

A feasibility and safety evaluation of the ^{99}Mo Molybdenum Producing Mini Loop

by

Laurens T.B. Haffmans

March 11, 2021
Bachelor's thesis
Delft University of Technology

Student number: 4675630
Supervisor: Dr. ir. M. Rohde
Department: Radiation Science and Technology (RST)
Section: Reactor Physics and Nuclear Materials (RPNM)
Group: Transport Phenomena in Nuclear Applications



Abstract

^{99}Mo is a vital medical isotope that enables 30 million patients each year being diagnosed with severe diseases in an early stage. Since the five main producing nuclear reactors are aging, increased downtime due to maintenance has led to worldwide ^{99}Mo shortages in 2009-2010. In the search for a new production design, researchers from the Reactor Institute Delft (RID) have been investigating a new method to produce the isotope. Prior to testing in the Hoger Onderwijs Reactor (HOR), numerical simulations form the basis for its development.

The ^{99}Mo Producing Mini Loop (MPML) is the most recent research design of the RID. It consists of a heat exchanger and a closed loop that is placed next to the core of the HOR. The loop contains a solution of uranyl nitrate in water. The solution flows due to natural convection. Exposure of the loop to a neutron flux provokes a fission reaction, from which ^{99}Mo is formed and heat is generated. The uranyl nitrate concentration determines the amount of ^{99}Mo that is produced and is preferably as high as possible. In order to prevent high pressure buildup, the maximum temperature in the system is required to remain below 90°C . The setup is surrounded by water at 40°C .

Dresen [2019] investigated the feasibility of the MPML through a mathematical simulation in Matlab. Steady state temperatures were calculated for varying concentrations and cooling conditions. These were acquired by solving steady state internal energy balances and a momentum balance. The results predicted that a concentration of 236 g/L could be obtained by inserting a coolant mass flow of 0.7 kg/s at a coolant inlet temperature of 5°C .

The aim of this thesis was to investigate feasibility and safety aspects of the loop. Mistakes made by Dresen [2019] were corrected and the stationary simulation was extended to solving a transient model containing less simplifications. Important extensions to the computation included heat flow in vertical parts of the loop, heat exchange with the surrounding water, the inclination angle of the near-horizontal parts and gamma radiation exposure to the entire loop wall. Moreover, the design was adjusted to be suitable for placement in the HOR. Using the extended simulation, various emergency cases were investigated.

Correcting the stationary Matlab simulation and extending the model in Python produced promising feasibility results. By correcting the Matlab simulation and including heat exchange with the surrounding water in the model, steady state temperatures were derived to be significantly lower than predicted by Dresen [2019]. Moreover, it was discovered that inclusion of the two vertical nodes in the simulation is necessary to unveil a temperature peak that occurs during the initial flow development. This temperature peak was found to be lower when simulating gamma radiation exposure in the tube wall instead of in the fluid, including heat exchange with the surrounding water and including the inclination angle of the near-horizontal nodes in the simulation. Besides, it was shown that gamma radiation shielding and a variation of the length of the side of the MPML do not affect the feasibility. Lastly, it was derived that the inclination angle forces the flow into either clockwise or anti-clockwise direction.

At a maximum possible uranyl nitrate concentration 310 g/L and a coolant mass flow 0.01 kg/s at 10°C , the maximum steady state temperature and peak temperature were respectively predicted to be 47.1°C and 53.5°C . Applying the extended model to different emergency cases showed that temperatures remain safely below 90°C even in the extreme conditions of a defect cooling pump, stopped heat flow to the surrounding water or an increased reactor power. Should the first two emergency cases occur simultaneously, heat production must be terminated within 100 seconds.

Contents

Abstract	i
1 Introduction	1
1.1 ⁹⁹ Molybdenum	1
1.2 Previous design of the ⁹⁹ Mo loop	1
1.3 New research loop	2
1.4 Goals	3
1.5 Research approach	4
2 Modeling approach	5
2.1 Nodes	5
2.2 Definitions for temperature and heat transfer coefficients	5
2.3 Parameters, variables, equations and conditions	7
2.4 Numerical research	9
3 Design of the MPML	10
3.1 Placement in the HOR	10
3.2 Geometry of the setup	10
3.3 Material properties	12
3.4 Shielding	12
4 Theoretical Background	14
4.1 Governing equations	14
4.1.1 Mass balance	14
4.1.2 Internal energy balance	14
4.1.3 Momentum balance	15
4.1.4 Modelling natural convection: the Boussinesq approximation	15
4.2 Heat generation	16
4.2.1 Nuclear fission	16
4.2.2 Gamma radiation	16
4.3 Dimensionless numbers	16
4.4 Friction	18
4.5 Flow of heat	19
5 Mathematical model	22
5.1 Boussinesq approximation	22
5.2 Mass balance	22
5.3 Internal energy balance	23
5.3.1 Internal energy balance node A	23
5.3.2 Internal energy balance node D	23
5.3.3 Internal energy balance node B	23
5.3.4 Internal energy balance node E	24
5.3.5 Internal energy balance node C	24
5.3.6 Internal energy balances over the tube walls	25
5.3.7 Negative velocities	26
5.4 Momentum balance	27
5.4.1 Negative velocities	29

6	Numerical method	30
6.1	Solving ordinary differential equations	30
6.1.1	Ordinary differential equations	30
6.1.2	Runge-Kutta method	30
6.1.3	Adaptive Runge-Kutta method	31
6.1.4	Multi-variable adaptive Runge-Kutta method	32
6.2	Exception handling	33
6.3	Flowchart and variable names	33
7	Evaluating and benchmarking Dresen [2019]	36
8	Results	39
8.1	Optimizing the simulation	41
8.1.1	Question 1a	41
8.1.2	Question 1b	43
8.1.3	Question 1c	45
8.1.4	Question 1d	48
8.1.5	Question 1e	50
8.1.6	Question 1f	53
8.1.7	Question 1g	54
8.1.8	Optimized model	55
8.2	Safety	56
8.2.1	Question 2a	56
8.2.2	Question 2b	58
8.2.3	Question 2c	59
8.2.4	Question 2d	61
9	Conclusions	62
10	Shortcomings and recommendations	63
	References	66
	Appendices	1
A	Conduction around a cylinder	1
B	Consequences for the momentum balance	1
C	Python code	3

1 Introduction

1.1 ^{99}Mo Molybdenum

One of the fundamental revolutions of the past centuries in medical treatment has been medical imaging. Tens of millions of patients yearly benefit from non-invasive diagnostic imaging techniques in order to safely identify critical conditions in an early stage. Common applications include heart diseases and cancer. The resulting diagnose supports important decision making in the choice of treatment. Today, 80% of worldwide diagnostic nuclear medical imaging is based on the medical isotope Technetium-99m (Ahmad [2011]). The latter is a decay product of ^{99}Mo Molybdenum. With short half-lives of respectively 6 and 66 hours, these isotopes decay quickly and require daily production.

Despite its vital importance, the worldwide supply of ^{99}Mo Molybdenum isotopes largely relies on five nuclear reactors in Belgium, the Netherlands, France, South Africa and Canada. In these reactors, ^{99}Mo Molybdenum is obtained as a fission product by irradiating uranium-235 targets with neutrons. This first step of the extensive production process is time consuming and capital intensive, since the separate targets require individual processing. In this time, valuable isotopes have already started decaying. Besides the disadvantage of the batch process, the production facilities are nearing 50 years of age and experience increased downtime due to maintenance. In 2009-2010, shortages up to 70% of worldwide demand disrupted the supply chain. The Nuclear Energy Agency has since been determined to safeguard future production of ^{99}Mo Molybdenum by developing new, more efficient production methods (nuc [2021]).

1.2 Previous design of the ^{99}Mo loop

In the search for a new ^{99}Mo Molybdenum production design, the Reactor Institute Delft (RID) has been investigating the feasibility of a loop-type setup containing an aqueous solution of uranium. Before testing the design in the 2MW research reactor - the Hoger Onderwijs Reactor (HOR) - numerical analyses are aimed to determine the optimal design. Ideally, the design is a continuously circulating system that has a small volume (0.5L) and is not driven by a pump. The pump omitting requirement is to circumvent the high risk of heat accumulation in the emergency case of a defect pump. A small volume is required to mitigate effects in the case of an emergency and the continuous aspect is more efficient than the current batch process.

Numerical research by Elgin [2014], Huisman [2016] and Pothoven [2016] showed feasible results for the pump-driven, U-shaped tube as depicted in figure 1-1: an aqueous solution of uranyl nitrate flows through the tube and is exposed to a neutron flux from the reactor core next to the tube. The researchers deduced that a high uranyl nitrate concentration is preferable in order to maximize ^{99}Mo Molybdenum production. High concentrations also establish more heat production by fission, however, which requires an effective cooling system. For a concentration of 27.6 g/L temperatures remained safely under the boiling temperature of the solution. Ideally, the concentration should become 310 g/L, which is the maximum concentration that allows for effective ^{99}Mo Molybdenum extraction.

Pendse [2018] upgraded the design by closing the loop and removing the pump, as depicted in figure 1-2. The loop was driven by natural convection. Moreover, an upgraded cooling system allowed for uranyl nitrate concentrations up to 310 g/L while keeping safe temperatures. There was, however, one disadvantage: the volume was 5 liters.

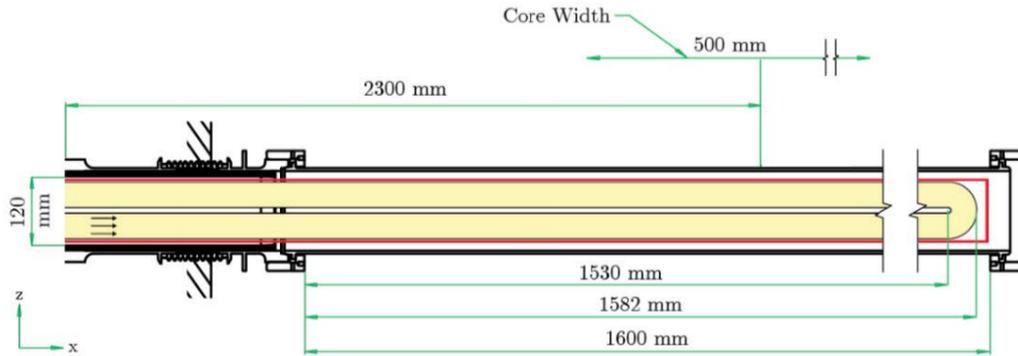


Figure 1-1: U-shaped research loop designed by Elgin [2014], Huisman [2016] and Pothoven [2016].

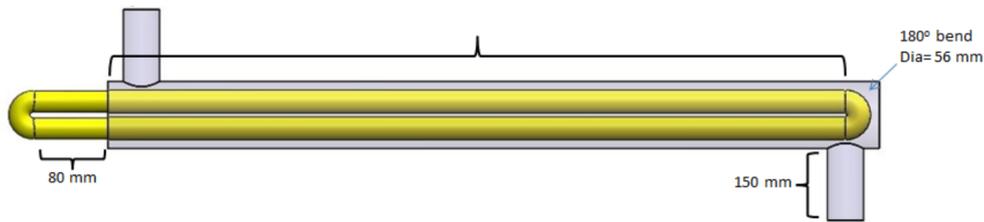


Figure 1-2: Previous design of the ^{99}Mo loop as investigated by Pendse [2018].

1.3 New research loop

The ^{99}Mo Molybdenum Producing Mini Loop (MPML) is a small loop with a maximum volume of 0.5 liters that is driven by natural convection. It consists of four connected, circular tubes containing a uranyl nitrate solution. The two near-horizontal nodes are placed at a 2° inclination angle. A sketch is given in figure 1-3. The lower tube is exposed to a neutron flux, where fission produces heat. Since the upper tube is cooled by a concentric heat exchanger, temperature and density differences in vertical direction cause the solution to flow by natural convection.

Dresen [2019] numerically determined steady state temperatures at varying concentrations and cooling conditions by simulating a simplified model. Stationary internal energy balances and a momentum balance were solved in Matlab. Results predicted that a concentration of 236 g/L could be obtained by inserting a coolant mass flow of 0.7 kg/s at an inlet temperature of 5°C . Since this coolant mass flow is too high to be obtained in practise, the simulation is evaluated and extended in this research to investigate the feasibility. Moreover, safety aspects in emergency cases are inquired.

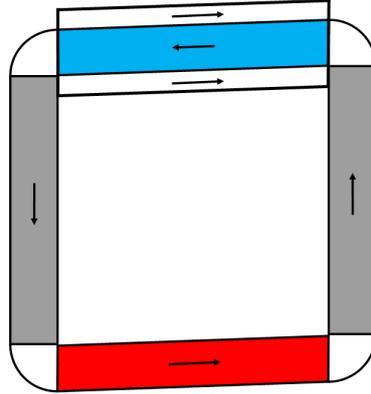


Figure 1-3: New research loop design: the $^{99}\text{Molybdenum}$ Producing Mini Loop. The lower tube is exposed to neutron flux and the upper tube is cooled by a concentric heat exchanger. Natural convection thrives the flow.

1.4 Goals

The aim of this research is to numerically investigate the feasibility and safety of the $^{99}\text{Molybdenum}$ Producing Mini Loop. The *feasibility* is defined as the extent to which a high uranyl nitrate concentration allows for natural convection and temperatures below $T < 90^\circ\text{C}$. *Safety* simulations are defined as those involving emergency cases. The research questions of this thesis are the following:

1. Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspects affect the feasibility predictions for the $^{99}\text{Molybdenum}$ Producing Mini Loop?
 - (a) Inclusion of the two vertical nodes in the simulation.
 - (b) Gamma radiation exposure in the tube wall instead of in the fluid.
 - (c) Heat exchange with the surrounding water.
 - (d) The inclination angle of the near-horizontal nodes.
 - (e) Positive and negative flow directions.
 - (f) Gamma radiation exposure in the non-fission nodes.
 - (g) Adjusted size for placement inside the DLDR.

The above adjustments to the model of Dresen [2019] are from here on referred to as *extensions* ranging from a to g.

2. Applying the extended simulation, how do the following emergency cases affect the safety of the MPML?
 - (a) A defect cooling pump causes the cooling to solely rely on heat flow to the surrounding water.
 - (b) An empty DLDR causes the cooling to solely rely on the heat exchanger.
 - (c) A defect cooling pump and an empty DLDR cause accumulation of heat in the loop.
 - (d) An increased neutron flux causes heat production by fission to rise.

1.5 Research approach

The research consists of two parts: extending the simulation of Dresen [2019] and using the extended simulation to inquire safety aspects.

In order to answer research question 1a-g, a simplified model is developed in which all extensions 1a-g are omitted. The results of this transient simulation are compared to the stationary simulation of Dresen [2019] as a benchmark. Subsequently the model is extended step by step in order to explore the effect of every single extension. For this part, isothermal initial conditions are applied. The resulting model including all the extensions 1a-g is referred to as the *extended* or *optimized* model.

To investigate emergency cases, steady state conditions obtained from the optimized model are applied as initials conditions to the optimized model.

2 Modeling approach

Below the definitions for the model are outlined, followed by an outline of the parameters, variables, equations, initial conditions and boundary conditions. Moreover, a short summary is given on the approach to answer the different research questions numerically.

2.1 Nodes

The MPML as analyzed by Dresen [2019] was divided into three grid points or nodes A , B and C . In this research, two additional nodes D and E are incorporated, as depicted in figure 2-4. Node A is exposed to a neutron flux from the reactor core and node B functions as the inner tube of a heat exchanger, surrounded by node C : nodes B and C together form a heat exchanger.

The four nodes inside the loop in the direction of the flow are respectively nodes A , D , B and E . These are from here on referred to as $j = A, D, B$ or E . The property V_j for instance refers to the volume of a loop node, which is equal for all the loop nodes j . In contrast, the index i is used when referring to all the nodes including node C : $i = A, D, B, E$ or C .

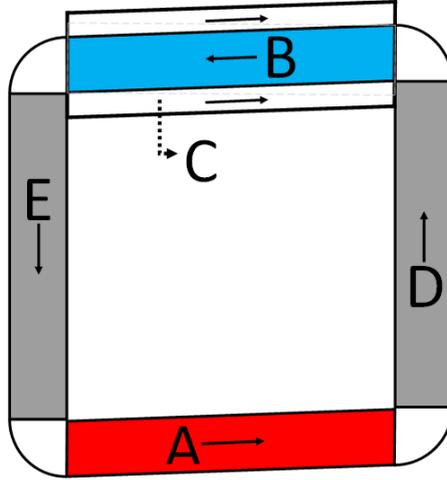


Figure 2-4: Five nodes A,B,C,D and E.

2.2 Definitions for temperature and heat transfer coefficients

Within every node the temperature is considered to be spatially constant. The outflow temperature equals the node temperature, as shown in 2-5. Time dependent densities are calculated using these temperatures inserted in equation (12). In the outer part of the heat exchanger, the inlet temperature is constant and the outlet temperature is transient. The temperature profile in node C is assumed to be linear.

Heat transfer from a fluid in a tube to the outside of the tube meets three heat resistances, defined by their heat transfer coefficients. In this research, the heat transfer coefficient from the flow within a tube to the tube is denoted by h_1 ; the heat transfer coefficient of the tube itself is denoted by h_2 ; and the heat transfer coefficient between the tube and the surrounding is denoted by h_3 .

