some usable uranium. In some countries this is extracted and used, but this reprocessing is costly and requires transportation.

The efficiency and safety of nuclear plants have constantly been improved since the introduction of the first nuclear power plant. Figure 1.1 shows an overview of the major steps in power plant technology. These large advancements in technology are used to divide the types of plants into generations. The reactor type examined in this thesis is a generation four design, aimed to further improve safety, sustainability and economics[1]. Note that the reactors in Borssele and Fukushima are of the second generation.

In the following sections (1.2-1.2.3) the physics of nuclear reactors will be introduced, followed by an introduction into the reactor examined.

1.2 Nuclear power generation

The power is produced by fission of certain isotopes. This fission occurs when a fissile isotope, uranium-235 for example, absorbs a passing neutron. During a fission the element falls apart, releasing, on average, 2.4 neutrons and, possi-
Evolution of Nuclear Power

Figure 1.1: Generations of nuclear technology[2]

...bly radioactive, fission products. Because the fission occured as a result of the absorption of a neutron, this is called neutron-induced fission.

The released neutrons can induce other fissions which releases neutrons inducing fission, resulting in a chain reaction. The energy released during a fission is mostly kinetic energy of the fission products. This energy eventually heats a coolant, for example water. The produced steam is used used to power a turbine, generating electrical energy.

1.2.1 Fission

Most of the fissions in the core are neutron induced, although the uranium used does exhibit non-induced fission this does not happen often. When a neutron comes in contact with an isotope, there is a chance that it will be absorbed. This probability is called the absorption cross-section. If the neutron is absorbed it might lead to a fission. This probability that a neutron will induce fission is called the fission cross-section. Its magnitude, determines whether an isotope is classified as fissile, fissionable or non-fissile. For simplicity, the existence of these fissionable elements will be ignored in the rest of this chapter.

A fission also 'creates' neutrons: 2.4 on average. This would result in an increasing number of neutrons in the reactor and an increasing number of fissions, leading to extreme temperatures. This is partially prevented by the non-fissile elements in the core. If they absorb a neutron, it will not lead to a fission. Also neutrons might leak to the surroundings. The average amount of fissions result-
ing from a single fission, or the change of the number of neurons in the core, is called the multiplication factor $k$. This is also referred to as the reactivity $\rho$ which is a scaled version of $k^1$. When $k = 1$ the reaction is stable, since the number of fissions is constant over time, and called critical. If $k$ is below or above one it is referred to as sub or supercritical. The chances of a fission depend on the speed of the neutron, the other influences will be discussed in further sections. In general the chances of fission decrease as its speed increases. To increase the number of fissions the speed of the neutrons has to be decreased. This is done by the moderator. A neutron will collide with the nuclii in the moderator rather than cause a fission. These collisions slow the neutrons down.

### 1.2.2 Safety

It is very important to ensure the reactor does not overheat during an accident. Overheating occurs if the production of heat increases as the temperature rises. So, to prevent overheating of the reactor, the chain reaction should stop when the temperature rises. This is (partially) done by ensuring that the multiplication factor naturally decreases as the temperature increases. The most important contribution to this temperature feedback is the Doppler-effect. The chances of a neutron capture depend not only on the energy of the neutron, which consists of its velocity, but also on the kinetic energy of the element, mostly due to its temperature. In general the chances of capture increase as the temperature rises. This is true for both the fissile an non-fissile isotopes.

To ensure that the production of heat decreases, the absorption of neutrons in non-fissile isotopes must increase faster than the absorption in fissile isotopes. This mechanism is called the Doppler-effect.

This feedback results in an equilibrium power production, the produced power is equal to the removed power. For example, should the cooling fail, the power production will eventually go to zero.

### 1.2.3 Precursors

When fission occurs free neutrons are released and the element falls apart. These fission products are non-fissile so they reduce the reactivity, as they absorb neutrons but do not generate new ones. The products may be radioactive, in which case they will, after some time, decay spontaneously, producing fission products and possibly a neutron. The radioactive fission product is called a

$^1\rho = \frac{k-1}{k} \approx k - 1$ Assuming $k \approx 1$
precursor, the neutron it can release is called a delayed neutron since it is not emitted at the moment of fission. The neutrons emitted immediately are called prompt neutrons.

1.3 Molten salt reactor, MSR

A conventional reactor consists of rods filled with uranium pellets cooled and moderated by water. A problem is that the fission products decrease the reactivity over time, as they absorb neutrons. If the number of neutrons decreases for too long the reaction will die out. A neutron source has to be added to prevent this. This is mostly done by loading the reactor in such a way that its multiplication factor is actually larger than one. The number of neutrons is then regulated by absorbing neutrons using control rods, rods of neutron absorbing material which can be moved into and out of the core.

Another issue is that the fuel rods burn up, these spent rods must then be replaced. When replacing the rods the reaction must be stopped and the core vessel opened.

A molten salt reactor overcomes these hurdles because it uses a fuel that is flowing. The fuel consists of a uranium fluoride salt and is being pumped through the reactor by a pump. This gives the possibility to continuously filter the unwanted elements out of the salt as well as adding new fuel. So there is no need to shut the reactor down to re-fuel it. However, the flow of the fuel introduces a new problem; the precursors. In a conventional reactor they do not move and therefore the delayed neutrons are created at the same place as the original fission. However, as the fuel is flowing in the MSR, these neutrons are released at a different location maybe not even in the core. If the precursor decays outside of the core, the delayed neutron will most likely be lost and the reactivity will decrease.

Apart from the advantages offered by the flowing fuel, the MSR concept has some safety advantages. For example, there is no pressure vessel needed as it operates at atmospheric pressure. Also, if an accident occurs, the liquid state of the fuel allows it to be easily drained from the core. This is done using a ‘freeze plug’, if the fuel gets too warm, the plug melts and the salt flows into storage containers where it poses little danger.

A schematic view of the core used can be seen in Figure 1.2. The yellow represents the salt whereas the black represents the solid parts of the reactor. The pump is located inside the channel.
1.3. Molten salt reactor, MSR

Chapter 1. Introduction

1.3.1 Molten Salt Fast Reactor, the MSFR

The particular reactor examined in this thesis is a non-moderated version of the MSR, utilising the thorium fuel cycle [3].

The fuel consists of a mixture of actinide fluorides, like uranium- or plutonium-fluoride, and thorium fluoride. Thorium, a uranium like element\(^2\), is non-fissile and non-radio-active but by capturing a neutron it eventually forms, the non-natural, fissile U-233. This uranium isotope is very well suited for fission. Thorium is also relatively cheap and easy to work with, as it does not need enrichment.

The most used form of moderating a MSR is to let the salt flow through graphite channels. However, the graphite degrades limiting the total power that can be produced[4]. The channels might become blocked by debris. The moderating structure is not present in the non-moderated reactor, so there is nothing to slow down the neutrons. This reduces the chances of the neutron being captured by a fissile isotope and leading to a fission. The reduced chance of absorption is compensated by an increased amount of fuel to absorb into. Also these fast neutrons have the capability of destroying transuranics; heavy fission products from reactors using the uranium fuel cycle, for example light water reactors.

---

\(^2\)Th contains two less protons compared to uranium and is also of the actinide series
Chapter 1. Introduction

1.4 Overview of the project

The first molten salt reactor was first built in 1970 in Oak-Ridge [5], this was a moderated breeder reactor called the Molten Salt Breeder Reactor, or MSBR. In 2008, E. Merle-Lucotte et al.[6] proposed a non-moderated reactor which was named the non-moderated Thorium Molten Salt Reactor, or TMSR-NM. This reactor was later named Molten Salt Fast Reactor, or MSFR. The main aim of this project is to simulate such a reactor.