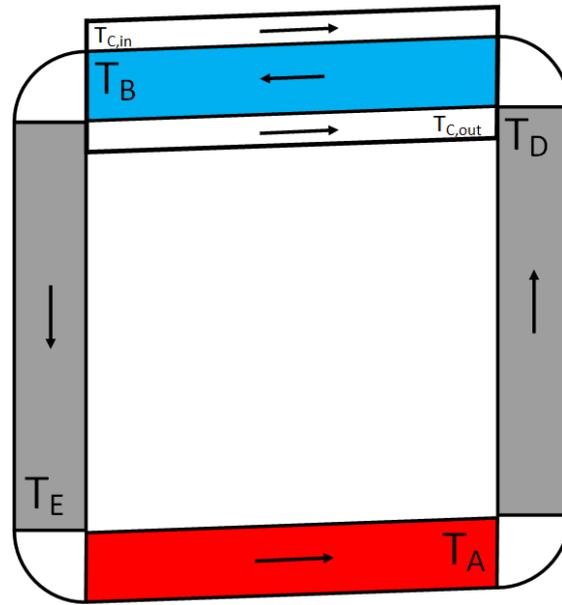


Figure 2-5: Temperature definitions in five nodes.

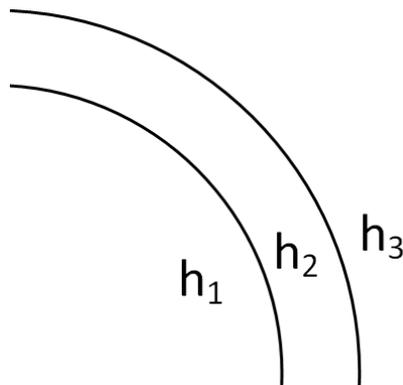


Figure 2-6: Representation of heat transfer coefficients for heat exchange through a tube wall. The heat transfer coefficients of the material inside the tube, the tube itself and the material outside the tube are denoted by h_1 , h_2 and h_3 respectively.

2.3 Parameters, variables, equations and conditions

The simulation is defined by six parameters, ten variables, ten differential equations, two boundary conditions and ten initial conditions.

Parameters and variables

Directly controllable quantities are named *parameters*. Ten time-dependent quantities or *dependent variables* follow from the simulation. The parameters and variables are outlined in figure 2-7 and table 1.

Type	Name	Symbol	Description
Parameters: independent	Coolant inlet temperature	$T_{in,C}$ ($^{\circ}\text{C}$)	A low coolant inlet temperature increases the cooling power of the heat exchanger, allowing for higher uranyl nitrate concentrations.
	Coolant mass flow	$\phi_{m,C}$ (kg/s)	A high coolant mass flow also increases the cooling power.
	Uranyl nitrate concentration	c (g/L)	The concentration c determines the heat production by fission in node A as well as the molybdenum production. It is desired to be as high as possible.
	Node length	l (m)	The length of the side of the MPML is equal to the length of a node, which is equal for all nodes. This length must be adjusted in order to fit inside the DLDR.
	Inclination angle	α ($^{\circ}$)	The inclination angle of the near-horizontal nodes affects the flow direction and confines the length l to fit inside the DLDR.
	Shielding transmission factor	stf	Gamma radiation may be shielded completely or partially from nodes D,C and E . The gamma radiation heat production in these tube walls is multiplied by a shielding transmission factor between 0 and 1.
Variables: dependent	Node temperatures	$T_A(t), T_D(t), T_B(t), T_E(t), T_{out,C}(t)$ ($^{\circ}\text{C}$)	The temperature of the uranyl nitrate solution in the five nodes A, D, B, E and C .
	Wall temperatures	$T_{w,A}(t), T_{w,D}(t), T_{w,C}(t), T_{w,E}(t)$ ($^{\circ}\text{C}$)	The temperature of the walls surrounding the nodes adjacent to the DLDR water.
	Average velocity	$\langle v(t) \rangle$	The uranyl nitrate solution velocity averaged over the loop: it is constant in space but varying in time.

Table 1: Parameters and variables

Differential equations

In total, ten first order differential equations are numerically solved: one for every variable. For every temperature there is a transient internal energy balance $\frac{dT_{...}}{dt} = \dots$ and the transient momentum

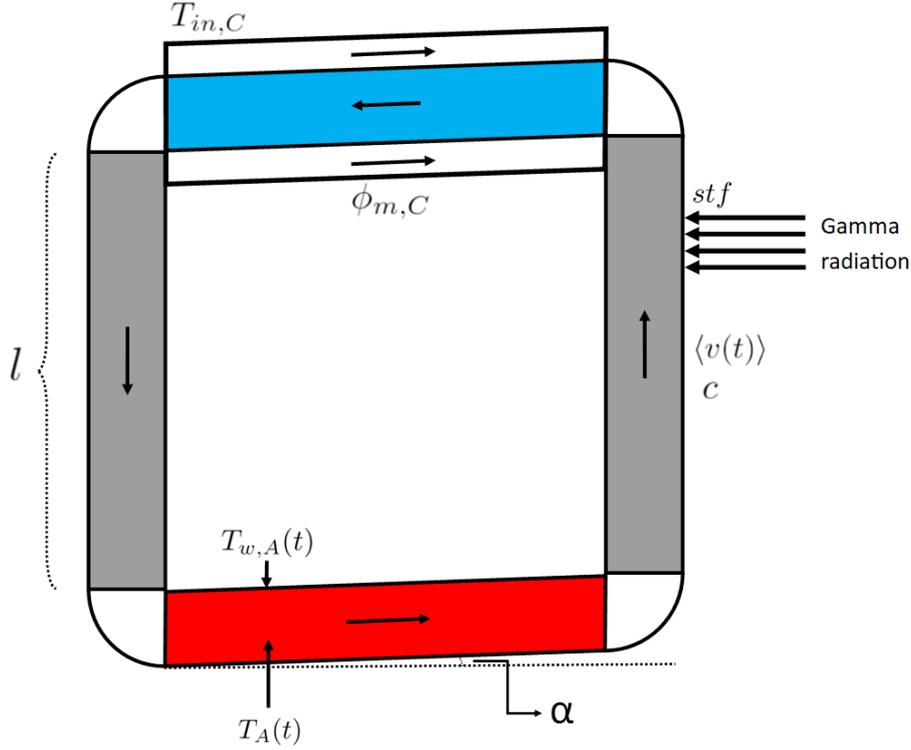


Figure 2-7: The parameters and variables as outlined in table 1

balance for the velocity $\frac{d\langle v(t) \rangle}{dt} = \dots$

Boundary conditions

Two boundary conditions are needed for the simulation. The first is the water temperature of the surroundings $T_{\infty} = 40^{\circ}C$. The second is a restriction for the heat exchanger: if the loop velocity is directed anti-clockwise ($v > 0$), then the heat exchanger is counter current. Given that the inlet temperature of the coolant is lower than the inlet temperature of the heated node, then the outlet coolant temperature must be lower than the inlet node temperature. If the loop velocity is directed clockwise ($v < 0$), the heat exchanger is co-current. The constraint becomes that the outlet coolant temperature must be lower than the outlet node temperature:

$$T_{out,C} < \begin{cases} T_D & v > 0 \\ T_B & v < 0 \end{cases} \quad (1)$$

The reason for this constraint originates in the heat exchanger (equation (34)): if the temperature difference decreases, heat flow also decreases until the temperature difference and the heat flow disappear. It is further noted that the above constraint theoretically only applies to the case that gamma radiation and heat exchange with the surrounding water are absent from node C . In practise, these two terms only slightly affect the node C temperature, however, and the constraint still holds.

Initial conditions

Ten initial conditions are needed for the variables. In this research two sets of initial conditions are relevant. The first defines the isothermal situation in which the reactor was inactive until $t=0$: temperatures are equal to the pool temperature and the velocity is zero. The second set of initial conditions describes the steady state conditions that were obtained from the simulation starting from the first set of initial conditions.

2.4 Numerical research

All research questions 1a-g consist of adding and removing the relevant terms from the internal energy balances over the nodes and the tube walls. The following adjustments are made in order to evaluate emergency cases:

- The steady state results of the optimized model are used as initial conditions.
- To simulate the effects of a defect cooling pump in questions 2a and 2c, the cooling mass flow is set to $\phi_{m,C} = 10^{-20}$ kg/s. Node C does not function as a heat exchanger anymore. Therefore, the boundary condition (1) does not apply and the heat exchanger equation (34) is replaced with ordinary flow of heat (equation (28)).
- To simulate an empty DLDR in question 2b, heat exchange between the wall and the surrounding is omitted.
- An increased neutron and gamma flux in question 2d are simply found by multiplying the neutron flux ϕ_n and the gamma radiation heating u by a factor larger than 1.

3 Design of the MPML

3.1 Placement in the HOR

For research purposes, the MPML is placed inside the horizontal DLDR tube in the Hoger Onderwijs Reactor (HOR) at the Reactor Institute Delft (RID), as depicted in figures 3-8 and 3-9. The DLDR is surrounded by a water pool at constant temperature $T = 40^\circ\text{C}$ and the DLDR itself is also filled with water. The only heat resistance between the DLDR water and the pool water is an aluminium tube, which conducts heat significantly better than water¹. Therefore, the DLDR water is assumed to be at constant temperature $T = 40^\circ\text{C}$ as well.

The DLDR water flows at low speed $\phi_V = 10^{-5} \text{ m}^3/\text{s}$ through the DLDR and through an external tube system in which contamination is removed. In this research, however, the DLDR water is simulated as if it is standing still: $\phi_V = 0 \text{ m}^3/\text{s}$.

Experiments in the HOR generally require at least two safe barriers between the radioactive material and the pool. From personal communications with A. Winkelman (2021) it has been determined that the MPML loop and the DLDR together comply with this constraint, if the first is robustly built. In a later stadium, an extra safety tube could be added between the loop and the DLDR.

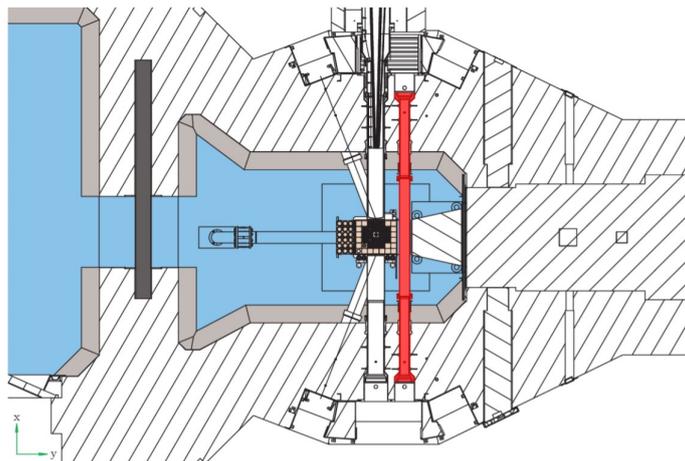


Figure 3-8: Representation of the DLDR setup next to the reactor core in the HOR. The red tube is the DLDR, next to the square black reactor core. It is surrounded by a water pool at $T = 40^\circ\text{C}$. (Huisman [2016])

3.2 Geometry of the setup

As shown in figure 3-9, the DLDR has a length of 1600 mm and an inner diameter of 140 mm (Pendse [2018]). The four tubes that together form the loop are of the same length l . The length was 200 mm in the research of Dresen [2019] but is adjusted for placement inside the DLDR. The loop has an inner diameter of $r = 6 \text{ mm}$ and a wall thickness of $d = 2 \text{ mm}$. The inner radius of node C is $R = 8 \text{ mm}$. The outer radii and diameters are shown in table 2 and figure 3-10. The

¹The heat conductivity of aluminium is significantly larger than that of water: $\lambda_{aluminium} = 237\text{W}/(\text{mK}) \gg \lambda_{water} = 0.596\text{W}/(\text{mK})$ (Janssen and Warmoeskerken [1987]).

inclination angle of nodes A and B with the horizontal axis is denoted by the quantity α in degrees. This gives the following constraint for the length of the tube to fit inside the DLDR:

$$l = \frac{D_{dlr} - R - 3r - 4d}{1 + \sin \alpha}. \quad (2)$$

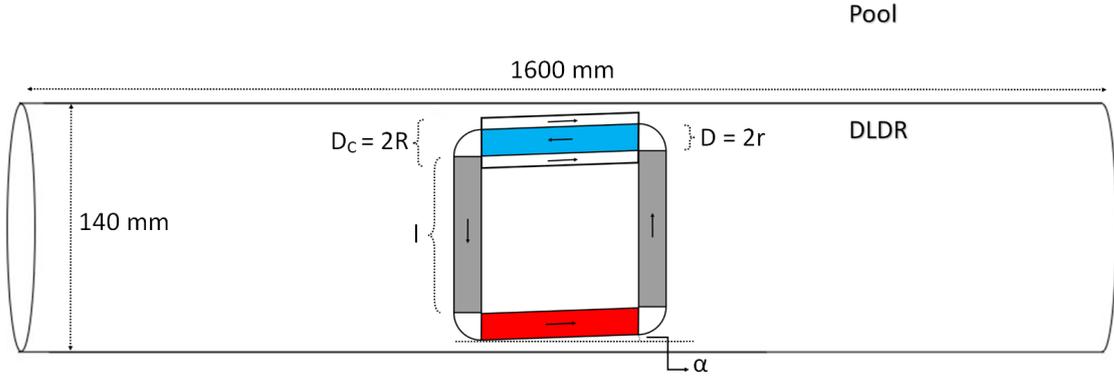


Figure 3-9: Representation of the MPML setup in the DLDR.

radius	diameter
$r = 3 \text{ mm}$	D
$r_{out} = 5 \text{ mm}$	D_{out}
$R = 8 \text{ mm}$	D_C
$R_{out} = 10 \text{ mm}$	$D_{C,out}$

Table 2: Radii and diameters corresponding to the tube around node B and the heat exchanger tube node C , as represented in figure 3-10 and 3-9. The radii r , r_{out} and diameters D , D_{out} are equal for all loop nodes j .

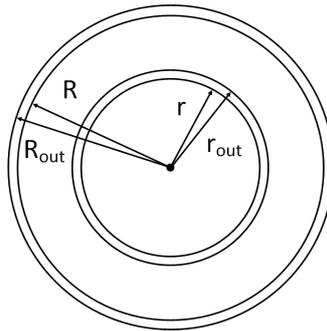


Figure 3-10: Cross-sectional area of the heat exchanger, that consists of nodes B and C .

3.3 Material properties

Zircaloy

The loop itself is made of zircaloy (zirconium alloy), for which the choice is outlined by Dresen [2019]. For the effective roughness, the general value for aluminium tubes is used since the zircaloy value was not available. All properties are shown in table 3.

Coolant

Water functions as the coolant in the heat exchanger: the fluid in node C . Besides the properties in table 3, the temperature dependent dynamic viscosity $\mu(T)$ is given for the range $0 - 100^\circ C$ (Dresen [2019]):

$$\log \frac{\mu(T)}{\mu_{20}} = \frac{A(20 - T) - B(T - 20)^2}{T + C}, \quad (3)$$

in which $A = 1.1709$, $B = 0.001827$, $C = 89.93$ and the reference value at $T = 20^\circ C$ is $\mu_{20} = 1.0020$ mPas. The Prandtl number is found from $Pr_w(T) = \frac{\mu_w(T) \cdot c_{p,w}}{\lambda_w}$.

Uranyl nitrate solution

The uranyl nitrate solution contains low enrichment uranium, with an enrichment $\epsilon_5 = 19.75\%$. The molar mass of uranyl nitrate is 394.04 g/mol and the thermal diffusivity of the uranyl nitrate solution is found from $a = \frac{\lambda}{\rho_{ref} \cdot c_p}$. The dynamic viscosity of the solution depends on the temperature and concentration (Grant et al. [1948]):

$$\mu(c_{mol}, T) = \mu(T) \cdot (1 + A \cdot \sqrt{c_{mol}} + B \cdot c_{mol}), \quad (4)$$

where c_{mol} is the concentration [mol/L], $\mu(T)$ is the dynamic viscosity of water as found from equation (3) and $A = 0.1687$, $B = 0.7904$. The Prandtl number becomes $Pr(T) = \frac{\mu(c_{mol}, T) \cdot c_p}{\lambda}$. Other properties are outlined in table 3.

DLDR water

The surrounding water in the DLDR is at $40^\circ C$. Properties of the surroundings are found in the table below.

3.4 Shielding

As shown in figure 3-11, the reactor core is placed close to the DLDR along with two air boxes. This leaves little space for shielding of gamma radiation that has an energy up to 20 MeV. Therefore the entire loop is assumed to be exposed to gamma radiation, contrary to the complete gamma-shielding assumption of Dresen [2019].

The neutron flux, however, can be blocked with a cadmium layer. The orange line in figure 3-11 represents a shielding to block neutrons from all parts of the loop except the lower horizontal part. Placement of this cadmium layer directly onto the DLDR would be disadvantageous since that would shield neutron flux towards the fission part as well, decreasing the production rate in equation (15). Placement directly onto the MPML would be disadvantageous for heat exchange between the loop and the surrounding water. It is therefore suggested to be placed at a short distance from the loop.

Material	Property		Value	Source
Zircaloy	Neutron cross section	$\sigma_{f,5}$	583 barn	Huisman [2016]
	Specific heat capacity	$c_{p,t}$	285 J/(kgK)	
	Thermal conductivity	λ_t	21,5 W/(mK)	Zir [2014]
	Density	ρ_t	$6.55 \cdot 10^3 \text{ kg/m}^3$	
	Effective roughness	ϵ	$1.5 \cdot 10^{-6} \text{ m}$	Dresen [2019]
Coolant: water properties at 10°C, 10 ⁵ Pa	Density	ρ_w	999.73 kg/m ³	
	Specific heat capacity	$c_{p,w}$	4203 J/(kgK)	Janssen and Warmoeskerken [1987]
	Thermal conductivity	λ_w	0.574 W/(mK)	
	Thermal diffusivity	a_w	$0.138 \cdot 10^{-6} \text{ m}^2/\text{s}$	
Uranyl nitrate solution	Density	ρ_{ref}	1330.6 kg/m ³	Huisman [2016]
	Specific heat capacity	c_p	2905.5 J/(kgK)	
	Thermal conductivity(λ^*	0.665 W/(mK)	Value of water Dresen [2019]
	Thermal expansion coefficient	β	$5.23 \cdot 10^{-4} \text{ K}^{-1}$	
Surroundings: water properties at 40°C, 10 ⁵ Pa	Density	ρ_s	992.95 kg/m ³	
	Specific heat capacity	$c_{p,s}$	4183.3 J/(kgK)	Janssen and Warmoeskerken [1987]
	Thermal conductivity	λ_s	0.6274 W/(mK)	
	Thermal expansion coefficient	β_s	$3.85 \cdot 10^{-4} \text{ K}^{-1}$	

Table 3: Material properties. *The property of water at 60°C, 10⁵Pa is used.

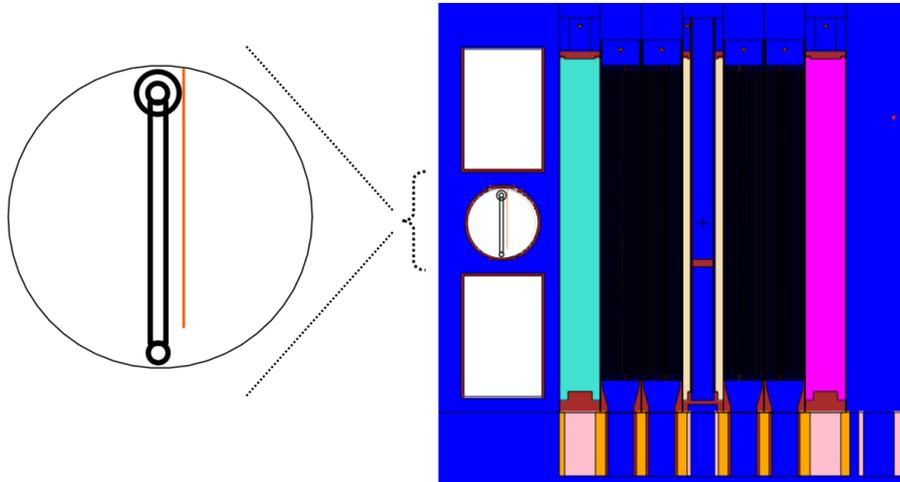


Figure 3-11: Cross-sectional representation of the reactor core, the DLDR and the MPML. The orange line inside the DLDR represents a layer of cadmium shielding against neutrons.

4 Theoretical Background

4.1 Governing equations

Transport phenomena processes are described by mass, heat and energy transfer. This transfer is defined by a balance over a control volume. The total change of a quantity inside the volume is given by the sum of the flow into the volume, the flow out of the volume and the production inside the volume:

$$\frac{d}{dt} = \text{in} - \text{out} + \text{'production'}. \quad (5)$$

The mass balance, internal energy balance and momentum balance define the transport phenomena within a control volume. These are outlined below in their general form for fluid flow through a tube and are evaluated for every node in section 5.

4.1.1 Mass balance

The general mass balance for fluid flow through a tube is given as follows:

$$\frac{dM}{dt} = \phi_{m,\text{in}} - \phi_{m,\text{out}}, \quad (6)$$

in which no production term is present since since mass is a conserved property.

As outlined in section 4.1.4, the transient term $\frac{dM}{dt}$ vanishes in this research because the Boussinesq approximation is applied. The mass flow $\phi_{m,\text{in}} = \phi_{m,\text{out}} = \phi_m$ given in $[kg/s]$ is therefore constant. Mass flow and volume flow $\phi_V [m^3/s]$ of a fluid in a tube with cross-section $A [m^2]$, average speed $\langle v \rangle [m/s]$ and constant density $\rho [kg/m^3]$ are then given by:

$$\phi_m = \rho A \langle v \rangle, \quad \phi_V = \frac{\phi_m}{\rho}. \quad (7)$$

4.1.2 Internal energy balance

The general internal energy balance for a reacting system with inflow and outflow of mass, that produces heat over a control volume V , is given by (Van den Akker and Mudde [2014]):

$$\frac{dU}{dt} = \phi_{m,\text{in}} \left(u + \frac{p}{\rho} \right)_{\text{in}} - \phi_{m,\text{out}} \left(u + \frac{p}{\rho} \right)_{\text{uit}} + \phi_q + P_u,$$

where $U [J]$ denotes the total internal energy in the control volume, ρ is the density, $p [Pa]$ is the pressure, $u [J/kg]$ is the specific internal energy, ϕ_m is the flow of mass, $\phi_q [J/s]$ is the net flow of heat and P_u is the total production of internal energy.

In this research constant mass flow $\phi_{m,\text{in}} = \phi_{m,\text{out}} = \phi_m$ is assumed within every node. That is because the density is assumed to be constant within a node, as outlined in section 5.2. The balance becomes:

$$\frac{dU}{dt} = \phi_m \left[u_{\text{in}} - u_{\text{out}} - \int_{\text{in}}^{\text{out}} p \cdot d \left(\frac{1}{\rho} \right) \right] + \phi_q + P_u.$$

With constant pressure and $du = c_V dT \approx c_P dT$ holding for fluids (Van den Akker and Mudde [2014]), this simplifies to:

$$\frac{dU}{dt} = \phi_m c_p (T_{\text{in}} - T_{\text{out}}) + \phi_m p \left(\frac{1}{\rho_{\text{in}}} - \frac{1}{\rho_{\text{out}}} \right) + \phi_q + P_u.$$

The left hand side can be rewritten firstly as a function of temperature and pressure and secondly using the material derivative. The derivation as given by equation 1.59 of Rohde [2014] gives the following result:

$$\rho V c_P \frac{d}{dt}(T) = \phi_m c_p (T_{\text{in}} - T_{\text{out}}) + \phi_m p \left(\frac{1}{\rho_{\text{in}}} - \frac{1}{\rho_{\text{out}}} \right) + \phi_q + P_u. \quad (8)$$

4.1.3 Momentum balance

The one-dimensional general momentum balance for a medium that flows in the direction \hat{l} through a volume V is given by (Van den Akker and Mudde [2014]):

$$\frac{d}{dt}(Mv_{\hat{l}}) = \frac{d}{dt}(\rho V v_{\hat{l}}) = \phi_{m,\text{in}} \cdot v_{\hat{l},\text{in}} - \phi_{m,\text{out}} \cdot v_{\hat{l},\text{out}} + \sum F_{\hat{l}}, \quad (9)$$

in which $v_{\hat{l}}$ is the velocity and $\sum F_{\hat{l}}$ is the sum over the forces exerted on the flowing medium. An important characteristic for flow in non-circular tubes is the hydraulic diameter, used for calculations analogous to the diameter of a circular tube:

$$D_h = \frac{4A}{S}, \quad (10)$$

in which A is the cross-sectional area of the flow and S the wetted perimeter. The latter is defined as the perimeter of the cross-sectional area tangent to the fluid.

4.1.4 Modelling natural convection: the Boussinesq approximation

In the MPML a temperature gradient causes density differences inside the loop, which makes natural convection the mechanism behind the flow. In computational fluid dynamics the Boussinesq approximation is used to simplify the mass, energy and momentum balance for fluid flows driven by buoyancy (Deen [1998]).²

The approximation is based on a description of the density as a transient density field, consisting of on the one hand a reference density that only varies in space, and on the other hand a term describing fluctuations in time for the different points in space (equation 5.5 in Rohde [2014]):

$$\rho(\vec{r}, t) = \rho_0(\vec{r}) + \rho'(\vec{r}, t). \quad (11)$$

Substituted into the momentum balance, the density change ρ' caused by a temperature gradient only significantly affects the gravitational term and has a negligible effect on all other terms in the mass, energy and momentum balance (Rohde [2014]). The approximation consists of two assumptions (Deen [1998]):

1. *The effect of pressure variation on the density are assumed to be negligible and the density change is a linear function of temperature change. Therefore the density is dependent on the temperature according to the following equation of states:*

$$\rho(T) = \rho_0 - \rho_0 \beta (T - T_0), \quad \beta \Delta T \ll 1. \quad (12)$$

β is the thermal expansion coefficient of the fluid, ΔT is the maximum temperature difference in the system, and 0 denotes an arbitrary reference value for the temperature and density. As denoted by Dresen [2019], the condition is satisfied in the MPML.

2. *The variable density $\rho(t)$ can be replaced everywhere by the constant value ρ_0 , except in the gravitational term of the momentum balance.*

²The Boussinesq approximation does not hold for gasses (Deen [1998])

4.2 Heat generation

4.2.1 Nuclear fission

A section of the loop is exposed to a neutron flux. These neutrons interact with the dissolved uranium in the form of nuclear fission. As outlined by Dresen [2019] the number of fission events per second per cubic meter, also known as the *reaction rate*, is given by:

$$R''' = \frac{\sigma_{f,5} \cdot \epsilon_5 \cdot c \cdot N_A \cdot \phi_n}{M}, \quad (13)$$

in which $\sigma_{f,5}$ is the fission cross-section in barn ($= 10^{-28} m^2$), ϵ_5 the enrichment, c is the uranium concentration [g/m^3], N_A is the number of Avogadro, ϕ_n [$1/(m^2s)$] is the neutron flux and M [g/mol] the molar mass of uranyl nitrate. In the HOR the average neutron flux directly in front of the reactor is $\phi_n = 3.5 \cdot 10^{16} m^{-2}s^{-1}$.

The heat generated per second by this process is:

$$Q = V E_f R''', \quad (14)$$

where Q [W] is the energy production, V is the volume and E_f [J] the thermal energy release per fission event: 192 MeV.

To calculate the 99 Molybdenum *production rate*, the reaction rate is multiplied by the characteristic *fission yield* of 99 Molybdenum, which is typically 6, 13%.

$$P'''_{Mo} = \gamma_{Mo} \cdot R'''. \quad (15)$$

The production rate P'''_{Mo} is the number of 99 Molybdenum elements produced per second per cubic meter.

4.2.2 Gamma radiation

As a byproduct of the nuclear fission, a gamma radiation flux heats the loop material. The energy production due to gamma radiation in a tube of radius r is given by (Dresen [2019]):

$$P_\gamma(r) = u \cdot \rho_{tube} \cdot V_t(r), \quad (16)$$

where $P_\gamma(r)$ is defined in [W], $V_t(r)$ and ρ_{tube} are the volume and density of the tube itself and u W/kg is the heating power per unit mass of construction material, close to the reactor core. The tube volume depends on the inner radius r , the tube thickness d and the length l :

$$V_t(r) = \pi l ((r + d)^2 - r^2) = \pi l (d^2 + 2dr). \quad (17)$$

As depicted in figure 4-12 the heat generation directly in front of the core at $x = 0.8m$ is $u = 300$ W/kg (Pendse [2018]).

4.3 Dimensionless numbers

Dimensionless numbers in general give the ratio between two physical phenomena of the same physical unit. This ratio can be a characteristic of either a process or a material. The following ratios are relevant for this research:

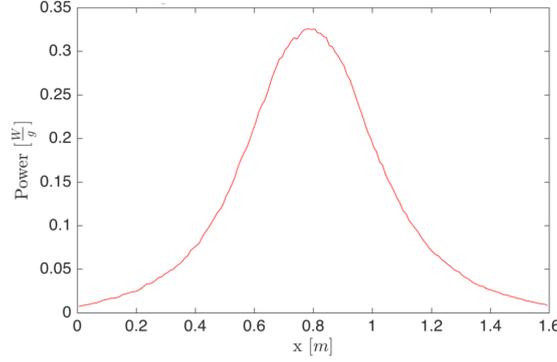


Figure 4-12: Gamma heat deposition in construction material plotted against the position in the DLDR (Pendse [2018]).

- The Reynolds number describes the ratio of viscous and inertia forces in a gas or fluid flow. For high Reynolds numbers inertia forces dominate, causing the flow to be turbulent as opposed to laminar.

$$Re = \frac{\text{inertia forces}}{\text{viscous forces}} = \frac{\rho \langle v \rangle \mathcal{L}}{\mu}. \quad (18)$$

$\langle v \rangle$ [m/s] is the average flow speed, μ [kg/(ms)] is the dynamic viscosity and \mathcal{L} [m] is the characteristic linear dimension. For flow in a tube, this is equal to the hydraulic diameter $\mathcal{L} = D_h$.

- The Nusselt number describes the ratio of the conductive heat transfer and the total heat transfer. It is used for heat transfer to flowing media. At Nusselt numbers of the order 1, conductive heat transfer dominates.

$$Nu_{\mathcal{L}} = \frac{\text{total heat transfer}}{\text{conductive heat transfer}} = \frac{h \cdot \mathcal{L}}{\lambda}, \quad (19)$$

where h [W/(m²K)] is the heat transfer coefficient and λ [W/(mK)] the thermal conductivity. For natural convection around horizontal and vertical cylinders, the spatial dimension is equal to respectively the cylinder diameter $\mathcal{L} = D$ and cylinder length $\mathcal{L} = L$. For flow in a tube it is equal to the hydraulic diameter $\mathcal{L} = D_h$.

- The Graetz number represents the ratio between convective and conductive heat transfer and is also used for heat transfer to flowing media.

$$Gz = \frac{\text{conductive heat transfer}}{\text{convective heat transfer}} = \frac{aL}{v \cdot D_h^2}, \quad (20)$$

in which a [m²/s] is the thermal diffusivity, L [m] the length and v [m/s] is the velocity.

- The Prandtl number is the ratio between thermal and momentum diffusivity. In this list it is the only material characteristic, contrary to flow characteristics.

$$Pr = \frac{\text{momentum diffusivity}}{\text{thermal diffusivity}} = \frac{\nu}{a}. \quad (21)$$

- The Grashof number describes the ratio between the viscous force and the buoyancy force caused by density differences in fluids or gasses in vertical direction. This density difference induces a natural convection flow if the buoyancy is larger than the viscous force.

$$Gr_{\mathcal{L}} = \frac{\text{buoyancy force}}{\text{viscous force}} = \frac{g\beta |T_w - T_{\infty}| \mathcal{L}^3}{\nu^2}, \quad (22)$$

in which $g = 9.81 \text{ m}^2/\text{s}$ is the gravitational constant and $\beta [K^{-1}]$ is the thermal expansion coefficient. T_w and T_{∞} are the heated material and surrounding temperatures. \mathcal{L} is the dimension in vertical direction. For horizontal and vertical cylinders respectively, it denotes the cylinder diameter $\mathcal{L} = D$ and length $\mathcal{L} = L$.

- The Rayleigh number translates the Grashof number into heat transfer: it is the ratio between the diffusive and the natural convective heat transfer.

$$Ra_{\mathcal{L}} = \frac{\text{natural convective heat transfer}}{\text{diffusive heat transfer}} = Gr \cdot Pr = \frac{g\beta |T_w - T_{\infty}| \mathcal{L}^3}{\nu a}, \quad (23)$$

where Ra, Gr and Pr are rewritten using $a = \frac{\lambda}{\rho c_p}$, $\nu = \frac{\mu}{\rho}$.

4.4 Friction

In the research loop, the flow is impacted by friction with the tube wall, which confines the flow speed.³ Two common friction factors are used in transport phenomena: the Fanning and the Darcy friction factor. These are related by a factor 4:

$$f_D = 4 \cdot f_F. \quad (24)$$

The Darcy friction factor in a pipe is given by the following relationship that holds for both laminar and turbulent flow:

$$\begin{aligned} f_D &= \left(\frac{64}{\text{Re}}\right)^a [0.75 \ln\left(\frac{\text{Re}}{5.37}\right)]^{2(a-1)b} [0.88 \ln\left(6.82 \frac{D}{\epsilon}\right)]^{2(a-1)(1-b)}, \\ a &= \frac{1}{1 + \left(\frac{\text{Re}}{2712}\right)^{8.4}}, \\ b &= \frac{1}{1 + \left(\frac{\text{Re}}{150 \frac{D}{\epsilon}}\right)^{1.8}}, \end{aligned} \quad (25)$$

in which ϵ/D is the relative roughness of the pipe. The latter is defined as the effective roughness $\epsilon [m]$ divided by the diameter $D [m]$. Re is the Reynolds number (Bellos et al. [2018]).

The frictional force in a tube exerted on the fluid is given by the following equations:

$$\tau_{f \rightarrow w} = f_F \cdot \frac{1}{2} \rho \langle v \rangle^2, \quad (26)$$

$$F_{\text{fr}} = \tau_{w \rightarrow f} \cdot SL, \quad (27)$$

in which $\tau_{f \rightarrow w} [N/m^2]$ is the shear stress exerted by the inner surface of the tube on the flowing fluid and S is the wetted perimeter.

³Theoretically, this friction in a tube of length L also produces heat according to $e_{fr} = 4f_F \cdot \frac{L}{D_h} \cdot \frac{1}{2} \langle v \rangle^2$, where e_{fr} is defined in $[J/(kg \cdot s)]$. This heat generation is negligible, however: it showed no contribution to the results and was therefore omitted from the model.

4.5 Flow of heat

Flow of heat through various adjacent media is described by *Newton's law of cooling*:

$$\phi_q = UA\Delta T, \quad (28)$$

in which A is the surface through which the heat flows and ΔT is the temperature difference between the inner and outer medium. U [$W/(m^2K)$] is the total heat transfer coefficient, given the individual heat transfer coefficients of the adjacent media h_1, h_2, h_3 etc.:

$$U = \left(\frac{1}{h_1} + \frac{1}{h_2} + \frac{1}{h_3} + \dots \right)^{-1}. \quad (29)$$

These individual heat transfer coefficients are either directly known or can be found by combining the definition for the Nusselt number (equation (19)) with empirical Nusselt relations for different situations.