The work of M.W. Hoogmoed[7] on the subject is continued. The coupling of the thermal-hydraulics code HEAT and of the neutronics code DALTON was done by him. However, this simulation did not include any precursor flow. Due to the flow of precursors the reactivity in the reactor decreases. Because the precursors 'create' neutrons in a place were they can not contribute to the reactions, therefore these neutrons are 'lost'. The faster the flow, the more precursors travel out of the core. Thus, the flow has an effect on the reactivity in the core. The temperature also has an effect on the reactivity via, among other things, the Doppler effect. This temperature feedback was already included in the simulation however, the flow of the precursors was not. This flow field had to be communicated between the different steps of the simulation.

1.5 General description of the reactor

A schematic cross section of the reactor can be seen in Figure 1.2, a more detailed sketch can be seen in Figure 1.3. The salt circulates, moving upward inside the core and being pushed down by a pump inside the channel. The fuel salt consists of LiF, UF$_4$ and ThF$_4$. With 29.5% heavy nuclei. The core is separated from the channel by a breeder zone. The breeder zone contains LiF and 28% ThF$_4$, the thorium is used to breed U-233. The entire vessel is contained in a reflector which is depicted as the black outline. The reflector consists of NiWCr with molar fractions of 7%, 6% and 87% respectively, this material is used to avoid thermalisation of neutrons.

1.5.1 Computational model

In the model the reactor is assumed to be an axisymmetric cylinder. Therefore, we can restrict our model to a two dimensional representation of the right hand side of the reactor, enforcing no flow of momentum through the center. The reactor used in the calculations is shown in Figure 1.4, which is a simplified version of the one shown in Figure 1.3. The injection and flow of the fertile salt
Figure 1.3: A schematic view of a non-moderated MSR[8]
is neglected. The code does not include material evolution. Also, for simplicity, the layer of B$_4$C was not implemented and neither was the helium bubbling. The helium bubbling serves to remove the gaseous fission products. The pump is modelled by setting the velocity of the salt across the channel halfway up the heat exchanger to the velocity of the pump. Thus, the pump does not induce any turbulence. The heat-exchanger is assumed to have infinite heat capacity, its effects are calculated using $h(T - T_0)$ where $h$ is constant. The temperature of the coolant is assumed to be constant at 600[K]. The heat exchanger is modelled to work only in the channel and inside the breeder zone. The temperature of the coolant seems low, this was needed to extract enough heat from the reactor. The reactor is placed on the ground, with a constant temperature of 800[K], which equals the initial temperature of the reflector, and surrounded by air with a constant temperature of 298[K].

The fuel is at all times assumed to be a homogenous mixture of the isotopes present in the salt. The concentration of these isotopes is constant in time. The same holds for the breeder salt.

In calculating the turbulent flow the k-ε model is used. The turbulence can affect the heat transfer between the fuel, breeder zone and reflector but not the heat extraction by the heat exchanger.
Chapter 2

Theory

2.1 Neutronics

2.1.1 General theory

If we consider an arbitrary volume $V$ the total number of neutrons is the density integrated over the volume. It is however more convenient to use the neutron flux, as this simplifies the calculation of any interaction governed by the cross-section (see section 2.2). The flux is defined as $\phi(\vec{r},t) = vN(\vec{r},t)$, this gives for the time rate of change in volume $V$.

$$\frac{d}{dt} \left[ \int_V \frac{1}{v} \phi(\vec{r},t) \, d^3r \right] = \int_V \frac{1}{v} \frac{d\phi}{dt} \, d^3r$$

(2.1)

production - absorption - net leakage

We define production to be the result of a neutron source $S(\vec{r},t)$. The absorption is defined by $\Sigma_a(\vec{r}) \phi(\vec{r},t)$, see section 2.2. The leakage is the neutron current density $\vec{J}(\vec{r},t)$ in the direction perpendicular to the corresponding surface element. To calculate this for the entire volume (surface of the volume) this must be integrated resulting in a surface integral. To get a volume integral Gauss’s theorem is used:

$$\int_S \vec{J}(\vec{r},t) \cdot d\vec{S} = \int_V \nabla \cdot \vec{J}(\vec{r},t) \, d^3r$$
Chapter 2. Theory 2.1. Neutronics

Substituting all in equation 2.2 results in

$$\int_V \left[ \frac{1}{v} \frac{d\phi}{dt} - S + \Sigma_a \phi + \vec{\nabla} \cdot \vec{J} \right] d^3r = 0 \quad (2.3)$$

The relation between the flux and the current is not known in this case but it is approximated to be similar to that of gaseous diffusion. Thus, Fick’s law with diffusion coefficient $D(\vec{r})$, is used:

$$\vec{J}(\vec{r},t) \approx -D(\vec{r}) \vec{\nabla} \phi(\vec{r},t).$$

Also the balance equation 2.3 must hold for any volume, thus the integrand must vanish, leading to:

$$\frac{1}{v} \frac{d\phi}{dt} = S - \Sigma_a \phi + \vec{\nabla} \cdot D(\vec{r}) \vec{\nabla} \phi \quad (2.4)$$

The above assumes that all the neutrons have the same speed and thus the same energy. This model describes the neutronics nicely but is inadequate for the reactor used, as it assumes all the neutrons to have the same speed. Because neutrons can gain or lose energy by interacting with matter this has to be modelled and added to the balance equation of the one-speed approximation (equation 2.4). Also, all the quantities used in that equation are energy-specific. Including this energy dependance results in

$$\frac{1}{v} \frac{d\phi(\vec{r},E,t)}{dt} = S(\vec{r},E,t) + \int_0^\infty \Sigma_s(E' \rightarrow E) \phi(\vec{r},E',t) dE' - \Sigma_{\alpha} \phi + \vec{\nabla} \cdot D(\vec{r},E) \vec{\nabla} \phi - \Sigma_{\tau} \tau(\vec{r},E) \phi(\vec{r},E,t) \quad (2.5)$$

The source term consists of neutrons emitted during fissions:

$$S(\vec{r},E,t) = \chi(E) \int_0^\infty \nu(E') \Sigma_f(E') \phi(\vec{r},E',t) dE' \quad (2.6)$$

Here $\nu$ is the total number of neutrons created by a single fission and $\chi(E)$ the energy spectrum of the released neutrons. This equation assumes that all neutrons are released instantaneously; the are all prompt neutrons. This is the case for most neutrons, but about 1% of the total number of neutrons is released after some time. These are called delayed neutrons.

2.1.2 Delayed neutrons

During a fission, fission products are produced. These products can also release a neutron by natural decay. This occurs after some time in which the precursor has moved with the fuel. The precursors are customarily divided into six dif-
2.1. Neutronics Chapter 2. Theory

<table>
<thead>
<tr>
<th>single-speed</th>
<th>multi-speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{v} \frac{d\phi}{dt}$</td>
<td>$\frac{1}{v_g} \frac{d\phi_g}{dt}$</td>
</tr>
<tr>
<td>$S(\vec{r},E',t)$</td>
<td>$S_g$</td>
</tr>
<tr>
<td>$\Sigma^a\phi$</td>
<td>$\Sigma_g^a\phi_g$</td>
</tr>
<tr>
<td>$\vec{v} \cdot D(\vec{r},E)\vec{\nabla}\phi$</td>
<td>$\nabla \cdot D_g\nabla \phi_g$</td>
</tr>
<tr>
<td>$\Sigma^s\phi_g(\vec{r},E,t)$</td>
<td>$\sum_{g'} \Sigma_{gg'}^s\phi_{g'} = \Sigma_g^s\phi_g$</td>
</tr>
</tbody>
</table>

Table 2.1: Terms from equation 2.5 with their multigroup equivalent and meaning

Different groups depending on their decay time. The fraction of the total neutron production over the precursor neutrons belonging to group $i$ is denoted by $\beta_i$. While the total fraction of precursor neutrons is denoted by $\beta$. The decay constant and concentration are given by $\lambda_i$ and $C_i$ respectively. Thus equation 2.6 becomes

$$S(\vec{r},E,t) = \chi_p(E) \int_0^\infty (1 - \beta) \nu(E')\Sigma^f(E')\phi(\vec{r},E',t)\,dE' + \sum_i \chi_d(E)\lambda_iC_i(\vec{r})$$

(2.7)

Also the concentration of precursor neutrons must now be described.

$$\frac{DC_i}{Dt} = \text{production - decay}$$

(2.8)

$$\frac{dC_i}{dt} + \nabla uC_i(\vec{r}) = \int_0^\infty \beta_i\nu(E')\Sigma^f(E')\phi(\vec{r},E',t)\,dE' - \lambda_iC_i(\vec{r})$$

(2.9)

2.1.3 The multigroup approximation

To approximate equations 2.4 and 2.9 the released neutrons are divided into energy groups. By defining $G$ energy groups denoted by subscript $g$. So for the flux this implies:

$$\phi_g(\vec{r},t) \equiv \int_0^{E_g} \phi(\vec{r},E,t)\,dE.$$

(2.10)

The terms in equation 2.5 are now rewritten as follows.