Flow in a tube

Heat flows in and out of the MPML and the heat exchanger through tube walls. For flow in a tube, the heat transfer coefficient between the tube wall and the flowing medium inside the tube depends on the flow regime (Janssen and Warmoeskerken [1987]):

$$\langle \text{Nu} \rangle = \begin{cases} 0.027 \cdot \text{Re}^{0.8} \cdot \text{Pr}^{0.33}, & (\text{for } \text{Re} > 10^4, \text{Pr} \geq 0.7) \\ 1.62 \cdot \text{Gz}^{-\frac{1}{3}}, & (\text{for } \text{Gz} < 0.05) \\ 3.66. & (\text{for } \text{Gz} > 0.1) \end{cases} \quad (30)$$

Natural convection around cylinders

The MPML roughly consists of four tubes that are at a different temperature than the surrounding water. Therefore, natural convection around these horizontal and vertical tubes is induced. The heat producing cylinders cause natural convection around the cylinder if the Grashof number is sufficiently large. The sign of the temperature difference between the cylinder and the water does not affect the Grashof number. The sign does, however, indicate the direction of natural convection and the direction of heat exchange between water and cylinder, as becomes clear from equation (28). A horizontal cylinder warmer than the water, heats the surrounding water and thereby lowers its density, causing convection in positive z -direction. A cold cylinder, however, extracts heat from the water, which therefore starts flowing in negative z -direction. Similarly, a warm or cold vertical cylinder induces the same flow directions albeit parallel to the cylinder instead of perpendicular.

As outlined by Akbari et al. [2019] and Ali and Sadek [2018], many different results for different cylinder types and inclination angles and regimes have been obtained. With many different available correlations for heated horizontal cylinders, Fand et al. [1977] makes a comparison. It is for instance pointed out that the Morgan correlation differs from many others, suggesting that Nusselt is a function of the Rayleigh number only. In contrast, other correlations take the Prandtl number into account in a separate term. The latter is shown to be more accurate.

One of the remaining options is the widely used and recommended correlation of Churchill and Chu [1975], which is shown to work well for water but shows a larger deviation for other fluids and gasses.⁴ This correlation is practical because it is valid for both the laminar and turbulent regime.

⁴For all the combined data found by Fand et al, including water and other fluids and gasses, the deviation from experimental data is 12%.

Moreover it is applicable to this experiment because water is used as a fluid around the cylinders. The following relation is used to simulate natural convection around horizontal cylinders for the laminar and turbulent regime (Churchill and Chu [1975]):

$$\text{Nu}_D = \left[0.6 + \frac{0.387\text{Ra}_D^{1/6}}{\left(1 + \frac{0.559^{9/16}}{\text{Pr}}\right)^{8/27}} \right]^2, \quad 10^{-5} < \text{Ra}_D < 10^{12}. \quad (31)$$

For natural convection around heated vertical cylinders, the correlation provided by Xian et al. [2015] is used due to its practicality: it holds for both the laminar and turbulent regime.

$$\begin{aligned} \log_{10} \left(\frac{\text{Nu}_L}{\text{Ra}_L^{1/4}} \right) &= 0.090 - 0.449 \log_{10} \left(\text{Ra}_L^{1/4} \cdot \frac{D}{L} \right) \\ &+ 0.107 \left(\log_{10} \left(\text{Ra}_L^{1/4} \cdot \frac{D}{L} \right) \right)^2 + 0.065 \left(\log_{10} \left(\text{Ra}_L^{1/4} \cdot \frac{D}{L} \right) \right)^3, \end{aligned} \quad (32)$$

holding for

$$10^8 < \text{Ra}_L < 1.45 \times 10^{14}, 10 < \frac{L}{D} < 500, \text{ water as working fluid.}$$

These two conditions are both met in this research. Although equation (32) is practical, it is not (yet) widely used. To verify its accuracy, it is compared to the widely used correlation of Le Fevre that only holds for the laminar regime. As discovered by Popiel [2008], the critical Grashof number for the transition from laminar to turbulent is $Gr_c = 4 \cdot 10^9$. This gives the upper bound for the Le Fevre correlation (Popiel [2008]):

$$\text{Nu}_L = \frac{4}{3} \text{Ra}_L^{0.25} \left[\frac{7\text{Pr}}{100 + 105\text{Pr}} \right]^{0.25} + \frac{4}{35} \frac{272 + 315\text{Pr}}{64 + 63\text{Pr}} \frac{L}{D}, \quad 10^8 < Gr_L < 4 \cdot 10^9. \quad (33)$$

Conduction around a cylinder

Heat transfer between the loop and the surroundings is described by natural convection around cylinders. An exception occurs when the temperature difference between the loop and the surroundings is small. If the temperature difference does not cause a buoyancy force large enough to conquer the viscous force, then natural convection does not occur. Heat transfer between a horizontal cylinder and a stationary surrounding is described by conduction. Since this only occurs as an exception, however, the derivation for the Nusselt relation is given in the appendix.

Heat exchanger

Flow of heat within a concentric tube heat exchanger differs from equation (28). It is given by (Van den Akker and Mudde [2014]):

$$\phi_q = UA \frac{\Delta T_L - \Delta T_0}{\ln(\Delta T_L / \Delta T_0)} = UA \frac{\Delta T_0 - \Delta T_L}{\ln(\Delta T_0 / \Delta T_L)} = UA \Delta T_{\text{ln}}, \quad (34)$$

in which ΔT_0 and ΔT_L are the temperature differences between the inner and outer tube fluid at the inlet and outlet of the heat exchanger with length L .

The heat resistance of the tube wall inside the heat exchanger is simulated as a thin wall. For a

thin wall in cylindrical coordinates of thickness d and inner radius r , the heat transfer coefficient is approximated by that for a one-dimensional wall:

$$h = \frac{d_{tube}}{\lambda_{tube}}, \quad (35)$$

where d_{tube} [m] and λ_{tube} [$W/(mK)$] are the thickness and thermal conductivity of the tube wall.

5 Mathematical model

In this section, the differential equations are derived that define the model mathematically.

5.1 Boussinesq approximation

In steady state conditions, the four loop nodes are at different temperatures and therefore have different densities. The density field that describes the density throughout the loop $\rho_0(\vec{r})$ only varies in space. With \vec{r} defining the grid points, namely the centers of the nodes, the density field is given by⁵:

$$\rho_0(\vec{r}) = \begin{pmatrix} \rho_{ref,A} \\ \rho_{ref,D} \\ \rho_{ref,B} \\ \rho_{ref,E} \end{pmatrix}, \quad \vec{r} = \begin{pmatrix} r_A \\ r_D \\ r_B \\ r_E \end{pmatrix}. \quad (36)$$

Time dependent density variations are only taken into account in the gravitational term of the momentum balance, while being omitted from all other terms in the momentum, energy and mass balance. The above density field therefore describes the density for all other terms in both stationary and transient simulations. Since the density differences are small, however, the reference densities are approximated by one average reference density:

$$\rho_{ref,A} = \rho_{ref,D} = \rho_{ref,B} = \rho_{ref,E} = \rho_{ref,avg}, \quad (37)$$

which is assumed to be the density at 60°C in this research. This value was chosen based on the temperature profile resulting from the final simulation. As a result, the density becomes constant in both space and time for all terms except for the gravitational term in the momentum balance. From here on, $\rho_{ref,avg}$ is used as this reference density: assumption (36) is from here on applied.

5.2 Mass balance

The transient mass balance (equation (6)) can be rewritten using equation (7). The temperature and density are assumed to be constant in space within a node. The mass balance over a loop node j becomes:

$$V_j \cdot \frac{d\rho_{ref,j}(t)}{dt} = \rho_{ref,j}(t) \cdot A \cdot \langle v_{j,in}(t) \rangle - \rho_{ref,j}(t) \cdot A \cdot \langle v_{j,out}(t) \rangle.$$

The second assumption of the Boussinesq approximation states that the transient term $\rho'(\vec{r}, t)$ should be neglected in the mass balance. The density is described by a reference density for the node and the transient term vanishes:

$$0 = \rho_{ref,j} \cdot A \cdot \langle v_{j,in}(t) \rangle - \rho_{ref,j} \cdot A \cdot \langle v_{j,out}(t) \rangle. \quad (38)$$

Since the cross-sectional area is constant, it follows for every loop node j that the velocity is constant within the node: $\langle v_{j,in}(t) \rangle = \langle v_{j,out}(t) \rangle = \langle v_j(t) \rangle$. If the reference densities are different for every node, the velocity also differs per node.

Applying assumption (37), however, implies that both the density and the cross-sectional area are constant over the loop. It therefore follows that the velocity only varies in time: $\langle v_A(t) \rangle = \langle v_D(t) \rangle = \langle v_B(t) \rangle = \langle v_E(t) \rangle = \langle v(t) \rangle$.

⁵The grid point positions r_A, r_D, r_B, r_E are not to be confused with the radius r .

5.3 Internal energy balance

In different nodes, the net flow of heat ϕ_q and the production of internal energy P_u vary. The production of internal energy only takes place in node A in the form of heat generation due to fission (denoted by Q). Contrary to the simulation of Dresen [2019], gamma heating is modelled in the tube wall and is therefore omitted from the production term. The net flow can consist of flow within the heat exchanger from node B to node C (denoted by $\phi_{q,H}$), and flow towards or from the tube wall adjacent the DLDR water (denoted by $\phi_{q,D}$).

According to the Boussinesq approximation, the transient density term should be neglected in the internal energy balance. Therefore the density is constant within a node and the pressure term vanishes from equation (8). Combining the above information with the simplified internal energy balance (equation (8)), the mass flow (equation (7)) and the equation of states (12) gives the general internal energy balance for the nodes:

$$\begin{aligned} \rho_{ref,i} \cdot V c_P \frac{d}{dt}(T_{i,out}) &= \rho_{ref,i} \cdot \pi r^2 \langle v(t) \rangle \cdot c_p (T_{i,in}(t) - T_{i,out}(t)) + \phi_q + P_u, \\ \phi_q &= \phi_{q,D} + \phi_{q,H}, \\ P_u &= Q. \end{aligned} \quad (39)$$

where $\rho_{ref,i} = \rho_{ref,avg}$ for the loop nodes j and $\rho_{ref,i} = \rho_{ref,C}$ for node C . The velocity and reference density are assumed to be constant in space over the loop by assumption (37). The internal energy balances over the nodes in the direction of flow are given below: A, D, B, E and the heat exchanger node C .

5.3.1 Internal energy balance node A

Node A is the only node in which heat production by fission is relevant. Moreover, heat generation by friction occurs in every node. Filling in the temperatures the internal energy balance becomes:

$$\begin{aligned} \rho_{ref,avg} \cdot V_j c_P \frac{d}{dt}(T_A(t)) &= \rho_{ref,avg} \cdot \pi r^2 \langle v(t) \rangle \cdot c_p (T_E(t) - T_A(t)) + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,A} = 2\pi r l \cdot h_{1,A} \cdot (T_{w,A}(t) - T_A(t)), \\ P_u &= Q, \end{aligned} \quad (40)$$

in which $V_j = \pi r^2 l$ is the node volume that is equal for all loop nodes j . $h_{1,A}$ denotes the heat transfer coefficient from the uranyl nitrate solution to the tube wall, which is found by combining the definition for the Nusselt number (equation (19)) with the Nusselt relation for flow in a tube (equation (30)).

5.3.2 Internal energy balance node D

In node D , only friction and heat exchange with the tube wall are relevant. The energy balance becomes:

$$\begin{aligned} \rho_{ref,avg} \cdot V_j c_P \frac{d}{dt}(T_D(t)) &= \rho_{ref,avg} \cdot \pi r^2 \langle v(t) \rangle \cdot c_p (T_A(t) - T_D(t)) + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,D} = 2\pi r l \cdot h_{1,D} \cdot (T_{w,D}(t) - T_D(t)), \\ P_u &= 0. \end{aligned} \quad (41)$$

5.3.3 Internal energy balance node B

In node B the only heat exchange is within the heat exchanger. This heat flow is negative because heat is extracted towards the coolant and is given by equation (34). The internal energy balance

becomes:

$$\begin{aligned}\rho_{ref,avg} \cdot V_j c_P \frac{d}{dt}(T_B(t)) &= \rho_{ref,avg} \cdot \pi r^2 \langle v(t) \rangle \cdot c_p (T_D(t) - T_B(t)) + \phi_q + P_u, \\ \phi_q &= -\phi_{q,H} = -UA\Delta T_{in}, \\ P_u &= 0.\end{aligned}\quad (42)$$

Here ΔT_{in} is given by equation (34) and the total heat transfer coefficient is given by

$$U = \left(\frac{1}{h_{1,B}} + \frac{d_W}{\lambda_w} + \frac{1}{h_{1,C}} \right)^{-1}, \quad (43)$$

in which the three heat resistances account for respectively the uranyl nitrate solution in node B as given by equation (30), the MPML tube material (equation (35)) and the coolant in node C (equation (30)). The surface in the heat exchanger is given by $A = 2\pi rl$ and the temperature differences in equation (34) are $\Delta T_0 = T_B - T_{in,C}$ and $\Delta T_L = T_D - T_{out,C}$.

5.3.4 Internal energy balance node E

The balance is analogous to that of node D :

$$\begin{aligned}\rho_{ref,avg} \cdot V c_P \frac{d}{dt}(T_E(t)) &= \rho_{ref,avg} \cdot \pi r^2 \langle v(t) \rangle \cdot c_p (T_B(t) - T_E(t)) + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,E} = 2\pi rl \cdot h_{1,E} \cdot (T_{w,E}(t) - T_E(t)), \\ P_u &= 0.\end{aligned}\quad (44)$$

5.3.5 Internal energy balance node C

The coolant is heated in two ways: by a positive heat flow from node B and by heat exchange with the wall of node C . The energy balance becomes:

$$\begin{aligned}\rho_{ref,C} \cdot V_C \cdot c_{P,C} \frac{d}{dt}(T_{out,C}(t)) &= \rho_{ref,C} \cdot \pi(R^2 - (r+d)^2) \langle v_C \rangle \cdot c_{p,C} (T_{in,C} - T_{out,C}(t)) \\ &\quad + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,C} + \phi_{q,H} \\ &= 2\pi Rl \cdot h_{1,C} \cdot (T_{w,C}(t) - \frac{T_{in,C} + T_{out,C}(t)}{2}) + UA\Delta T_{in}, \\ P_u &= 0.\end{aligned}\quad (45)$$

Note that $T_{in,C}$ is constant, as it is a parameter and that the cross section used to calculate the mass flow is equal to $\pi(R^2 - (r+d)^2)$, as becomes clear from figure 3-10. The volume of node C is given by:

$$V_C = \pi(R^2 - (r+d)^2)l. \quad (46)$$

$h_{1,A}$ denotes the heat transfer coefficient from the uranyl nitrate solution to the tube wall, which is found by combining the definition for the Nusselt number (equation (19)) with the Nusselt relation for flow in a tube (equation (30)), Here, the Reynolds number is computed using the hydraulic diameter (equation (10)), which is rewritten as:

$$D_h = \frac{4 \cdot \frac{\pi(D_C^2 - D_{out}^2)}{4}}{\pi(D_C + D_{out})} = \frac{D_C^2 - D_{out}^2}{D_C + D_{out}} = \frac{(D_C - D_{out})(D_C + D_{out})}{D_C + D_{out}} = D_C - D_{out}.$$

5.3.6 Internal energy balances over the tube walls

Before deriving the internal energy balances over the tube walls, it is first shown that the heat resistance of the tube wall is negligible. As shown in figure 2-6, this is denoted by $h_2 \gg h_1, h_3$. Table 4 shows the different heat transfer coefficients for the cases that h_1 and h_3 are maximal according to Dresen [2019]⁶ and for moderate conditions used in this research. For extreme and moderate conditions respectively, it follows from table 4 that $h_2 > 5 \cdot h_1, h_3$ and $h_2 > 10 \cdot h_1, h_3$.

Conditions	Heat transfer coefficient	$W/(m^2K)$
$\phi_{m,C} = 0.1(kg/s), T = 90^\circ C, T_{out,C} = 30^\circ C$	h_1	$6 \cdot 10^2$
	$h_{1,C}$	$2 \cdot 10^3$
	$h_{3,hor}$	$1.6 \cdot 10^3$
	$h_{3,ver}$	6
	$h_2 = \lambda_t/d$	$1 \cdot 10^4$
$\phi_{m,C} = 0.01(kg/s) T = 50^\circ C, T_{out,C} = 15^\circ C$	h_1	$6 \cdot 10^2$
	$h_{1,C}$	$9 \cdot 10^2$
	$h_{3,hor}$	$1.0 \cdot 10^3$
	$h_{3,ver}$	6
	$h_2 = \lambda_t/d$	$1 \cdot 10^4$

Table 4: Heat transfer coefficients for extreme and moderate conditions. It follows that in both cases, $h_2 \gg h_1, h_3$

Since the heat transfer coefficient in the tube wall is at least a factor five larger than the heat transfer coefficients outside the tube wall, a temperature difference between the inside and the outside part of the tube wall is restored approximately instantly. Therefore the tube temperature is assumed to be spatially constant in the radial direction. As a result, the tube wall temperature is constant in space within a node, varying over time.

Since the tube is a solid material, the mass convection term vanishes from equation (8) and the internal energy balance over the tube wall surrounding node A becomes:

$$\begin{aligned}
 \rho_t \cdot V_t(r) \cdot c_{P,t} \frac{d}{dt}(T_{w,A}(t)) &= \phi_q + P_u, \\
 \phi_q &= 2\pi r l \cdot h_{1,hor,A} \cdot (T_A(t) - T_{w,A}(t)) \\
 &\quad - 2\pi(r+d)l \cdot h_{3,A} \cdot (T_{w,A}(t) - T_s), \\
 P_u &= P_{gamma}(r),
 \end{aligned} \tag{47}$$

in which the two heat flow terms are derived from equation (28), h_1 is given by equation (30) and $h_{3,hor,A}$ is given by combining the Nusselt relation for natural convection around a horizontal cylinder equation (31) with the Nusselt definition (equation (19)). The tube volume $V_t(r)$ is given by equation (17).