In the terms considering only one single energy a subscript for the group is added, see table 2.1.

And, for the inscatter

$$\int_0^\infty \Sigma^s(E' \rightarrow E)\phi(\vec{r},E',t)\,dE' \text{ becomes } \sum_{g'=1}^G \Sigma_{gg'}^s\phi_{g'}$$
Chapter 2. Theory 2.2. Cross-sections and other group constants

The constants per group: the diffusion constant \( D_g \) and the cross sections \( \Sigma_g \) are the flux-weighted average of the constant over that group (see section 2.2). Resulting in the muligroup diffusion equation:

\[
\frac{1}{v_g} \frac{d\phi_g}{dt} = \nabla \cdot D_g \nabla \phi_g + S_g - \Sigma_g^a \phi_g - \Sigma_g^s \phi_g + \sum_{g'} \Sigma_{g'g}^s \phi_{g'}
\] (2.11)

Also, the fission source (equation 2.7) now becomes

\[
S_g = \sum_{g'} \chi_{pg'} (1 - \beta) \nu_{g'} \Sigma_{g'}^f \phi_{g'} + \sum_i \chi_{dg} \lambda_i C_i
\] (2.12)

with equation 2.9 becoming

\[
\frac{dC_i}{dt} = \sum_g \beta_i \nu_g \Sigma_g^f \phi_g - \lambda_i C_i - \nabla u C_i
\] (2.13)

The constants \( \nu_g \) and \( \chi_x \) are also referred to as group constants.

2.2 Cross-sections and other group constants

The cross sections define the chance per unit distance that an neutron interacts with a proton. These depend on temperature. The group constants are averaged over the energies in the group. For example the fission cross section:

\[
\Sigma_g^f \equiv \frac{1}{\phi_g} \int_g^{g-1} \Sigma^f (E) \phi(\vec{r}, E, t) dE.
\]

This expression also contains the flux, which we eventually want to calculate. The group constants are calculated as follows. The constants are known for very small group sizes. But for larger groups the appropriate constants must be calculated using the unknown flux. The constants are calculated using the following steps:

1. the time and space independent flux is calculated using the finely structured group constants and simple geometry.

2. this flux is then used to calculate the group constants of the coarser groups

It is too time consuming to do this every time the flux is calculated. Therefore a library of group constants is created for several temperatures. Using the temperature of the core these are then interpolated to the actual temperature of the core.
2.3 Thermal Hydraulics

First of all it is important to know if the flow in the reactor is turbulent or laminar. A good indication for this is the Reynolds number which is given by

\[ \text{Re} = \frac{\rho v D}{\mu} \]

Using \( \rho = 3000 \text{[kg m}^{-3}\text{]} \), approximately the density of the salt, \( D = 0.25 \text{[m]} \), the diameter of the channel\(^1\) and \( \mu = 4 \cdot 10^{-3} \text{[Pa s]} \) this gives

\[ \text{Re} \approx 1.9 \cdot 10^5 \cdot v \]

And with the turbulent region starting at around \( 10^4 \), the system will most likely be turbulent.

2.3.1 Temperature

The most important quantity in a reactor is its temperature. This affects both the production of power and its criticality. Beginning with the most general case, the balance equation for temperature is given by\(^{[10]}\)

\[ \rho c_p \frac{dT}{dt} + \rho c_p v \nabla T = -\nabla q^{tot} + S + L \] \hspace{1cm} (2.14)

or

\[ \rho c_p \left( \frac{dT}{dt} + v \nabla T \right) = -\nabla q^{tot} + S + L. \] \hspace{1cm} (2.15)

Equation 2.15 can be further specified for different regions of the reactor. These different regions are: where salt is present, the breeder zone and the reflector. A somewhat more detailed cross section of half a reactor is shown in Figure 2.1.

In the core, channel and plena salt is present so the heat transfer is the gradient of the temperature times the transfer coefficient. The loss term is the heat extracted by the heat exchanger, this is assumed to extract heat from the entire channel. The source is the production of heat due to fission. Thus equation 2.15 can be written as

\[ \rho c_p \left( \frac{dT}{dt} + v \nabla T \right) = -\lambda_{\text{fuel}} \nabla T - h (T - T_0) + S. \] \hspace{1cm} (2.16)

With \( \lambda_{\text{fuel}} \) and \( h \) the heat transfer coefficient of the fuel and the heat exchanger respectively, note that \( h \) is zero outside of the channel, \( T_0 = 600 \text{[K]} \) and \( S \) the

\(^1\) the distance between the breeder and the side reflector is used
source term.

For the other regions there is no flow. In the reflector there is no heat exchanger, heat is only lost due to radiation and free convection to the surrounding air. In the breeder heat is assumed to be removed in the same way as in the heat exchanger, inside the breeder zone there is some power production due to fission. Equation 2.15 can, for the breeder zone, be written as

\[
\rho_c p \frac{dT}{dt} = -\lambda_b \nabla T - h (T - T_0) + S
\]  

(2.17)

resulting in

\[
\rho_c p \frac{dT_r}{dt} = -\lambda_r \nabla T + \frac{Nu \lambda_{air}}{D} (T_r - T_0)
\]  

(2.19)

The heat transfer at the interfaces salt-breeder and salt-reflector are governed by the 'law of the wall', if applicable. This method of calculating the heat transfer applies to turbulent flows. More information about the type of flow will be provided in further parts of this section. To determine whether this law should be used the dimensionless wall coordinate \( y^+ \) should be larger than 11.
2.3. Thermal Hydraulics

[10]. this coordinate is defined as:

\[ y^+ = \Delta \frac{\rho \mu \tau}{\mu_l} \]  
(2.20)

where

\[ u_{\tau} = k^{1/2} C_\mu^{1/4} \]  
(2.21)

Where \( \Delta \) is the distance to the wall. If the law of the wall is used, the heat transfer coefficient is given by

\[ \lambda_{\text{wall}} = \frac{u_{\tau} \kappa |u_{\parallel}|}{\log (E \cdot y^+)} \left[ \frac{c_p \mu_t}{P_r \mu_l} \right]^{1/2} \]  
(2.22)

Where \( u_{\parallel} \) is the velocity parallel to the wall in question. Otherwise the heat transfer coefficient of the fluid is used directly.