The internal energy balances over the walls surrounding nodes C, D and E are different in three ways. Firstly, gamma-radiation may be shielded completely or partially. Therefore the gamma radiation heat production is multiplied by a shielding transmission factor between 0 and 1, $stf = 1$

⁶Since Dresen [2019] $\phi_{m,C} = 0.7kg/s$ is not unattainable in practise, $\phi_{m,C} = 0.1kg/s$ is regarded as the most extreme case.

meaning that all gamma radiation is transmitted and thus no gamma shielding is present. Secondly, the tube volume of the wall surrounding node C is larger since it has a larger radius $R > r$. This also increases the gamma radiation heating. Thirdly, h_3 may represent natural convection around a horizontal cylinder (equation (31)) or vertical cylinder (equation (32)). The balances over the walls surrounding nodes D , E and C respectively become:

$$\begin{aligned}
\rho_t \cdot V_t(r) \cdot c_{P,t} \frac{d}{dt}(T_{w,D}(t)) &= \phi_q + P_u, \\
\phi_q &= 2\pi r l \cdot h_{1,D} \cdot (T_D(t) - T_{w,D}(t)) \\
&\quad - 2\pi(r+d)l \cdot h_{3,ver,D} \cdot (T_{w,D}(t) - T_s), \\
P_u &= stf \cdot P_{gamma}(r).
\end{aligned} \tag{48}$$

$$\begin{aligned}
\rho_t \cdot V_t(r) \cdot c_{P,t} \frac{d}{dt}(T_{w,E}(t)) &= \phi_q + P_u, \\
\phi_q &= 2\pi r l \cdot h_{1,E} \cdot (T_E(t) - T_{w,E}(t)) \\
&\quad - 2\pi(r+d)l \cdot h_{3,ver,E} \cdot (T_{w,E}(t) - T_s), \\
P_u &= stf \cdot P_{gamma}(r).
\end{aligned} \tag{49}$$

$$\begin{aligned}
\rho_t \cdot V_t(R) \cdot c_{P,t} \frac{d}{dt}(T_{w,C}(t)) &= \phi_q + P_u, \\
\phi_q &= 2\pi R l \cdot h_{1,C} \cdot \left(\frac{T_{in,C} + T_{out,C}(t)}{2} - T_{w,C}(t) \right) \\
&\quad - 2\pi(R+d)l \cdot h_{3,hor,C} \cdot (T_{w,C}(t) - T_s), \\
P_u &= stf \cdot P_{gamma}(R).
\end{aligned} \tag{50}$$

5.3.7 Negative velocities

In order to simulate a negative flow direction, a number of adjustments must be made to the internal energy balances over the nodes.

- Temperatures in the convection term change: the node temperature should still be equal to to outlet temperature, but the inlet temperature is now the temperature on the other side of the node. The temperature difference in the energy balance for node A , for instance, changes from $(T_E(t) - T_A(t))$ to $(T_D(t) - T_A(t))$.
- The velocity becomes the absolute value of the velocity, since the friction term should for instance always be positive and the direction for convection is taken into account by the temperature definitions. Other terms such as the Reynolds number also need a positive velocity value.
- The heat exchanger becomes co-current. That gives $\Delta T_0 = T_E - T_{in,C}$ and $\Delta T_L = T_B - T_{out,C}$ in equation (34).

The balances for the nodes in the direction of the flow A, E, B, E and C become:

$$\begin{aligned}\rho_{ref,avg} \cdot V_j c_P \frac{d}{dt}(T_A(t)) &= \rho_{ref,avg} \cdot \pi r^2 |\langle v(t) \rangle| \cdot c_p (T_D(t) - T_A(t)) + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,A} = 2\pi r l \cdot h_{1,A} \cdot (T_{w,A}(t) - T_A(t)), \\ P_u &= Q.\end{aligned}\quad (51)$$

$$\begin{aligned}\rho_{ref,avg} \cdot V_j c_P \frac{d}{dt}(T_E(t)) &= \rho_{ref,avg} \cdot \pi r^2 |\langle v(t) \rangle| \cdot c_p (T_A(t) - T_E(t)) + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,E} = 2\pi r l \cdot h_{1,E} \cdot (T_{w,E}(t) - T_E(t)), \\ P_u &= 0.\end{aligned}\quad (52)$$

$$\begin{aligned}\rho_{ref,avg} \cdot V_j c_P \frac{d}{dt}(T_B(t)) &= \rho_{ref,avg} \cdot \pi r^2 |\langle v(t) \rangle| \cdot c_p (T_E(t) - T_B(t)) + \phi_q + P_u, \\ \phi_q &= -\phi_{q,H} = -UA\Delta T_{in}, \\ P_u &= 0.\end{aligned}\quad (53)$$

$$\begin{aligned}\rho_{ref,avg} \cdot V_j c_P \frac{d}{dt}(T_D(t)) &= \rho_{ref,avg} \cdot \pi r^2 |\langle v(t) \rangle| \cdot c_p (T_B(t) - T_D(t)) + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,D} = 2\pi r l \cdot h_{1,D} \cdot (T_{w,D}(t) - T_D(t)), \\ P_u &= 0.\end{aligned}\quad (54)$$

$$\begin{aligned}\rho_{ref,C} \cdot V_C \cdot c_{P,C} \frac{d}{dt}(T_{out,C}(t)) &= \rho_{ref,C} \cdot \pi (R^2 - (r+d)^2) \langle v_C \rangle \cdot c_{p,C} (T_{in,C} - T_{out,C}(t)) \\ &\quad + \phi_q + P_u, \\ \phi_q &= \phi_{q,D,C} + \phi_{q,H} \\ &= 2\pi R l \cdot h_{1,C} \cdot (T_{w,C}(t) - \frac{T_{in,C} + T_{out,C}(t)}{2}) + UA\Delta T_{in}, \\ P_u &= 0,\end{aligned}\quad (55)$$

where ΔT_{in} is found from $\Delta T_0 = T_E - T_{in,C}$ and $\Delta T_L = T_B - T_{out,C}$ in equation (34). Besides, all relations dependent on the velocity such as the Reynolds number (equation (18)) and the friction factor (equation (25)) are found using the absolute value of the velocity.

5.4 Momentum balance

Combining equation (7) for flow of mass and equation (9) for the momentum balance leads to the following momentum balance for a node j inside the loop. The direction \hat{l} is defined as the anti-clockwise direction along the loop. The momentum balance for node j becomes:

$$\rho_{ref,j} \cdot V_j \frac{d}{dt}(\langle v_j(t) \rangle) = (\rho_{ref,j-1} - \rho_{ref,j}) \cdot A \cdot \langle v_j(t) \rangle^2 + \sum F_{\hat{l}}. \quad (56)$$

There are three forces: the gravitational force, the friction force and pressure. Theoretically, the velocity and these forces work in the $\pm\hat{x}$ - and $\pm\hat{z}$ -direction. To simplify the notation, however, the problem can be made one-dimensional. In order to find the integrated momentum balance over the loop, all forces are summed in the direction of the loop, since it is a closed system. Therefore, the forces and velocity are defined along the \hat{l} -axis anti-clockwise along the loop, as depicted in figure 5-13.

The friction force is given by equation (27) and works in the $-\hat{l}$ direction in every node. The gravitational force predominantly works in the vertical parts: in the $-\hat{l}$ direction in node D and in the $+\hat{l}$ direction in node E . In the near-horizontal parts, the gravitational forces depend on the inclination angle α . Note that the densities in the gravitational terms are the only time-dependent densities, as prescribed by the Boussinesq approximation. The momentum balances for respectively

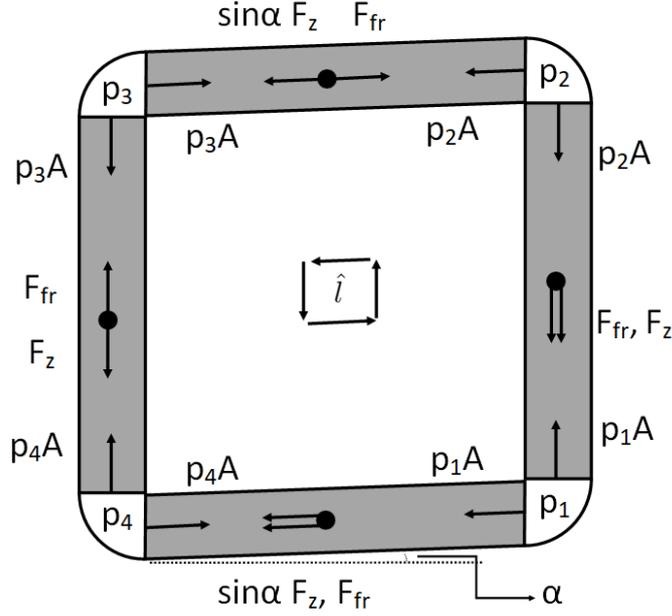


Figure 5-13: The direction of all the forces contributing to the momentum balance.

nodes A, D, B and E become:

$$\begin{aligned}
\rho_{ref,A} \cdot V_j \frac{d}{dt} (\langle v_A(t) \rangle) &= (\rho_{ref,E} - \rho_{ref,A}) \cdot A \cdot \langle v_A(t) \rangle^2 - f_D \cdot \frac{1}{8} \rho_{ref,A} \langle v_A(t) \rangle^2 \cdot 2\pi r l \\
&\quad - \pi r^2 l g \cdot \rho(T_A(t)) \cdot \sin \alpha + A (p_4 - p_1), \\
\rho_{ref,D} \cdot V_j \frac{d}{dt} (\langle v_D(t) \rangle) &= (\rho_{ref,A} - \rho_{ref,D}) \cdot A \cdot \langle v_D(t) \rangle^2 - f_D \cdot \frac{1}{8} \rho_{ref,D} \langle v_D(t) \rangle^2 \cdot 2\pi r l \\
&\quad - \pi r^2 l g \cdot \rho(T_D(t)) + A (p_1 - p_2), \\
\rho_{ref,B} \cdot V_j \frac{d}{dt} (\langle v_B(t) \rangle) &= (\rho_{ref,D} - \rho_{ref,B}) \cdot A \cdot \langle v_B(t) \rangle^2 - f_D \cdot \frac{1}{8} \rho_{ref,B} \langle v_B(t) \rangle^2 \cdot 2\pi r l \\
&\quad + \pi r^2 l g \cdot \rho(T_B(t)) \cdot \sin \alpha + A (p_2 - p_3), \\
\rho_{ref,E} \cdot V_j \frac{d}{dt} (\langle v_E(t) \rangle) &= (\rho_{ref,B} - \rho_{ref,E}) \cdot A \cdot \langle v_E(t) \rangle^2 - f_D \cdot \frac{1}{8} \rho_{ref,E} \langle v_E(t) \rangle^2 \cdot 2\pi r l \\
&\quad + \pi r^2 l g \cdot \rho(T_E(t)) + A (p_3 - p_4),
\end{aligned} \tag{57}$$

in which the temperature dependent densities $\rho(T_j(t))$ in the gravitational terms are found by the equation of states from the Boussinesq approximation (equation (12)). Applying assumption (37), an average reference density is applied for every node and the velocity thereby becomes constant in space over the loop. The assumption causes a maximum error of 1.6% in the affected term, as outlined in appendix B. The four balances become:

$$\begin{aligned}
\rho_{ref,avg} \cdot V_j \frac{d}{dt} (\langle v(t) \rangle) &= -f_D \cdot \frac{1}{8} \rho_{ref,avg} \langle v(t) \rangle^2 \cdot 2\pi r l - \pi r^2 l g \cdot \rho(T_A(t)) \cdot \sin \alpha \\
&\quad + A (p_4 - p_1), \\
\rho_{ref,avg} \cdot V_j \frac{d}{dt} (\langle v(t) \rangle) &= -f_D \cdot \frac{1}{8} \rho_{ref,avg} \langle v(t) \rangle^2 \cdot 2\pi r l - \pi r^2 l g \cdot \rho(T_D(t)) \\
&\quad + A (p_1 - p_2), \\
\rho_{ref,avg} \cdot V_j \frac{d}{dt} (\langle v(t) \rangle) &= -f_D \cdot \frac{1}{8} \rho_{ref,avg} \langle v(t) \rangle^2 \cdot 2\pi r l + \pi r^2 l g \cdot \rho(T_B(t)) \cdot \sin \alpha \\
&\quad + A (p_2 - p_3), \\
\rho_{ref,avg} \cdot V_j \frac{d}{dt} (\langle v(t) \rangle) &= -f_D \cdot \frac{1}{8} \rho_{ref,avg} \langle v(t) \rangle^2 \cdot 2\pi r l + \pi r^2 l g \cdot \rho(T_E(t)) \\
&\quad + A (p_3 - p_4).
\end{aligned} \tag{58}$$

The integrated momentum balance over the loop can now be found by adding the above balances. As a result, only frictional and gravitational terms remain on the right hand side. The cumulative and frictional terms can be summed to two terms. The total momentum balance over the loop reduces to:

$$\begin{aligned}
4 \cdot \rho_{\text{ref,avg}} \cdot V_j \frac{d}{dt}(\langle v \rangle(t)) &= -f_D \cdot \frac{1}{2} \rho_{\text{ref,avg}} \langle v(t) \rangle^2 \cdot 2\pi r l \\
&+ \pi r^2 l g \cdot (\rho(T_E(t)) - \rho(T_D(t))) \\
&+ \pi r^2 l g \cdot (\rho(T_B(t)) - \rho(T_A(t))) \cdot \sin \alpha,
\end{aligned} \tag{59}$$

in which V_j is the volume of one node.

5.4.1 Negative velocities

For negative velocities, the sign of all buoyancy forces switch with respect to the velocity direction, while the friction force remains in the negative flow direction. When calculating $\frac{d}{dt}(\langle v \rangle(t))$, the entire right hand side should then be multiplied by a minus sign, in order to obtain negative velocity corrections. The momentum balance for negative velocities becomes:

$$\begin{aligned}
-4 \cdot \rho_{\text{ref,avg}} \cdot V_j \frac{d}{dt}(\langle v \rangle(t)) &= -f_D \cdot \frac{1}{2} \rho_{\text{ref,avg}} |\langle v(t) \rangle|^2 \cdot 2\pi r l \\
&+ \pi r^2 l g \cdot (-\rho(T_E(t)) + \rho(T_D(t))) \\
&+ \pi r^2 l g \cdot (-\rho(T_B(t)) + \rho(T_A(t))) \cdot \sin \alpha.
\end{aligned} \tag{60}$$

6 Numerical method

In this section, a method is outlined to solve the differential equations that were derived in section 5.

6.1 Solving ordinary differential equations

6.1.1 Ordinary differential equations

First order differential equations (ODE's) are characterised by one independent variable, e.g. t , and one or more dependent variables. They have the following form (Newman [2013]):

$$\frac{d\mathbf{r}}{dt} = \mathbf{f}(\mathbf{r}, t), \quad (61)$$

where $\mathbf{r} = (x, y, \dots)$ and $f(r, t) = (f_x(r, t), f_y(r, t), \dots)$ are vectors of respectively the dependent variables and the corresponding (non-)linear functions. A first order one-variable ODE is for instance given by:

$$\frac{dx}{dt} = f(x, t), \quad (62)$$

and a first order two-variable ODE by:

$$\frac{dx}{dt} = f_x(x, y, t), \quad \frac{dy}{dt} = f_y(x, y, t).$$

Higher order ODE's have the same form, except that the derivative is of a higher order. The second and third order ODE are for instance given by respectively:

$$\frac{d^2x}{dt^2} = f\left(x, \frac{dx}{dt}, t\right),$$

$$\frac{d^3x}{dt^3} = f\left(x, \frac{dx}{dt}, \frac{d^2x}{dt^2}, t\right).$$

These higher order ODE's can be solved by writing them as a set of first order ODE's. The second order ODE can for instance be rewritten as follows:

$$\frac{dx}{dt} = y, \quad \frac{dy}{dt} = f(x, y, t).$$

6.1.2 Runge-Kutta method

The Runge-Kutta method is widely used for solving ODE's. It covers a set of solution methods. The different methods are called 'orders' and indicate the accuracy of the method. Higher order methods are more complex and more accurate: the m 'th order Runge-Kutta method is accurate to the order h^m with h being the step size. The error is of the order $O(h^{m+1})$ and is caused by omitting all terms that are of the order $O(h^{m+1})$ and higher. It is therefore called the *truncation error* (Hoogstraten [1985]). When adding the errors over multiple steps, the Runge-Kutta method is however only accurate to order $O(h^{m-1})$ and carries an $O(h^m)$ error: the cumulative error over all steps is one order worse in h than it is for each single step (Newman [2013]).

The fourth order has a high accuracy, is sufficiently simple to program and is therefore widely

used. For an arbitrary number of dependent variables, performing one step of the fourth order Runge-Kutta method is given by the following set of equations (Newman [2013]):

$$\begin{aligned}
\mathbf{k}_1 &= h \cdot \mathbf{f}(\mathbf{r}, t), \\
\mathbf{k}_2 &= h \cdot \mathbf{f}\left(\mathbf{r} + \frac{1}{2}\mathbf{k}_1, t + \frac{1}{2}h\right), \\
\mathbf{k}_3 &= h \cdot \mathbf{f}\left(\mathbf{r} + \frac{1}{2}\mathbf{k}_2, t + \frac{1}{2}h\right), \\
\mathbf{k}_4 &= h \cdot \mathbf{f}(\mathbf{r} + \mathbf{k}_3, t + h), \\
\mathbf{r}(t + h) &= \mathbf{r}(t) + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4),
\end{aligned} \tag{63}$$

in which \mathbf{r} and $f(\mathbf{r}, t)$ are defined the same as in equation (61). Here, $\mathbf{r}(t)$ contains all variable values at $t = t$, and $\mathbf{r}(t + h)$ contains the values at the consecutive time step $t = t + h$.