2.3.2 Velocity field

The velocity field is in principle defined by the Navier-Stokes equation[10]:

\[ \rho \frac{Dv}{Dt} = -\nabla p + \nabla T + f \]  
(2.23)

or

\[ \rho \left( \frac{d\vec{v}}{dt} + \vec{v} \nabla \vec{v} \right) = -\nabla p + \nabla T + f \]  
(2.24)

In this case the density is assumed to be constant. So the continuity equation

\[ \rho \nabla \vec{v} = -\frac{dp}{dt} \]  
(2.25)

becomes

\[ \rho \nabla \vec{v} = 0. \]  
(2.26)

The only body force on the salt is gravity, so \( f = \rho \vec{g} \) hence the Navier-Stokes equation (2.24) becomes

\[ \rho \frac{d\vec{v}}{dt} = -\nabla p + \nabla T + \rho \vec{g} \]  
(2.27)
with
\[ T = \mu \left[ \nabla \vec{v} + (\nabla \vec{v})^T - \frac{2}{3} \nabla \vec{v} I \right] \] (2.28)

The viscosity \( \mu \) is the effective viscosity, in this turbulent case the sum of the laminar and turbulent viscosity:

\[ \mu = \mu_l + \mu_t \]

### 2.3.3 Turbulence model: \( k - \epsilon \)

As stated before, the flow in the reactor is turbulent. To model this turbulence the widely used \( k - \epsilon \) model is applied\[10\]. This model works by solving two extra sets of equations: one for the existence and one for the dissipation of turbulent kinetic energy. The turbulent kinetic energy, \( k \), is the energy associated with eddies in the flow. Its dissipation, \( \epsilon \), is mainly due to the internal friction resisting deformation by these eddies. Although this model has its shortcomings, it is assumed to give realistic values for the flow field. Especially as the inaccuracies of the other parts of the code will outweigh those introduced by the \( k - \epsilon \) model. In this model the turbulent viscosity is defined as

\[ \mu_t = \rho C_{\mu} \frac{k^2}{\epsilon} \] (2.29)

where \( C_{\mu} \) is an empirical constant. The balance equation for \( k \) is given by

\[ \frac{Dk}{Dt} = \nabla \cdot \left( \left[ \frac{\mu_t}{\sigma_k} + \mu_l \right] \nabla k \right) + G_k - \epsilon \] (2.30)

And for \( \epsilon \):

\[ \frac{D\epsilon}{Dt} = \nabla \cdot \left( \frac{\mu_t}{\sigma_\epsilon} \nabla \epsilon \right) + \frac{\epsilon}{k} (C_{\epsilon 1} G_k + C_{\epsilon 2} \epsilon) \] (2.31)

In equations 2.31 and 2.30 \( G_k \) is the production due to sheer stress while the influence of buoyancy is neglected. The \( k - \epsilon \) model is only valid for high Reynolds numbers. So the model is not applicable at the wall, assuming no-slip conditions. Here the solutions are approximated by wall-functions

### 2.3.4 Laws of the wall

These wall functions are semi-empirical equations used near the walls in order to avoid singularities. The average velocity at the point next to the wall, at
2.4 Interaction between neutronics and thermal hydraulics

As mentioned in the introduction, the cross-sections of materials change when their temperature changes. If the cross sections change, the amount of fissions and the reactivity change as well. This results in a change in power production. Which, in turn, changes the temperature of the materials. So, the temperature influences the power production which in turn influences the temperature.

In the case of a liquid fuel reactor the velocity of the fuel is also important as it influences the reactivity. A flow of fuel causes the precursors to flow with the fuel. This displacement may cause the precursors to decay outside the core even though the fission was inside. This results in a loss of the precursor neutron and decreases the multiplication factor. Thus, the production of power decreases as the speed of the flow increases. Aside from influencing the power production, the flow also influences the temperature. The removal of heat depends mostly on the convective transport of heat. A higher velocity results in a more efficient heat removal and thus in a lower temperature. The temperature also influences the flow as the material properties of the fuel, such as the viscosity, depend on it. So, the velocities influence the power production and the temperature. The temperature also impacts the velocities.

For a diagram of all these dependancies, see Figure 2.2.
Chapter 2. Theory 2.4. Interaction between neutronics and thermal hydraulics

Figure 2.2: Interaction of the quantities involved
Chapter 3

Simulating a MSR

3.1 Coupling

3.1.1 Introduction

The aim of this project evaluate the safety and performance of a MSR. The reactor is simulated by coupling neutronics to thermal hydraulics code, since they affect each other, as explained in section 2.4. The programs used in this coupling are:

- DALTON, which is used to calculate the neutronics of the system.
- HEAT, which is used to calculate the thermal-hydraulics involved.
- MIX, which is used to calculate the group constants as a function of temperature.

The temperature and the velocities in the reactor are calculated by HEAT. The velocities are passed to DALTON while the temperatures are passed to MIX. MIX calculates the appropriate group constants and passes these to DALTON. DALTON then uses these group constants together with the velocity field to calculate a power production. This production together with the velocities and temperatures of the previous step are used by HEAT to calculate new profiles for the temperature and velocity. In short the coupling links DALTON to HEAT and HEAT via MIX to DALTON. A schematic view of the coupling can be seen in figure 3.1.

3.1.2 Velocity

The velocity field is calculated by HEAT. However, it does not create an output in the format used by DALTON. HEAT outputs the velocities on the interface.
between two cells while DALTON needs the flow out of the element. DALTON calculates the corresponding surface areas.

Dalton turned out to be very slow when presented with the chosen gridsizes, so the grid was scaled down by a factor 4, joining 4 cells from, for example, (1,1) to (2,2) into one. The outflow velocities to the neighboring large cells is then calculated by weighing the individual outflow velocities of the cells with their respective areas.

This speeds up DALTON significantly, a factor 10 to 15, and enables HEAT to use the desired gridsize. This reduction of the number of elements also serves to make convergence by DALTON possible. Between cells a small amount of mass may be lost or gained due to numeric inaccuracies in the calculations. If these discrepancies become too large, DALTON can not calculate a solution. Joining four cells into one reduces these mass errors. When even more elements are used, even the joining of four cells was inadequate. Thus, an even coarser grid had to be used. The flow was calculated for elements similar to those used for communicating the temperatures, see figure 3.2. These different averaging schemes resulted in differences of a few kelvin in equilibrium temperature.

### 3.1.3 Temperature

The temperatures are calculated by HEAT. The temperatures are calculated using the same grid size as used for the velocities per element and then averaged over certain zones. These temperatures per zone are passed to MIX, which
3.1. Coupling

3.1.4 Cross-sections and other group constants

The temperatures per zone from the output of HEAT, are used by MIX to estimate group constants. At the start of the simulation a library of group constants at different temperatures was generated by SCALE. These values were generated from 300K to 2100K in steps of 300K. Using the current temperature MIX
Chapter 3. Simulating a MSR

3.2. Execution

uses linear interpolation to find the group constant at the current temperature. The code assumes that the mixture is homogeneous and that the concentration of the different isotopes is constant.

3.1.5 Power production

The constants calculated by MIX are used by DALTON to solve the multi-group diffusion equation (equation 2.11). This results in a value for $k_{\text{eff}}$ and a power profile. DALTON scales the profile to sum to the power production as set in the input. The multiplication factor is used to scale the power production for the next loop:

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

3.2 Execution

3.2.1 Initialisation

First of all, a work library for MIX is created using SCALE. To get a first estimate of the conditions in the reactor, the initial temperatures inside the reactor (these are set in MAININPUT) are used by MIX to calculate the cross sections. DALTON is then used to calculate the flux, using the initial power production. Since HEAT has not yet been run, the velocity profile is still unknown. Therefore DALTON calculates the flux for stationary fuel.

3.2.2 Looping

Now that everything has an initial value, the quantities described above are calculated. Using a fine grid, the set pump velocity and the power production profile, HEAT calculates a temperature and velocity profile. The temperatures and velocities are then averaged to their respective zonal profiles which can be seen in Figures 3.3 and 3.2. creating input for other programs. It does save the full profiles for its own calculations at the next timestep. The temperatures calculated are used by MIX to interpolate the group constants.

These constants together with the velocity profile are then used by DALTON to calculate a new power production profile and $k$-value. The production is averaged over zones and passed to HEAT, the $k$-value is used to calculate the value of the production for the next loop. This concludes the first loop.
Chapter 4

Results

In this section the final results of the project are presented. In Section 4.1 the normal operating of the reactor is examined. Here the effect of the pump speed is investigated. In Section 4.2, the safety of the reactor is examined by simulating some transients. A start and stop of the pump is simulated, as well as a failure of the heat exchanger.

4.1 Properties of the reactor

4.1.1 Temperature and velocity fields

Figures 4.1 and 4.2 show the temperature and velocity fields respectively, at normal operating conditions. These Figures were made after the reactor had reached its equilibrium. The velocities were calculated on the interfaces between elements; the depicted velocities are a linear interpolation between the fuel velocities on opposing faces. To show the influence of the flow on the temperature, the velocities (using only arrows) has been plotted on top of the temperature profile in Figure 4.1. In Figure 4.2, the velocities are indicated by the size and direction of the arrows, the arrow-density has no connection to the flow speed. Because the speed is poorly represented by the size of the arrows it is also indicated by the colours in the plot. The breeder zone and the reflectors can be recognised by their (semi)uniform temperature and the absence of any flow. To the left of the breeder zone an extremely hot region is located; the temperature of these elements far exceeds 1400 [K] but this has been altered for visibility reasons. The alteration was done by setting every value higher than 1400 to 1400, the highest temperature was approximately 2300 [K]. The existence of this zone in the actual reactor is unlikely. These particular elements are hot because there is barely any flow at their location. Thus, heat removal from these
Figure 4.1: Temperature field (colours) with velocities (arrows) of the reactor at normal operating conditions
elements is slow. More flow near the breeder would be expected. This flow would be mainly due to sheer forces. However, the sheer stress was neglected in HEAT. Also, a hot spot would result in very few fissions at that point, due to the expansion of the fuel and the temperature feedback. This is not modelled since the temperatures are averaged per zone. It should also be noted that the 2300[K] is far above the boiling point of the fuel. Also note that the expansion of the fuel is used when calculating the cross sections but it is not used when calculating the flow.

Inside the vortices the temperature is high compared to the surrounding area as the salt here takes more time to reach the heat exchanger. Figure 4.2 shows even more clearly that there is no flow at the extremely hot elements next to the breeder zone. The flow is also less present at the corners of the reactor, this is due to the model and in reality less true. At the right hand side of the reactor\(^1\) there are in and outlets of the salt. These probably cause more radial velocity in the core, also these in and outlets are not axisymmetric.

Apart from this, the model assumes that there is no flow through the central plane, which is not necessarily true. Also the turbulence model is not very accurate since the sheer forces are neglected. Although in the core shear forces play a small role, this might be influential in transferring the upward momentum, causing more flow in the corners of the reactor.

The sheer stress term in equation 2.30 had to be neglected otherwise HEAT produced large errors it was thought to be of significance at the edges of the breeder and not the centre of the core. The temperature mentioned in the graphs is the volume-weighted average of the temperature of the elements in the core and the areas above and below it. Thus, in the radial direction everything until the breeder, in the vertical direction everything between the bottom and top reflectors.

### 4.1.2 Normal operating condition

A static equilibrium state exists if the temperature feedback coefficient is negative. In such a steady state, the temperature must be constant. As the power production depends on the temperature, a steady state implies a constant power production. This is constant if \( k = 1 \). Due to the different locations of the core and heat exchanger, there is some lag between the production and the extraction of heat.

In theory, equilibrium is reached as \( k = 1 \). But in the actual simulation, equilibrium is defined using equation 4.1 with \( \epsilon = 10^{-7} \).

\(^1\)as shown in the pictures
Chapter 4. Results

4.1. Properties of the reactor

Figure 4.2: Velocity field of the reactor at normal operating conditions, arrows indicating direction and colour indicating the amplitude of the velocity.
4.1 Properties of the reactor

\[ |k_{\text{eff}}^i - 1| < \epsilon \text{ and } |k_{\text{eff}}^i - k_{\text{eff}}^{i-1}| < \epsilon \quad (4.1) \]

First of all a plot of the simulation from start to equilibrium can be seen in Figure 4.3. Parameters used in all simulations if not stated otherwise are given in Table 4.2.

The reactor is started with its various components (salt, breeder and reflector) at a uniform temperature. The oscillations seen represent the process of iterating towards the correct temperature profile and are partially due to the delay between the production and extraction of power. The discontinuities that can be seen between 50 and 100s are most likely minor errors in the program due to the emergence of turbulence. In these steps HEAT was not able to reach convergence. However, after this period the line becomes smooth again and an equilibrium is reached. The existence of these discontinuities raises questions about the correctness of the equilibrium and should be investigated further.

The calculations by DALTON took quite some time, especially the flow field was included. Thus, to speed up the simulation, the flowfield is neglected by DALTON until the simulation nears equilibrium. At this point, after 289 seconds in this case, DALTON includes the effects of the flow in its calculation. This creates a jump in \( k \)-values resulting in a discontinuity in the power production, see Section 4.1.4. Note that in Figure 4.3, the power extracted is actually more than the power produced. This difference is constant during a simulation at 3\% of the power produced for the chosen gridsize, the relative size of this error is constant; if the power production goes to zero as a result of some transients, this error also approaches zero. The error increases rapidly as the number of elements increase, see Table 4.1 for the actual values.

As DALTON gives slightly different values for \( k_{\text{eff}} \) when the velocities are averaged differently, all results were obtained using the zonal averaging. See section 3.1.2.

<table>
<thead>
<tr>
<th>elements</th>
<th>error/production</th>
</tr>
</thead>
<tbody>
<tr>
<td>1536</td>
<td>2%</td>
</tr>
<tr>
<td>2400</td>
<td>3%</td>
</tr>
<tr>
<td>3456</td>
<td>4%</td>
</tr>
<tr>
<td>4704</td>
<td>6%</td>
</tr>
</tbody>
</table>

Table 4.1: The error in HEAT of different grids. The error in this shown here is defined as error=|production-extraction-power absorbed-power leaked to the environment|
Figure 4.3: A view of a simulation to equilibrium, top left a graph of $k_{\text{eff}}$ vs time, top right temperature and on the bottom the power production (blue) and removal (red). The point where the flow is included in the calculation by DALTON is indicated by a $\Rightarrow$, the step in multiplication factor is clearly visible.
4.1.3 Timetep sizes

In the simulation, the calculations done by DALTON and MIX are independent of time. During the execution of HEAT the time advances. This time between DALTON runs was varied. The results for three different times are plotted in Figure 4.4. As can be expected, the results converge with different speeds but towards the same point. The time between runs was 0.5, 1 and 2 seconds, these times correspond to the red, blue and black lines, respectively. It can be seen that the final result, as well as the (real) time to convergence does not differ between timesteps. So, at least for the simulation towards equilibrium it would suffice to use a timestep of 2 [s].

Also, the timestep used by HEAT was varied. Timesteps characterised by \( n = 0.5, 1 \) and 2 were used. The value for the timestep size in HEAT in the table corresponds to \( n = 1 \). Here \( n \) is the number of elements travelled inside the channel per timestep. The full simulations for these timesteps can be seen Figure 4.5. All timesteps produce almost identical results, so a timestep of \( n = 2 \) would have produced equally good results.

4.1.4 Reactivity loss due to flow

The initial power production change due to flow was estimated. This was done by setting a pump speed and calculating the velocity profile. This profile was passed to DALTON, which then calculated a value for \( k_{\text{eff}} \), this was plotted against the velocity of the pump. During these calculations the cross section set was constant.