A disadvantage of this method, however, is that it is an explicit method which may become unstable depending on, amongst others, the step size. To illustrate this, fourth order Runge-Kutta is applied to solving a mass-spring system for varying step sizes in figure 6-14. Clearly the result is divergent for a step size of 5 seconds and becomes stable and more accurate for smaller step sizes.

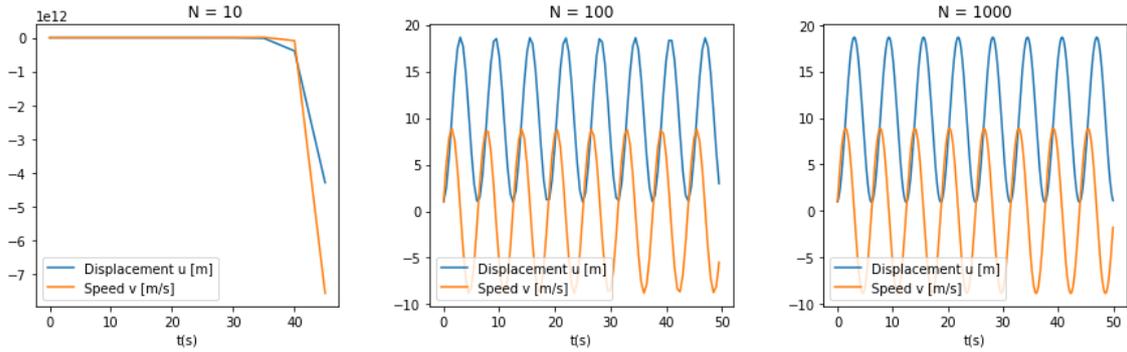


Figure 6-14: Mass-spring system solved using fourth order Runge-Kutta method for varying step sizes. The calculation is unstable for a step size of 5 seconds.

6.1.3 Adaptive Runge-Kutta method

In this section the adaptive fourth order Runge-Kutta method is discussed. The concept of the method is to adapt the step size in every iteration in such a way, that the step size is as large as possible while still having the desired accuracy and stability. This reduces the calculation time in many cases. The key of the method is as follows: every time step, we calculate the error in performing a step with the old step size. From that error, the desired step size is found with which the next time step is calculated.

Suppose an estimate $\hat{x}(t)$ of $x(t)$ was accurately found by using a step size h , starting from $\hat{x}(t-h)$. Before finding the next value $\hat{x}(t+h')$, we want to find the *new step size* h' that causes the *desired error* ϵ' by taking a step of size h' . In order to do so, we first calculate the error ϵ that is caused by taking one step from $\hat{x}(t)$ with the *old step size* h . The new step size can then be found from

the latter error ϵ and the desired accuracy per unit time, which is defined as the ratio between desired error ϵ' and the desired step size h' . It is simply the proportionality constant between the step size and error that is maintained throughout the calculation:

$$\delta = \frac{\epsilon'}{h'}. \quad (64)$$

The error ϵ caused by taking one step of size h from $x(t)$ can be found by estimating $x(t + 2h)$ in two ways: on the one hand by taking two steps of size h and on the other hand by taking one step of size $2h$ starting from $x(t)$. The estimates are denoted x_1 and x_2 respectively. The error of one step h in the fourth order Runge-Kutta method is of order h^5 , so we define $\epsilon = c \cdot h^5$, in which c is a constant. Taking two steps gives an error $2 \cdot c \cdot h^5$. The estimate x_1 , the true value $x(t+2h)$ and the error are therefore related by:

$$x(t + 2h) = x_1 + 2 \cdot c \cdot h^5,$$

and similarly for x_2 in which one step of error $c \cdot (2h)^5$ is taken:

$$x(t + 2h) = x_2 + c \cdot (2h)^5.$$

The error $\epsilon = c \cdot h^5$ that is caused by taking one step of size h follows:

$$\epsilon = \frac{1}{30} (x_1 - x_2). \quad (65)$$

In order to find the new desired step size h' , the error ϵ should become equal to ϵ' :

$$\epsilon' = ch^5 \left(\frac{h'}{h} \right)^5. \quad (66)$$

By firstly calculating the estimates x_1 and x_2 using fourth order Runge-Kutta the new step size can subsequently be found from combining equations (64), (65) and (66). These three equations can be rewritten into one (Newman [2013]):

$$h' = h \left(\frac{30h\delta}{|x_1 - x_2|} \right)^{1/4}. \quad (67)$$

6.1.4 Multi-variable adaptive Runge-Kutta method

For a set of ODE's in which there are various dependent variables x, y, \dots , a choice has to be made on how to calculate equation (67). One possibility is to use the most important dependent variable. Another possibility is to combine them in a common error such as $\sqrt{\epsilon_x^2 + \epsilon_y^2}$. An even more accurate method is to ensure that the step size is always sufficiently small for the variable that changes most rapidly. That can be achieved by finding equation (67) for all the dependent variables and use the smallest step size h' as the new step size. For a multiple-variable $\mathbf{r} = (x, y, \dots)$ set of ODE's, Equation (67) becomes:

$$h' = h \left(\frac{30h\delta}{\max(|x_1 - x_2|, |y_1 - y_2|, \dots)} \right)^{1/4}.$$

A problem occurs in the rare case that the estimates x_1 and x_2 are identical. To prevent an inaccurately large step size increase, an upper bound of $2h$ is used as a rule of thumb:

$$h' = \begin{cases} h \left(\frac{30h\delta}{\max(|x_1 - x_2|, |y_1 - y_2|, \dots)} \right)^{1/4} & h' < 2h \\ 2h & h' > 2h \end{cases} \quad (68)$$

Compared to an 'ordinary' fourth order Runge-Kutta method, there is another alteration: a while loop is used instead of a for loop, because the time array is created as the step size is varied.

In total, performing one fourth order adaptive Runge-Kutta step consists of finding the estimates $x_1, x_2, y_1, y_2, \dots$ using ordinary fourth order Runge-Kutta method and calculating the new step size from equation (68).

The total error should then be equal to the accuracy per unit time δ multiplied by the calculated time: a calculation over $\Delta t = 1000s$ with an accuracy per unit time $\delta = 10^{-4} \text{ }^\circ C/s$ is accurate to $0.1^\circ C$. In this research simulations are performed for 1000 seconds with an accuracy per unit time of at least $1 \cdot 10^{-5} \text{ }^\circ C/s$, which should make the accuracy of the total calculation $10^{-2}^\circ C$. Due to this high accuracy, the total accuracy is hampered by approximation (37) and not by the Adaptive Runge-Kutta method.

Since the above described method requires more calculations per time step, adapting the step size is not helpful if the dependent variables keep fluctuating throughout time. Choosing a small, non-adaptive step size is in that case more efficient. The adaptive method is however ideal for a calculation that involves fluctuations in the start and becomes stationary after a while. This becomes clear from a reduced calculation time when applying the method to these situations.

6.2 Exception handling

In order to maintain the boundary condition (1), an exception is raised if the boundary condition is violated. Before handling the exception, it is important to understand from where the exception originates.

As becomes clear from equation (63), the new value of $T_{out,C}(t + h)$ is calculated as a linear combination of multiplying the step size h with the differential equation $\frac{dT_{out,C}}{dt} = \dots$. If the coolant mass flow and therefore the coolant velocity $\langle v_C \rangle$ is large, the following convective term can become so large that the boundary condition is violated:

$$\frac{\phi_{m,C,in} - \phi_{m,C,out}}{\rho_{ref,C} V_{(R-r)} c_{P,C}} = \frac{\rho_C(t) \pi (R^2 - (r+d)^2) \langle v_C \rangle c_{p,C} (T_{in,C} - T_{out,C}(t))}{\rho_{ref,C} V_{(R-r)} c_{P,C}}$$

The solution, therefore, is to lower the step size and try again.

The above exception is caught using a try-except-finally structure in Python: the code written in the 'try' part is tried. If an exception occurs, the code in the 'except' block is executed. Lastly the code in the 'finally' block is executed, whether or not an exception has occurred. In the case that this exception occurs, the step size is lowered by a factor 0,7 and the step is tried again.⁷

6.3 Flowchart and variable names

The flowchart of the final script is displayed in figure 6-15. In order to prevent double variable names, the following ones are used in the simulation for the radii and variable vector:

- The inner radius r of the loop tubes is denoted by "radius" in python.
- The inner radius R of the node C tube is denoted by "Radius" in python.

⁷The factor 0.7 was chosen through trial and error.

- The vector r containing all dependent variables is denoted by "r" in python (equation (61)).

It is further noted that the initial condition for the velocity is set to 10^{-5} m/s instead of exactly 0 m/s , since a slightly positive value is necessary for calculating the Reynolds number in heat transfer coefficients. Moreover, the angle α is denoted by 'angle' in the Python script.

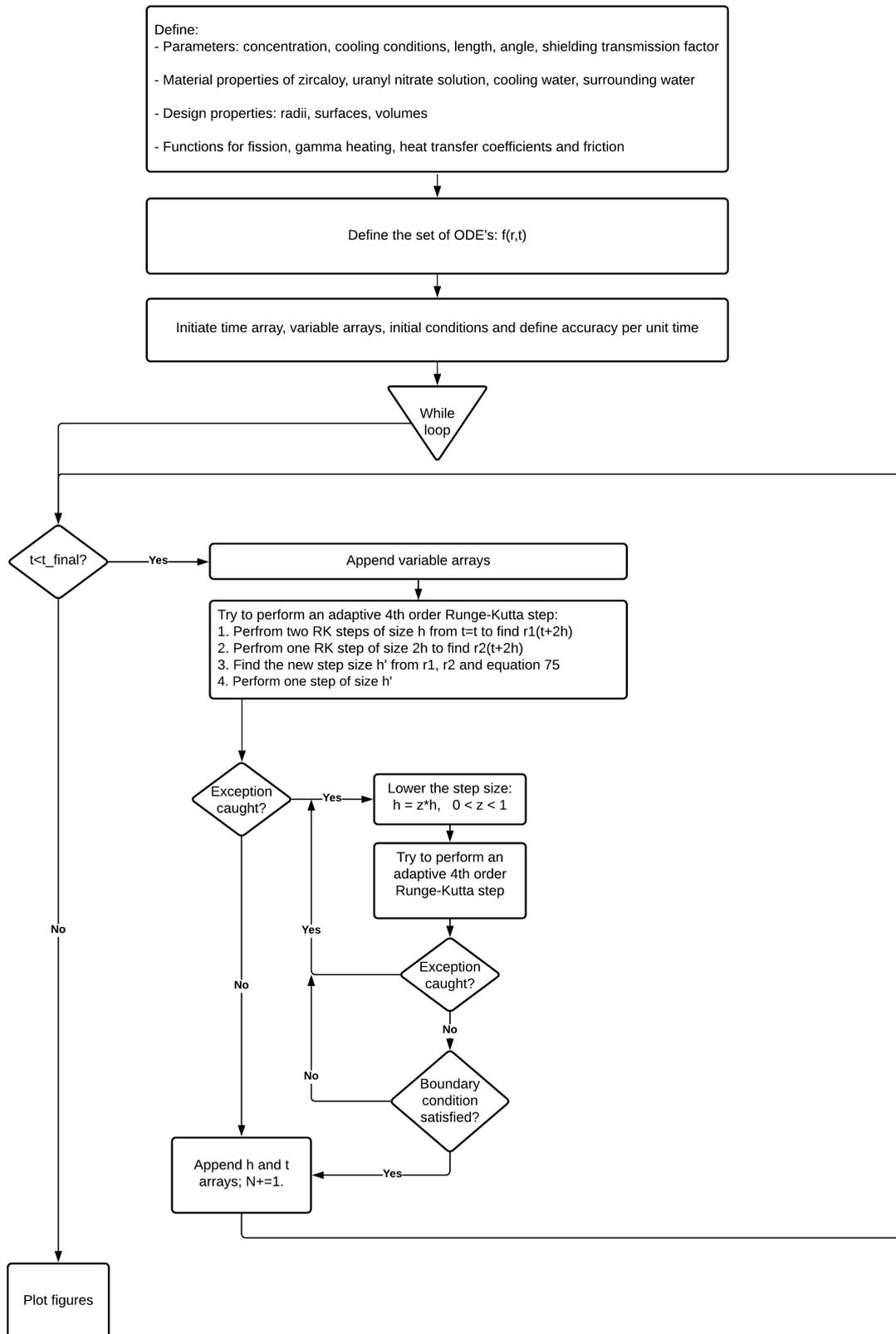


Figure 6-15: Flowchart of the final script. It consists of a number of definitions, a while loop and plotting figures.

7 Evaluating and benchmarking Dresen [2019]

The stationary simulation provided by Dresen [2019] contains a few mistakes that were not noticed before. The mistakes and their consequences are discussed below.

Moreover, the model of Dresen [2019] is used as a benchmark in two ways: firstly by deliberately applying the mistakes to the model developed in this research and using the model of Dresen [2019] as a benchmark. And secondly by improving the model of Dresen [2019] and using the corrected Matlab script as a benchmark.

Improvements with respect to Dresen [2019]

- For the thermal conductivity of the tube material, Dresen [2019] used the thermal conductivity of water by assigning both properties the name λ_w for 'wall' and 'water'. As the thermal conductivity of water is more than 10 times smaller than that of zircaloy, the heat resistance through the tube is simulated too largely.
- In the Reynolds number for node C , Dresen [2019] omits the dynamic viscosity, which decreases the Reynolds number with an order 10^{-4} . Therefore the heat transfer coefficient in node C decreases significantly.
- For the thermal diffusion coefficient of the uranyl nitrate solution, Dresen [2019] initially assigns the correct value to the variable 'a'. The value is however overwritten by the larger variable 'a' in the friction factor. This has no dominant consequences, however, as the thermal diffusion coefficient is only used for finding the heat transfer coefficient in a tube in the laminar regime.
- Calculating the shear stress in order to find the friction force for a fluid in a tube, Dresen [2019] uses $\tau_{w \rightarrow f} = -f_D \cdot \frac{1}{2} \rho \langle v \rangle^2$ instead of equation (26). This causes the friction force in the momentum balance being a factor 4 too large, which results in lower velocity values.⁸
- Dresen [2019] calculates the volume of the concentric tube node C as $V_C = 3\pi r^2 l$ instead of $V_C = \pi l (R^2 - (r + d)^2)$. Therefore the volume of node C is smaller in Dresen [2019]'s simulation, which alters cooling slightly.
- Dresen [2019] uses slightly wrong values for λ_w , ρ_w , a_w and $c_{p,w}$ by taking the values for 20°C instead of 10°C. This has no significant effect.

All in all, the decreased thermal conductivity of the tube material and the decreased Reynolds number of the cooling water cause the heat flow to the cooler being simulated too small. This lowered cooling power elevates temperatures in the loop, which explains why the unrealistically high cooling water flow $\phi_{m,C} = 0.7 \text{ kg/s}$ was needed to keep temperatures below 90°C. Moreover, the velocity in the loop is simulated too small.

Benchmark 1: Dresen [2019] without improvements

To benchmark the simulation, the above mistakes are deliberately applied to the transient model as developed in this thesis. Moreover, $T_{in,C}$ is used instead of $\frac{T_{in,C} + T_{out,C}}{2}$ in the heat exchanger.

⁸Moreover, equation (3.10) of Dresen [2019] seems wrong by a factor $\sqrt{2}$, compared to equation (3.9) of Dresen [2019]. This was however corrected in his Matlab simulations.

Furthermore, all extensions a-g as proposed in research question 1a-g are omitted from this benchmark, except that the model is transient instead of stationary. In other words, the model simulates the MPML according to the same assumptions as Dresen [2019]. It is therefore a simplified version of the model developed in this thesis:

- (a) Omitting the two vertical nodes from the simulation, simulating 3 nodes instead of five.
- (b) Simulating gamma radiation exposure in the fluid instead of in the tube wall.
- (c) Simulating no heat exchange with the surrounding water.
- (d) Setting the inclination angle of the near-horizontal nodes to $\alpha = 0^\circ$.
- (e) Simulating only positive flow directions.
- (f) Omitting gamma radiation exposure in the non-fission nodes.
- (g) Setting the length to $l=0.20$ m.

When comparing the results, the following temperature definitions of Dresen [2019] apply:

- T_A and T_B are named respectively T_2 and T_1 .
- The bulk temperature T_{Bulk} is defined as the average of temperatures T_2 and T_1 .
- The maximum bulk temperature is equal to $T_2 = T_A$.
- The maximum wall temperature is calculated from the bulk temperature as follows:

$$T_{max,wall} = P_{gamma}(r)/(\pi D l h_1) + T_{Bulk}.$$

As can be seen from the results in table 5, the steady state temperatures match with an error in temperature less than 2 degrees.

Cooling condition	Steady state values	Dresen	Current simulation	Percentual deviation
$\phi_{m,C} = 0.5$ (kg/s)	$T_{max,bulk}$ ($^\circ\text{C}$)	103.66	104.56	0.86%
	$T_{max,wall}$ ($^\circ\text{C}$)	105.82	107.09	1.19%
	v (m/s)	0.02206	0.02286	3.50%
$\phi_{m,C} = 0.1$ (kg/s)	$T_{max,bulk}$ ($^\circ\text{C}$)	106.39	107.53	1.06%
	$T_{max,wall}$ ($^\circ\text{C}$)	108.68	110.25	1.42%
	v (m/s)	0.02235	0.02327	3.95%
$\phi_{m,C} = 0.01$ (kg/s)	$T_{max,bulk}$ ($^\circ\text{C}$)	114.98	116.02	0.90%
	$T_{max,wall}$ ($^\circ\text{C}$)	117.75	119.22	1.23%
	v (m/s)	0.02339	0.02443	4.26%

Table 5: The current 'wrong' simulation compared to the simulation of Dresen [2019] for different cooling flows and $c = 300$ g/L, $T_{in,C} = 10^\circ\text{C}$. The current simulation deliberately contains all mistakes made by Dresen [2019].