As mentioned in Section 4.1.2, DALTON, HEAT and in this case MIX were run in a loop until convergence criteria (see equation 4.1) were met. In this case \( \epsilon = 10^{-5} \) was used. During these initial loops DALTON ignored both the flowfield and the loss of reactivity. As the above limit was reached DALTON included the flowfield in its calculations resulting in a sudden change in reactivity. The sizes of these steps were compared to the initial finds, this is shown in

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>gridsizes</td>
<td>40:60</td>
</tr>
<tr>
<td>pump speed</td>
<td>2 m/s</td>
</tr>
<tr>
<td>timestep size in HEAT</td>
<td>0.045 s</td>
</tr>
<tr>
<td>heat transfer coefficient</td>
<td>900 kW m^{-3}K^{-1}</td>
</tr>
<tr>
<td>time between Dalton runs</td>
<td>1 s</td>
</tr>
<tr>
<td>initial temperature of the reflectors and breeder</td>
<td>800 K</td>
</tr>
<tr>
<td>initial temperature of the salt</td>
<td>1000 K</td>
</tr>
</tbody>
</table>

Table 4.2: Values of parameters used in all simulations, unless stated otherwise
Figure 4.4: $k_{\text{eff}}$, Temperature, power production and extraction at different timesteps: time between DALTON calculations. For every timestep the same equilibrium is reached.

Figure 4.5: $k_{\text{eff}}$, Temperature, power production and extraction at different timesteps within HEAT. Except for the the region between 50 and 100 seconds the results are similar if not identical.
Due to the flow the reactivity decreases up to a certain maximum when the precursors start to flow back into the core. The maximum loss is around 120 [pcm] which is in agreement with the ratio between the volumes of the core and the channel. The channel is about half the size of the core and, for U-235, 0.7% of neutrons are due to a precursor, this gives a maximum loss of 700 \( \times \frac{1}{3} \approx 233 \text{pcm} \). This maximum is not reached as precursors can flow back into the core before they decay. Also plena are neglected in this estimation.

In Figure 4.6 it can be seen that the values corresponding to more lengthy simulations show a similar form and a similar maximum but different values. The shape is approximately the same as the precursors are lost in the same manner, the rate at which they flow out of the core is approximately the same. However, the temperatures of the reactor and the concentrations differ. For example, at the bottom of the core the temperature is lower than at the top. And more fissions result in more production of precursors. Resulting in more precursors decaying before entering the channel. This effect is stronger as the velocity increases.

### 4.1.5 Power production change due to flow

The flow of the fuel leads to a decrease in reactivity as some precursors decay outside the core. This lowers the value of \( k \). However, the value of \( k \) must reach one for a stable operation. So the value of \( k \) must increase which occurs for example when the temperature decreases. This decrease in equilibrium temperature results in less heat extraction. On the other hand, an increased velocity enables the extraction of more power as the mass flow through the heat exchanger increases, enabling the extraction of more power.

Figure 4.7 shows the change in equilibrium temperature and power extraction due to an increase of flow velocity.

### 4.1.6 Temperature feedback coefficient

Another important safety characteristic of a reactor is the temperature feedback. This was estimated as follows. First a uniform temperature was set. Then the corresponding cross sections and velocity profile were calculated and passed to DALTON. DALTON then calculated a corresponding value for \( k \). For the next point the temperature of the salt is changed and the calculation is redone. Then \( k \) as a function of temperature is plotted and the feedback coefficient defined as[9]

\[
\alpha = \frac{dp}{dT} = \frac{1}{k^2} \frac{dk}{dT}
\]
Chapter 4. Results

4.1. Properties of the reactor

Figure 4.6: Reactivity loss due to fuel flow, the initial estimation (red) and the final calculation (blue).

Figure 4.7: Power and temperature versus speed of the pump. The temperature is quite constant however, the power production almost doubles as the velocity increases.
is calculated around \( k = 1 \) using

\[
\alpha = \frac{k_{i+1} - k_{i-1}}{T_{i+1} - T_{i-1}} \text{ where } k_i \approx 1
\]

The temperature does have an effect on the cross-sections but not on the flow profile. The results have been plotted in Figure 4.8.

The stable point, \( k = 1 \), is reached at approximately \( T = 925[K] \). The datapoints at \( T = 901[K] \) and \( T = 951[K] \) were used to calculate the derivative. This resulted in

\[
\frac{dk}{dT} = -5.974[\text{pcm K}^{-1}]
\]

which is first of all negative resulting in an equilibrium. For a non-moderated reactor Mathieu et al.[4] found \( \alpha \approx -5.5[\text{pcm/K}] \).

Also the temperature at equilibrium reached in Figure 4.3 is approximately the same as the temperature resulting in \( k = 1 \) in the run depicted in Figure 4.8.

### 4.2 Transients

Below some transients are simulated. All simulations are started with the reactor in its equilibrium state. When starting the transient equilibrium is defined using \( \epsilon = 10^{-6} \) in equation 4.1. The simulation is stopped as the original convergence criteria are met or when the simulation ran for 2000 seconds after starting the transient. But most figures show only the first 500 seconds, as a new equilibrium was reached at that time.

As the power production of the reactor goes to zero, the error in the thermal hydraulics calculations also decreases to around zero.

The time constant \( \tau = 60[\text{s}] \) during all transients.

It should be noted that there is no density-change driven circulation in the model. Also, heat production due to the decay of the fission products in no included in the model.

#### 4.2.1 Pump failure

This transient simulates the failure of the pump which moves the fuel. If this pump fails, precursor neutrons are no longer lost, which increases the production of neutrons in the core. Due to the failure of the pump and the stagnation of the flow, heat can no longer be removed by convection. This dramatically slows down the heat removal. This causes the temperature to rise sharply. This, due to the temperature feedback, results in a much lower reactivity. The production will die out swiftly due to the high temperature. Eventually, it can be
expected that the reactor eventually reaches a stable state with very little power production.
The speed of the pump after the start of the transient is defined by

\[ v(t) = v_0 e^{-\frac{(t-t_0)}{\tau}}. \]

Where \( v_0 \) is the initial pump speed of 2 \([m/s]\). The results of this simulation can be seen in Figure 4.9. As expected, the production of power decreases.

4.2.2 Pump startup

The start of the pump is expected to increase heat removal due to convective transfer. This would result in a drop in temperature and eventually a higher power production. Also, the increased flow results in the loss of precursors, resulting in a lower equilibrium temperature.

First the system was brought to a stable state using a low flow speed. The transient is started by increasing the velocity using the cosine in equation 4.2.

\[ v_{\text{pump}}(t) = v_0 + \frac{v_\infty - v_0}{2} \left( \cos \left( \frac{(t - \tau) \pi}{\tau} \right) + 1 \right), \quad t < \tau \quad (4.2) \]

With \( v_\infty \) the target speed of the pump: 2 \([m/s]\). At the start of this transient the timestep used by HEAT is adapted to the final pump speed. The results of this simulation can be seen in Figure 4.10. As expected the power production increases while the temperature drops slightly.

4.2.3 Partial pump failure

A partial loss of power to the pump is simulated using equation 4.2 with \( v_\infty = \frac{v_0}{2} \). Also, this transient shows that the reactor does reach a new equilibrium with less power produced. And, as less precursors are lost, the equilibrium temperature rises. This can be seen in Figure 4.11.

4.2.4 Heat exchanger failure

This transient simulates the loss of cooling, caused by a failure of the pump in the cooling system. It is started as the reactor is in a stable state. The failure of the heat exchanger is simulated by exponentially decreasing its transfer coefficient:

\[ h(t) = h_0 \cdot e^{-\frac{(t-t_0)}{\tau}}, \]
4.2. Transients

Figure 4.8: $k_{\text{eff}}$ Values vs. temperature, the circles represent the datapoints whereas the red line represents the fit through the points around $T=925[K]$.

Figure 4.9: Left to right, top to bottom: $k_{\text{eff}}$, $T$, power production (blue) and removal (red) and fuel velocity during a pump failure. As the pump stops, the temperature rises. This causes $k_{\text{eff}}$ and the production of fission power to decrease.
Eventually a new equilibrium is established, with a lower temperature to compensate for the loss of precursors.
4.2. Transients

Chapter 4. Results

Figure 4.11: Left to right, top to bottom: $k_{eff}$, $T_s$, power production (blue) and removal (red) and fuel velocity as the pump slows down. A new equilibrium is established with a higher temperature as there is less loss of precursors to compensate for.
where $h_0$ is the initial transfer coefficient.