Benchmark 2: Dresen [2019] containing improvements

A second benchmark consists of correcting the Matlab file produced by Dresen [2019] and comparing the results to the current model. Again, all extensions from research question 1a-g as are omitted from the benchmark.

As becomes clear from table 5, the steady state temperatures differ less than 0.5°C . It is concluded that the transient model without extensions correctly predicts the temperature and velocity profile since the same results are predicted from a stationary calculation.

A striking difference with the simulation containing the mistakes of Dresen [2019], is the significantly lower steady state temperatures: even for moderate cooling power the maximum temperature remains well below 90°C .

Cooling condition	Steady state values	Dresen	Current simulation	Percentual deviation
$\phi_{m,C} = 0.5$ (kg/s)	$T_{max,bulk}$ ($^{\circ}\text{C}$)	45.00	45.19	0.42%
	$T_{max,wall}$ ($^{\circ}\text{C}$)	46.47	46.55	0.17%
	v (m/s)	0.03088	0.03010	-2.59%
$\phi_{m,C} = 0.1$ (kg/s)	$T_{max,bulk}$ ($^{\circ}\text{C}$)	51.71	51.71	0.00%
	$T_{max,wall}$ ($^{\circ}\text{C}$)	53.44	53.26	-0.34%
	v (m/s)	0.03270	0.03213	-1.78%
$\phi_{m,C} = 0.01$ (kg/s)	$T_{max,bulk}$ ($^{\circ}\text{C}$)	60.47	60.85	0.62%
	$T_{max,wall}$ ($^{\circ}\text{C}$)	62.42	62.84	0.67%
	v (m/s)	0.03452	0.03488	1.03%

Table 6: Comparing the current simulation to the corrected simulation of Dresen [2019] for different cooling flows and $c = 300$ g/L, $T_{in,C} = 10^{\circ}\text{C}$. All mistakes made in the simulation of Dresen [2019] were corrected.

8 Results

The simplified transient simulation that is benchmarked in section 7, forms the basis for all extensions a-g. Following the research questions 1a-g in respective order, the model is gradually extended cumulatively. The result of every research question is compared to the previous research question, in order to see the individual effect of the extension of the research question. The complete model containing all extensions is used to answer research questions 2a-d.

In order to see the effect of the first extension, the results of the simplified model are shown in figure 8-16. The temperature profile is plotted against time, along with the velocity $\langle v(t) \rangle$ and the step size h . The parameters were set to $T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 0^\circ$, $l = 0.20 \text{ m}$ and shielding transmission factor = 0. The simulation was started from isothermal zero-velocity initial conditions.

From here on, the results are shown by plotting all the variables (the temperatures and the velocity) against time. Moreover, the step size h is plotted against time. The figure captions will consist of four parts:

- A description of the simulation
- An outline of the adaptive Runge-Kutta method: the number of steps, the execution time and the accuracy per unit time δ (equation (64))
- An outline of the parameter values
- An outline of the steady state variable values and the maximum node temperature throughout time $T_{max,t}$

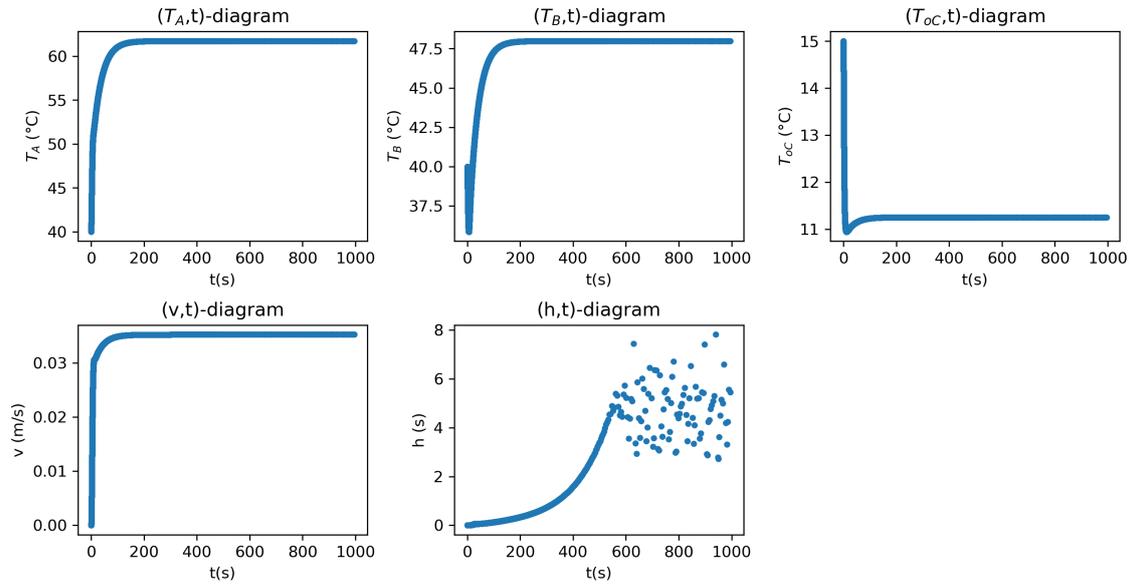


Figure 8-16: Results from the simplified transient simulation as benchmarked in section 7. The temperatures, velocity and step size h are plotted against time.

Adaptive RK	5534 steps, 5.51 seconds, $\delta_1 = 10^{-13} \text{ } ^\circ\text{C/s}$, $\delta_2 = 10^{-13} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 0^\circ$, $l = 0.20 \text{ m}$, $stf = 0$.
SS results	$T_A = 61.73^\circ\text{C}$, $T_B = 47.99^\circ\text{C}$, $T_{node} = 54.86^\circ\text{C}$, $T_{out,C} = 11.25^\circ\text{C}$, $v = 0.03526 \text{ m/s}$
$T_{max,t}$	61.73°C

8.1 Optimizing the simulation

8.1.1 Question 1a

Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspect affect the feasibility predictions for the ⁹⁹Molybdenum Producing Mini Loop? Inclusion of the two vertical nodes in the simulation.

Simulating the fluid and heat fluxes in the vertical parts of the loop gives five temperature nodes instead of three. The results are displayed in figure 8-17. The results are compared to the model without extension 1a (figure 8-16).

Although steady state temperatures differ only by a negligible amount of 0.04 °C, the flow development until $t = 100$ s differs markedly: the gravitational terms in the momentum balance causing natural convection now depend on the densities of the vertical nodes D and E . As becomes clear from the v , T_D and T_E diagrams below, it takes a period of 25 seconds until a temperature difference between these nodes has developed. During these 25 seconds the velocity remains zero and heat accumulates in node A to a temperature of 107°C. This causes the temperature in node A to soar and in node B to plummet, until enough heat has entered node D to establish natural convection.

After 25 seconds, node A has absorbed a lot of heat and node B has cooled down to a temperature close to $T_{in,C} = 10$ °C. The flow slowly starts to develop, which leads to an extreme temperature difference between nodes D and E because the warm fluid flows into node D and the cold fluid into E . This becomes clear from the T_D maximum and T_E minimum at $t = 35$ s. These extreme temperatures generate a peak in the velocity after which a steady state balance is accomplished. Clearly, the temperature peak rises above 90°C.

To conclude, inclusion of the two vertical nodes in the simulation unveils a temperature peak $T_{max} > 90$ °C in the first hundred seconds when activating the setup from zero-velocity, isothermal initial conditions. Should this simulation accurately represent reality, a uranyl nitrate concentration of 310 g/L is not feasible.

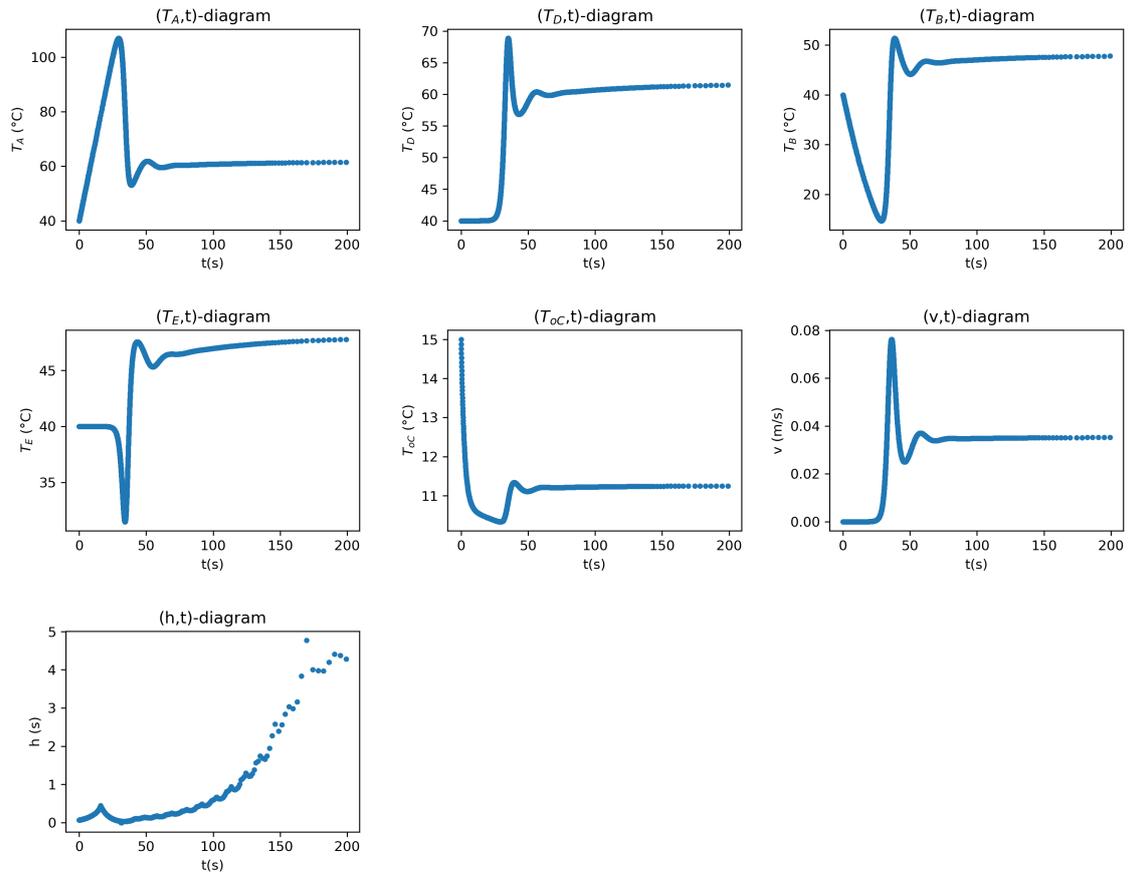


Figure 8-17: The results for research question 1a: The first 200 seconds are shown for the simulation extended to 5 nodes. When continuing the simulation to 1000 s, the following steady results are obtained:

Adaptive RK	1071 steps, 1.75 seconds, $\delta_1 = 10^{-8} \text{C/s}$, $\delta_2 = 10^{-8} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 0^\circ$, $l = 0.20 \text{ m}$, $stf = 0$.
SS results	$T_A = 61.69^\circ\text{C}$, $T_D = 61.69^\circ\text{C}$, $T_B = 47.99^\circ\text{C}$, $T_E = 47.99^\circ\text{C}$, $T_{out,C} = 11.26^\circ\text{C}$, $v = 0.03536 \text{ m/s}$
$T_{max,t}$	106.97°C

8.1.2 Question 1b

Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspect affect the feasibility predictions for the $^{99}\text{Molybdenum}$ Producing Mini Loop? Gamma radiation exposure in the tube wall instead of in the fluid.

The results are shown in figure 8-18 and are compared to the model without this extension (figure 8-17). Removing the gamma radiation exposure P_{gamma} from the internal energy balance in node A and adding it to the internal energy balance over the tube wall surrounding node A , significantly lowers the energy peak from $T_{max} = 107^\circ\text{C}$ to $T_{max} = 88.6^\circ\text{C}$ ⁹. Steady state temperatures, however, remain exactly the same. Clearly, adding P_{gamma} to the energy balance inside the node as done by Dresen [2019] is an accurate approximation for steady state values, since it predicts the same results. It does lead to an overestimated peak, however.

The lowered peak is manifested by a delay in gamma heating of node A : as becomes clear from the (T_A, t) and $(T_{w,A}, t)$ -diagram in figure 8-18, both the tube wall and the fluid are heated. The heat is distributed over the node and the wall. Therefore less heat accumulates in node A during the first 25 seconds.

To conclude, feasibility predictions are improved by simulating gamma radiation exposure in the tube wall instead of in the node, because the temperature peak is lowered. A concentration of 310 g/L is now anticipated to be feasible.

⁹This becomes clear from the (T_A, t) -diagrams in figures 8-17 and 8-18, as well as from the T_A and $T_{max,t}$ values in the figure descriptions.

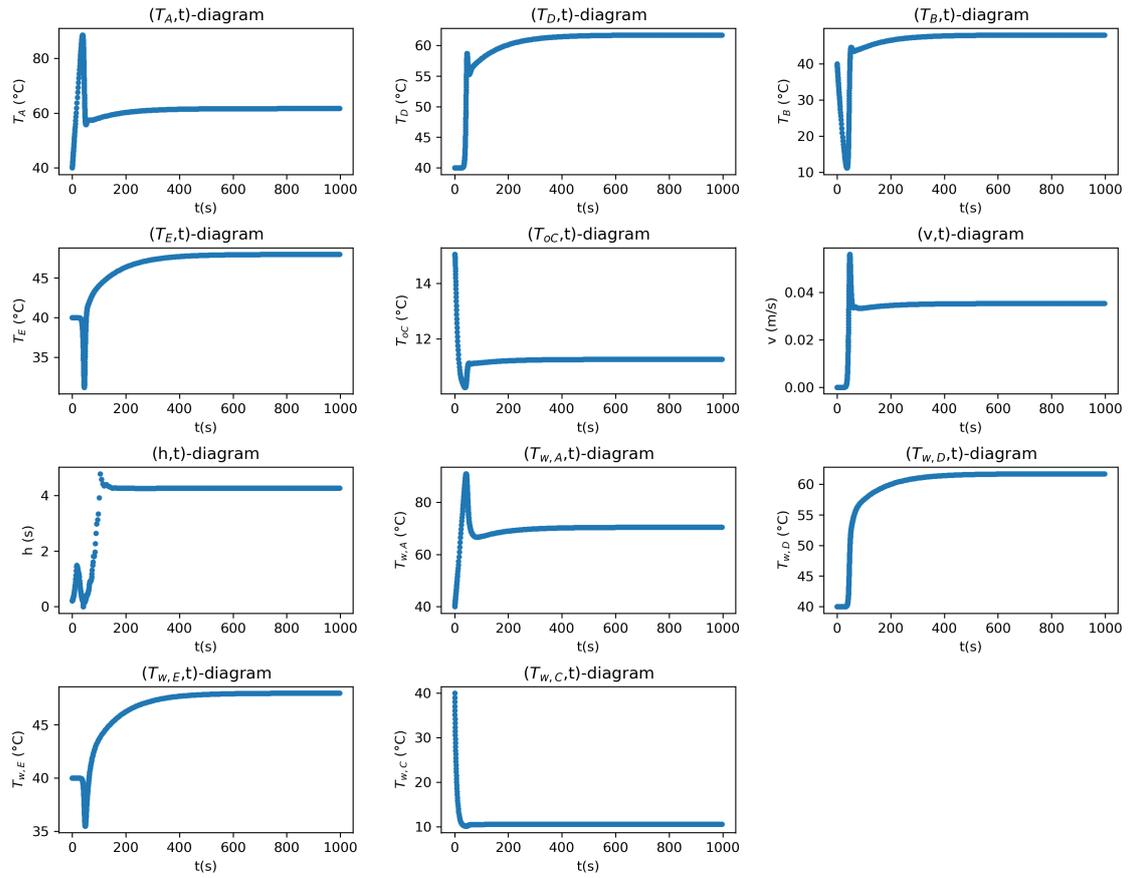


Figure 8-18: The results for research question 1b: Simulating gamma radiation exposure in the tube wall of node A instead of in the node itself.

Adaptive RK	416 steps, 1.23 seconds, $\delta_1 = 10^{-6} \text{ }^\circ\text{C/s}$, $\delta_2 = 10^{-6} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 0^\circ$, $l = 0.20 \text{ m}$, $stf = 0$.
SS results	$T_A = 61.69^\circ\text{C}$, $T_D = 61.69^\circ\text{C}$, $T_B = 47.99^\circ\text{C}$, $T_E = 47.99^\circ\text{C}$, $T_{out,C} = 11.26^\circ\text{C}$, $v = 0.03536 \text{ m/s}$, $T_{w,A} = 70.45^\circ\text{C}$, $T_{w,D} = 61.69^\circ\text{C}$, $T_{w,E} = 47.99^\circ\text{C}$, $T_{w,C} = 10.63^\circ\text{C}$.
$T_{max,t}$	88.56°C

8.1.3 Question 1c

Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspect affect the feasibility predictions for the $^{99}\text{Molybdenum}$ Producing Mini Loop? Heat exchange with the surrounding water.

The results are shown in figure 8-19 and are compared below to the model without this extension (figure 8-18). Simulating heat exchange with the surrounding water significantly improves feasibility predictions: the temperature peak in node A is reduced by 21°C to only $T_{max,t} = 67.2^\circ\text{C}$, and the maximum steady state temperature is diminished by 14°C to $T_{A,SS} = 47.4^\circ\text{C}$.¹⁰

Since the coolant wall and node temperature are below the surrounding temperature $T_s = 40^\circ\text{C}$, these temperatures have increased by respectively 18 and 2.3°C .¹¹ The coolant itself remains at low temperature due to its constantly renewed inlet flow, while the wall is constantly heated by the surroundings.

All other temperatures were above 40°C and therefore lost heat to the surrounding water because of a negative temperature difference.