If the heat exchanger fails, the temperature will rise. This results in less reactions inside the core leading to less power production due to fission. The production decreases to almost zero as there is no heat removed. No significant removal, heat is still lost to the surrounding air. The decay heat is not included here.

This transient is started after a few seconds, the result of the simulation can be seen in Figure 4.12. Due to the sharp rise in temperature, the $k_{\text{eff}}$ drops and so does the power production. The reactor loses less and less heat via the heat exchanger, so the cooling becomes very slow. However, the power production goes to zero so the reactor is not heated up and the reactor will slowly cool down further.

### 4.2.5 Partial heat exchanger failure

A partially congested or damaged pipe can cause partial loss of cooling. As expected the reactor finds a new equilibrium with a lower power production. The transient was simulated by diminishing the heat transfer coefficient using

\[
h(t) = h_0 + \frac{h_\infty - h_0}{2} \left( \cos \left( \frac{(t - \tau) \pi}{\tau} \right) + 1 \right), \quad t < \tau
\]

with \(h_\infty = 0.9h_0\). The results are shown in Figure 4.13. This graph does show a new equilibrium at a low power production.

### 4.2.6 Over cooling

If the reactor is suddenly cooled, the temperature drops, causing the reactivity to increase. The results of this simulation can be seen in Figure 4.14. As the reactivity increases, so does power production and subsequently the temperature, lowering the reactivity. Eventually a new equilibrium will be established. The temperature should return to the old level as the there is little extra loss of precursors to compensate for. The production of power increases so the density of precursors in the fuel increases, resulting in a slight increase of precursors being lost. The power extraction is expected to increase.

The overcooling is simulated similar to the pump startup:

\[
h(t) = h_0 + \frac{h_\infty - h_0}{2} \left( \cos \left( \frac{(t - \tau) \pi}{\tau} \right) + 1 \right), \quad t < \tau
\]

With \(h_\infty = 2 \cdot h_0\) and \(\tau = 60\) s. 44
Figure 4.12: Top left a graph of $k_{eff}$ vs time, bottom left temperature and right the power production (blue) and removal (red) as the heat exchanger stops working. As no heat is extracted, the temperature rises, lowering the production of power.
4.2. Transients

Figure 4.13: Top left a graph of $k_{\text{eff}}$ vs time, bottom left temperature and right the power production (blue) and removal (red) as the heat exchanger fails partially. As the temperature rises the reactor becomes subcritical until the temperature has gone down sufficiently. After 400 seconds a new equilibrium is established, the temperature reaches the same value as the loss of precursors does not change.
Figure 4.14: Top left a graph of $k_{eff}$ vs time, bottom left temperature and right the power production (blue) and removal (red) as the heat transfer coefficient doubles. The reactor can produce more power as more power is extracted. The equilibrium temperature is the same as before the transient because the loss of precursors is (approximately) constant.
4.2.7 Total failure

It was also tested what would happen if both the heat exchanger and the pump fail. This was done by starting the transients in the same manner as when tested individually. It is expected that the system behaves similar to a heat exchanger failure. Since the decrease in velocity, given that the heat exchanger fails, only slows the transfer of heat to the walls, which does not have a significant impact on the temperature of the salt. The result of this transient can be seen in Figure 4.15. The graphs show similar forms to those of the heat exchanger failure but the temperature decreases more slowly. Note that the timescale of this graph is a factor two larger than the one in Figure 4.12.
Figure 4.15: Top left a graph of $k_{eff}$ vs time, bottom left temperature and right the power production (blue) and removal (red) as the heat exchanger and pump stop working. As with the failure of the heat exchanger (Figure 4.13), the power production decreases to zero watt.
Chapter 5

Conclusions and recommendations

Conclusions to be drawn from this project and its results:

• A MSFR can be simulated by coupling the in-house developed codes HEAT and DALTON.

• The loss of reactivity due to the speed of the salt was shown to be almost constant if \( v < 1 \) \([\text{m/s}]\). At this speed a substantial amount of precursors flow back into the core area before decaying.

• When the flow speed increases above 1 \([\text{m/s}]\) the power production increases slightly.

• At \( v = 2 \) \([\text{m/s}]\) around 1.8GW of thermal energy is produced and extracted. The temperature of the reactor was around 1050[K], see Figure 4.3.

• The equilibrium temperature of the reactor slowly decreases as the velocity increases. The decrease in temperature increases the reactivity of the fuel, compensating the loss of precursors.

• The temperature feedback coefficient of the reactor was shown to be negative, around -6[pcm/K]. This expansion of the fuel was taken into account when calculating the group constants but was neglected as the flow was calculated.

Recomendations and ideas for further research on this subject:
Chapter 5. Conclusions and recommendations

- The sheer forces have been neglected in the simulations. Although the sheer forces are only large around the edges of the breeding zone, it might have an effect on the outcome of the simulation. Especially, the speed of the flow in the centre of the reactor. Aside from this, the accuracy of the code (±2% of the production, even more as more elements are used) raises questions.

- The discontinuities of the system during the emergence of turbulence should be investigated.

- The distribution of the zones could be refined. For instance, the plena each consist of a single zone, while the temperature distribution shows a substantial gradient in the area. It might be good to split these regions. For example into the zones used for the velocity-averaging (Figure 3.2).

- In the thermal hydraulics code the density of the salt did not depend on its temperature, resulting in spots with high temperature gradient and little flow. This would also impact the reactivity loss due to flow.

- Due to an unresolved error there is a large difference between the power produced and the power extracted. This may be due to the heat transfer between the salt in the core and the reflector. This should be investigated and resolved to reach a more accurate equilibrium temperature.

- No material evolution calculations were implemented.
Appendix A

Introduction

In the simulation of the molten salt reactor, pre-existing codes were combined. Apart from SCALE, the in-house developed FORTRAN codes Mix, HEAT and DALTON were used. These codes are discussed in Appendix B. Also, a Perl script has been created to initialise and run the simulation. This script is discussed in Appendix C. Running this script executes the simulation. Note that SCALE must first generate a library of group constants, this is done by running qsub quemakelibrary in ~/SCALE/. See Appendix B.1.

Some general remarks about the appendices:

- names of files and commands are given in typewriter font
- in the appendix ~/ is the directory containing sub-directories of HEAT, DALTON, Mix and SCALE. This is /coupling/, in this case.
- unless stated otherwise, files mentioned can be found in the directory of the code discussed.
- the only persistent exception is MAININPUT which can be found in ~/
- files following qsub contain instructions for the cluster followed by regular commands
Appendix B

Code systems used

B.1 SCALE

SCALE creates libraries of cross sections at different temperatures for further use by Mix. The input to SCALE consists of:

- the temperatures to be used in the library, set in \texttt{MAININPUT}
- the materials used, which are hard-coded in \texttt{makescale.f90}

The routine created from \texttt{makescale.f90} generates intermediate files: libraries and a Perl script (\texttt{runbatch5}) which controls the final calculations. The output consists of libraries of group constants in specific regions for specific temperatures. An overview of the system is given in Figure B.1. This routine is executed only once and can be executed on the cluster by running \texttt{qsub quemakelibrary} from the SCALE directory. The commands are then sent to the hpc11 cluster of the tudelft.

Figure B.1: Flow chart for SCALE, \( n \) runs from 1 to the number of temperatures given in \texttt{MAININPUT}
B.2  Mix

B.2.1  Introduction

Based on the temperature of the reactor, Mix interpolates the data from the libraries created by SCALE to estimate the cross sections per zone. Recall that various quantities are averaged over zones, these zones are shown in Figure B.2.

B.2.2  The in and output

The input to Mix consists of:

- the cross sections from the library and their temperatures, contained in the library files (library.1.n) and MAININPUT respectively.
- the temperatures per zone, defined in fort.223

The output consists of a cross section library for the current state of the reactor, library.xs.

B.2.3  Methodology

The library consists of group sections at various temperatures. Based on the current temperature of the zone, the appropriate values are calculated using linear interpolation.

B.3  DALTON

B.3.1  Introduction

The in-house developed code system DALTON uses the cross sections estimated by Mix together with the velocity profile to calculate the multiplication factor of the system. Using the power production of the previous step, the new production is estimated.

B.3.2  The input and output

The input to DALTON consists of:

- the geometry of the core and the elements, the power production from the previous step and the flowfield to include, if any. This inputfile (inputdalton) is made by makedalton.f90 (which is compiled by the coupling routine to DINPUT). The resulting executable is ran with three arguments:
1. 'T' or 'F' true for extra output, mainly useful for debugging.

2. '1', '2' or '4' for size of the flowfield. If this argument is 1 the full flowfield, identical to the one used by HEAT is used. This is time consuming and likely to result in an error. If it is 2, the size of the field is one fourth of that used in HEAT: for example, cells (1,1) to (2,2) are joined. If this is set to 4 the velocities between the zones in figure 3.2 are used.

3. 'T' or 'F' true for any flow at all

This routine also uses the value for $k_{\text{eff}}$ stored in $\text{keff}$ together with the power production stored in $\text{Ptot}$ to estimate the new power production.

- the velocity field (in [cm/s]) is given in either $\text{flowfield}$, $\text{flowfield_5}$ or $\text{flowfield-zone}$. Depending on which grid was used. The location of this file can be found in $\text{makedalton.f90}$.

- the cross sections of the material in the core, $\text{library.xs}$ in the directory of Mix

The output of Dalton is:
Chapter B. Code systems used

B.4. HEAT

Figure B.3: files involved in the execution of DALTON

- the power production, written to \texttt{fort.33}. This file consists of the region number\textsuperscript{1} the volume of the region in \([\text{cm}^3]\) and the power production in that region in \([\text{W/cm}^3]\).

- the total power production in \([\text{W}]\). This is written to \texttt{Ptot}.

- the \(k_{\text{eff}}\) calculated. This is written to \texttt{keff}.

Or in terms of files see figure B.3

\textbf{B.3.3 Methodology}

DINPUT estimates the total power production by scaling the old production by \(k_{\text{eff}}\) of that step:

\[
P_{\text{new}} = k_{\text{eff}} P_{\text{old}}.
\]

Using the input arguments, it also selects which flowfield and grid to use. This is all written to \texttt{inputdalton}.

DALTON then calculates the shape of the flux using the cross-sections and the velocity of the salt. The amplitude of the flux is then scaled to meet the total power production set in the inputfile. This production per element is then used to calculate the production per zone.

\textbf{B.4 HEAT}

\textbf{B.4.1 Introduction}

The in-house developed code system HEAT calculates the temperature and velocity field in the reactor.

\textsuperscript{1}see B.2
B.4. HEAT

The input to HEAT consists mainly of the power production due to fission and the geometry. Also, properties of the salt are hard-coded. During the calculations HEAT solves the balances for energy and momentum; calculating the temperature and velocity fields. See Section 2.3.

B.4.2 The input and output

The input to HEAT consists of:

- in MAININPUT:
  - the geometry of the core and its elements
  - the maximum number of iterations to be done
  - the heat transfer coefficient
  - the velocity of the pump
  - the size of the timestep
  - various initial temperatures
- the power production as defined in fort.33 in the directory of DALTON. This file consists of the region number\(^2\), the volume of the region in [cm\(^3\)] and the power production in that region in [W/cm\(^3\)].
- the value of some constants and properties of the salt are coded in physconst.f90.
- if applicable, the values of the calculated quantities at the end of the last run, which are stored in the binary file DATA.bin.

The output of HEAT consists of:

- the values of the calculated quantities are written to DATA.bin.
- the temperatures averaged per zone, the same zones as used for the power production. These are written to fort.223 in the directory of Mix.
- the velocity output is written to flowfield in the Dalton directory. Also, scaled down versions of the flow profile are written to flowfield5 and flowfield-zone. Note that Dalton is a reactor physics code and therefore uses centimeters while HEAT uses meters.
- the logfiles: (in ~/logs)

\(^2\)see Figure B.2
Chapter B. Code systems used

B.4. HEAT

Figure B.4: Files involved in the execution and coupling of HEAT

- relchanges
- statistics

Also, some other files are written for diagnostic purposes, see writeflow.f90 and calc_T.f90 for more details. A flowchart of the in and output is shown in Figure B.4

B.4.3 Methodology

The problem is discretised on a grid as defined in the input file, with the temperatures defined at the center of the elements and the velocities at the interfaces. The grid used is shown in figures B.5 and B.6. The actual calculation in HEAT is done in two loops: the inner and outer loop.

In the inner loop the quantities are calculated time independently, this is done a few times. Then in the outer loop this is repeated for a number of timesteps, the time advances and a new sequence of inner iterations start.

Each quantity is calculated by forming a linear problem using the appropriate balance. This linear problem is then solved using a maximum number of steps, this maximum is hard-coded in the respective calc_files. The result is the value calculated during the final step, either as the convergence limit is reached or as the maximum number of loops is reached.

The input file has a restart option, this is used to make HEAT load the results of the previous run as initial values. However, this value is overwritten in read_input.f90 to prevent mistakes.

The production of heat is done by fission and is set in fort.33 the removal of heat happens by leakage to the surroundings and by a heat exchanger. This heat exchanger is modeled to work only in the channel.
Figure B.5: The radial grid used in HEAT

Figure B.6: The vertical grid used in HEAT
Chapter B. Code systems used

B.4. HEAT
Appendix C

coupling of the codes

C.1 Introduction

The libraries containing cross sections must be made first by executing qsub quelmakelibrary from ~//SCALE. Once this has been done the simulations can be performed. The main executable in the coupled calculation scheme is the script (coupled). The command qsub job_couple executes job_couple which sends coupled to the que of the hpc11.tudelft.net cluster. This script is described here.

Its goal is to:

1. initialise the loop
2. run the loops
3. initiate transients
4. save the simulation data to a unique file

A view of the programs and the quantities passed during the simulation can be seen in Figure C.1, note that the execution of SCALE is not included in the script. The subroutines used are either in the file itself or located in ~/Perlsub.

C.2 Methodology

- First the script sets up the grid. A single gridsize parameter is given in the file of the script. This parameter is called $m$ which has to be an even integer, this parameter is used in the subroutine to define all elements.
  This subroutine also uses input parameters $n$ and the speed of the pump.
  Parameter $n$ defines the timestep size according to the speed of the pump.
The size of the timestep is such that a particle travelling at the speed of the pump through the channel, covers a distance of $n$ times the vertical length of an element in that region. The amount of the timesteps to be used by HEAT is also calculated; the script contains a variable to determine how much time should be between DALTON runs, and calculates the number of timesteps by HEAT accordingly. These are written in MAININPUT. This file is a copy of MAININPUToriginal, this original file is not used in the simulation but only used as a starting point.

- After Each timestep the script writes the output of the programs to $\sim$/results/coupling_data$x$, where $x$ is a time-dependent addition.

- The running of the simulation should be clear from section 3 or Figure C.1 or 3.1

- The script monitors the value of $k_{\text{eff}}$ and
  - After convergence is reached without any flow (see section 3) it turns on the flow
  - If instructed to do so it starts transient(s) at the second time convergence is reached (see section 3)
  - It exits the loop if convergence is reached again.
C.2. Methodology

Chapter C. coupling of the codes

Figure C.1: Files involved in the execution the simulation
Bibliography