To conclude, feasibility predictions are further improved by simulating heat exchange with the DLDR water, as the temperature peak and steady state temperatures significantly decrease.

¹⁰The temperature peak and maximum steady state temperature become clear from the (T_A, t) -diagrams in figures 8-18 and 8-19, as well as from the T_A and $T_{max,t}$ values in the figure descriptions.

¹¹This becomes clear from the $(T_{out,C}, t)$ -diagrams in figures 8-18 and 8-19.

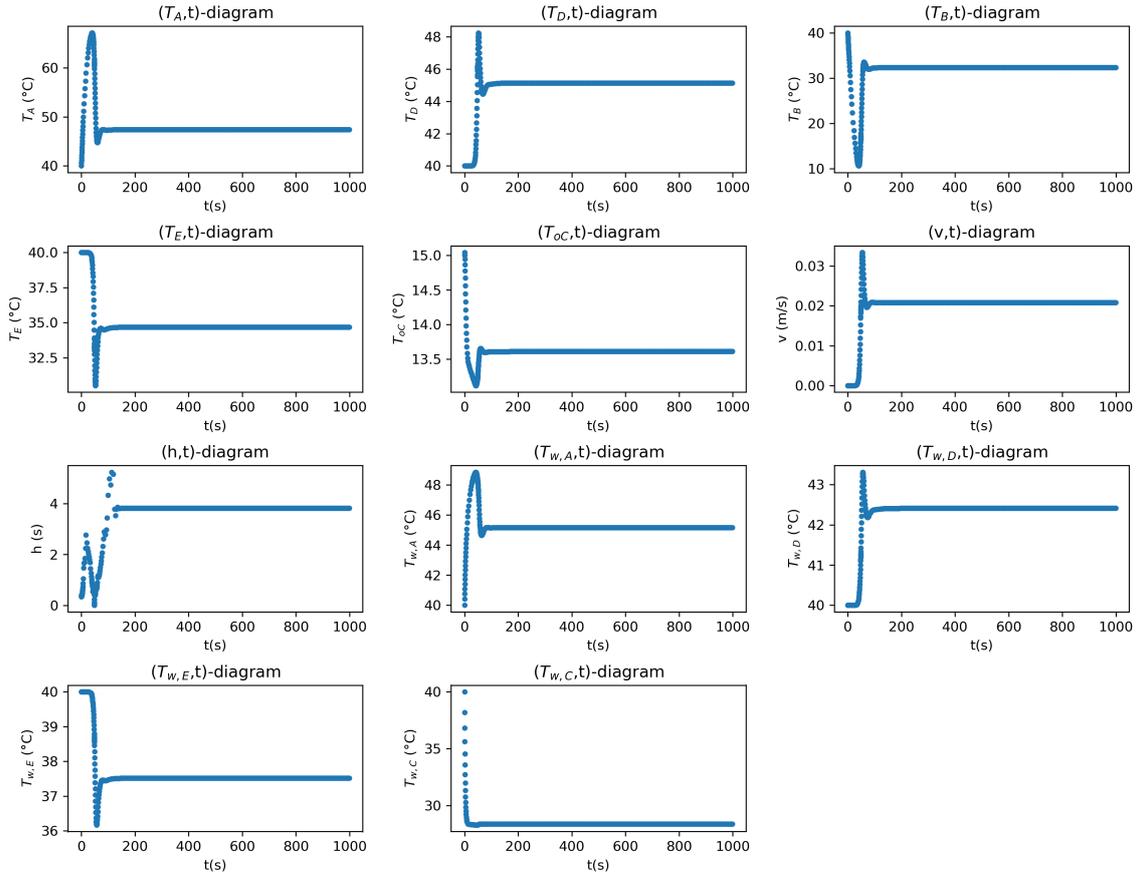


Figure 8-19: The results for research question 1c: Simulating heat exchange with the surrounding DLDR water through natural convection.

Adaptive RK	336 steps, 1.1 seconds, $\delta_1 = 10^{-5} \text{C/s}$, $\delta_2 = 10^{-5} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 0^\circ$, $l = 0.20 \text{ m}$, $stf = 0$.
SS results	$T_A = 47.41^\circ\text{C}$, $T_D = 45.15^\circ\text{C}$, $T_B = 32.34^\circ\text{C}$, $T_E = 34.69^\circ\text{C}$, $T_{out,C} = 13.61^\circ\text{C}$, $v = 0.02084 \text{ m/s}$, $T_{w,A} = 45.18^\circ\text{C}$, $T_{w,D} = 42.41^\circ\text{C}$, $T_{w,E} = 37.52^\circ\text{C}$, $T_{w,C} = 28.39^\circ\text{C}$
$T_{max,t}$	67.18°C

To benchmark the correlation by Xian et al. [2015] for natural convection around vertical cylinders, it is replaced with that of Le Fevre (equation (33)) in figure 8-20. As becomes clear from figures 8-19 and 8-20, the temperature peak is exactly the same. The maximum difference in steady state temperatures is found in the wall temperatures of the vertical nodes. This temperature difference is only 0.05°C , however. The accuracy of the Xian et al. [2015] correlation is thus confirmed, since equation (33) is a widely used and recommended correlation.

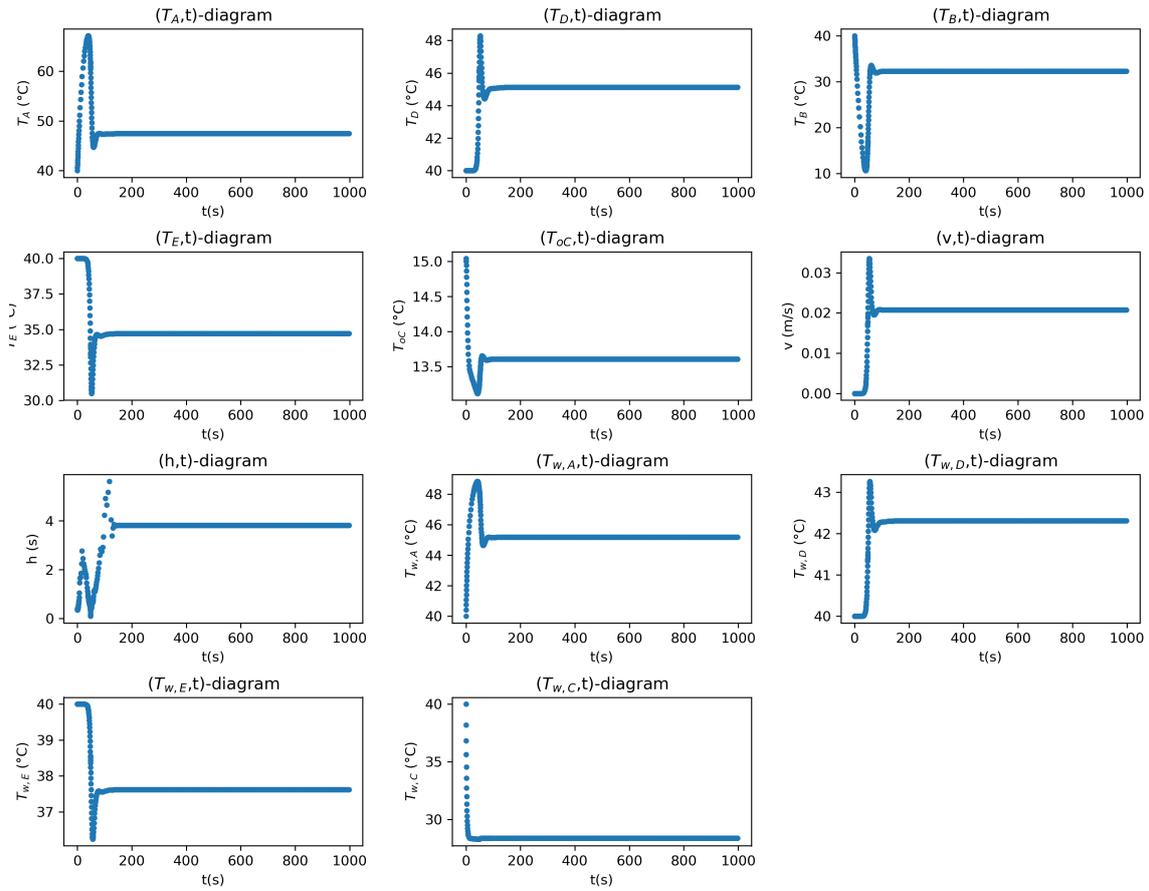


Figure 8-20: Applying correlation (33) instead of (32) for natural convection around vertical cylinders.

Adaptive RK	329 steps, 1.03 seconds, $\delta_1 = 10^{-5} \text{ }^\circ\text{C/s}$, $\delta_2 = 10^{-5} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 0^\circ$, $l = 0.20 \text{ m}$, $stf = 0$.
SS results	$T_A = 47.46^\circ\text{C}$, $T_D = 45.12^\circ\text{C}$, $T_B = 32.29^\circ\text{C}$, $T_E = 34.71^\circ\text{C}$, $T_{out,C} = 13.61^\circ\text{C}$, $v = 0.02075 \text{ m/s}$, $T_{w,A} = 45.19^\circ\text{C}$, $T_{w,D} = 42.31^\circ\text{C}$, $T_{w,E} = 37.62^\circ\text{C}$, $T_{w,C} = 28.39^\circ\text{C}$
$T_{max,t}$	67.18°C

8.1.4 Question 1d

Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspect affect the feasibility predictions for the ^{99}Mo Producing Mini Loop? The inclination angle of the near-horizontal nodes.

As depicted in figure 8-22, simulating the near-horizontal nodes at a 2° angle lowers the temperature peak in node A by 12°C to only $T_{max,t} = 56.5^\circ\text{C}$, while the maximum steady state temperature is only reduced by 0.2°C .¹² The lowered peak is due to a quick flow development, as depicted in the velocity diagram of figure figure 8-22. The flow develops immediately from $t=0$, which is 25 seconds earlier than the horizontal case as clearly visible in figure 8-17. This is because a temperature difference between the near-horizontal nodes now also causes a buoyancy force in the momentum balance. As heat accumulates in node A and heat is extracted from node B from the start, natural convection is induced from the beginning.

Steady state, however, this extra gravitational term is negligible compared to the gravitational terms in the vertical parts. Long-term temperatures are therefore not significantly affected. As depicted in figure 8-21, increasing the angle further to 45° lowers the temperature peak $T_{max,t}$ by 10°C but has no significant effect on the steady state temperatures. Even though larger angles seem advantageous, a larger angle allows for smaller side lengths of the loop (equation (2)). To maximize the loop volume and thereby the production rate (equation (15)), a 2° angle suffices.

To conclude, feasibility predictions are further improved by simulating a 2° inclination angle as the temperature peak is lowered.

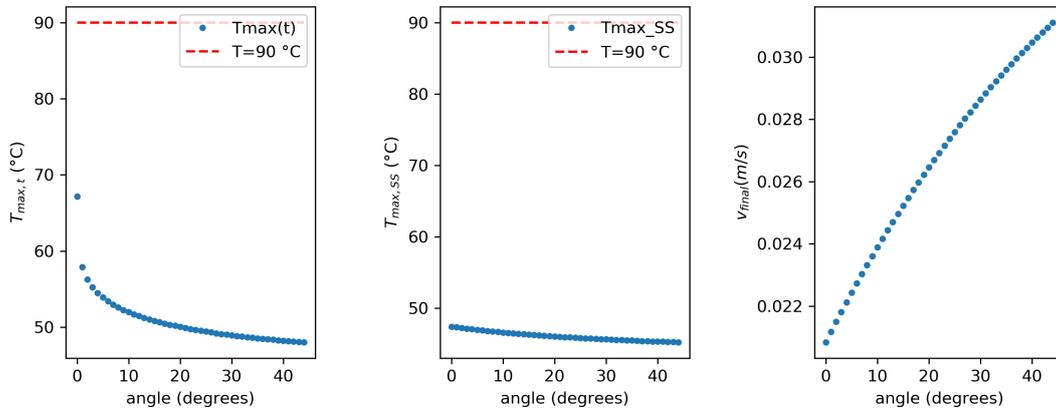


Figure 8-21: The results for research question 1d: The maximum temperature throughout time, the maximum steady state temperature and the final velocity are plotted against varying angles from 0° to 44° .

¹²The temperature peak and maximum steady state temperature become clear from the (T_A, t) -diagrams in figures 8-19 and 8-23, as well as from the T_A and $T_{max,t}$ values in the figure descriptions.

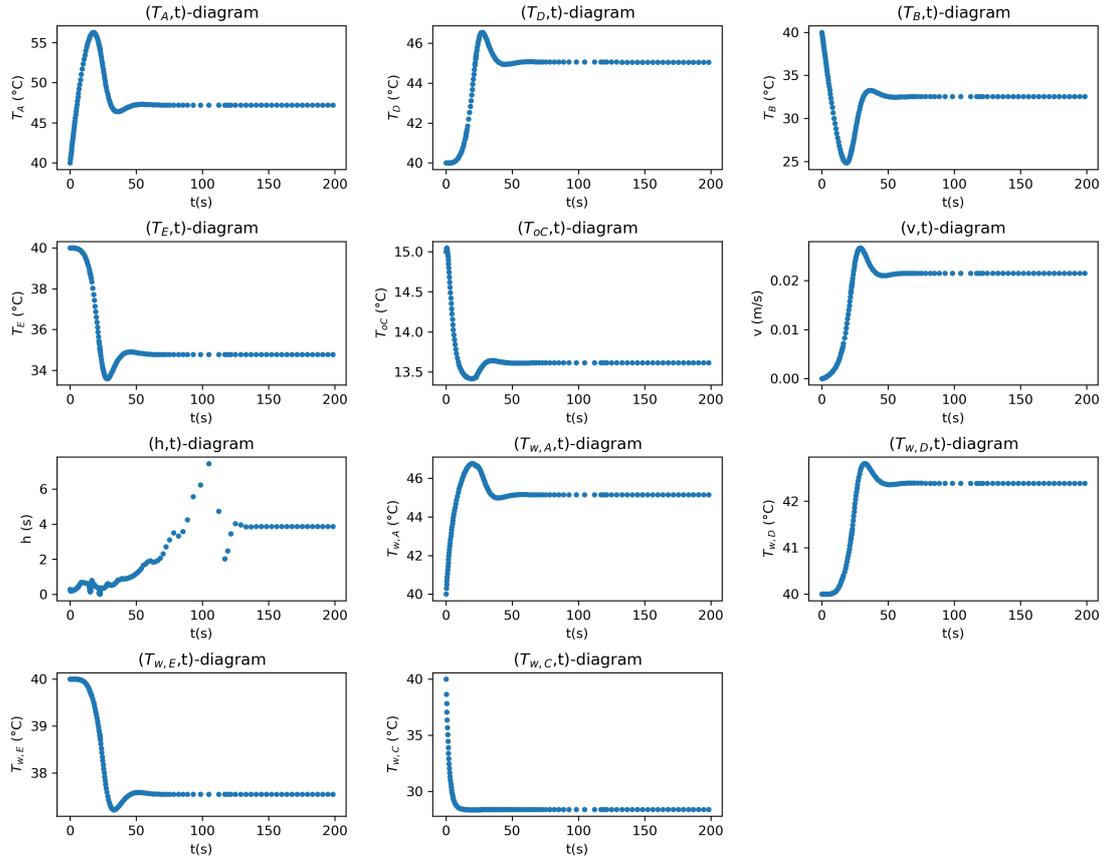


Figure 8-22: The results for research question 1d: Simulating the near-horizontal nodes at a 2° angle. The first 200 seconds are shown to illustrate the effect on the temperature peak. When continuing the simulation to 1000 s, the following steady results are obtained:

Adaptive RK	363 steps, 1.73 seconds, $\delta_1 = 10^{-6} \text{C/s}$, $\delta_2 = 10^{-6} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 2^\circ$, $l = 0.20 \text{ m}$, $stf = 0$.
SS results	$T_A = 47.22^\circ\text{C}$, $T_D = 45.05^\circ\text{C}$, $T_B = 32.53^\circ\text{C}$, $T_E = 34.78^\circ\text{C}$, $T_{out,C} = 13.61^\circ\text{C}$, $v = 0.02149 \text{ m/s}$, $T_{w,A} = 45.14^\circ\text{C}$, $T_{w,D} = 42.39^\circ\text{C}$, $T_{w,E} = 37.55^\circ\text{C}$, $T_{w,C} = 28.39^\circ\text{C}$
$T_{max,t}$	56.28°C

8.1.5 Question 1e

Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspect affect the feasibility predictions for the ⁹⁹Molybdenum Producing Mini Loop? Positive and negative flow directions.

Extending the simulation to both positive and negative velocities is useful to model the behaviour close to zero-velocity. To test the behaviour of the simulation, the $\alpha = 2^\circ$ situation is simulated using the new script that allows negative velocities, as depicted in figure 8-23. The result is exactly equal to figure 8-22 and is therefore accurate for positive velocities.

Subsequently, it is interesting how the flow develops for an angle of $\alpha = -2^\circ$. The results are displayed in figure 8-24. As becomes clear from comparing figures 8-23 and 8-24, the velocity sign has switched and the peak and steady state temperatures differ at most 0.2°C .¹³ The small difference in temperature can be explained by the fact that the heat exchanger becomes co-current for negative velocities, as opposed to the counter-current heat exchanger that has a stronger cooling power. Another difference is that the temperature profiles in nodes *D* and *E* have switched, which is because the two nodes have exchanged position before and after the cooler.

It is concluded that the model is accurate for both positive and negative velocities. Moreover, the above proves that an inclination angle of $\alpha = \pm 2^\circ$ forces the flow into either positive or negative direction.

¹³The temperature peak and maximum steady state temperature become clear from the (T_A, t) -diagrams in figures 8-23 and 8-24, as well as from the T_A and $T_{max,t}$ values in the figure descriptions.

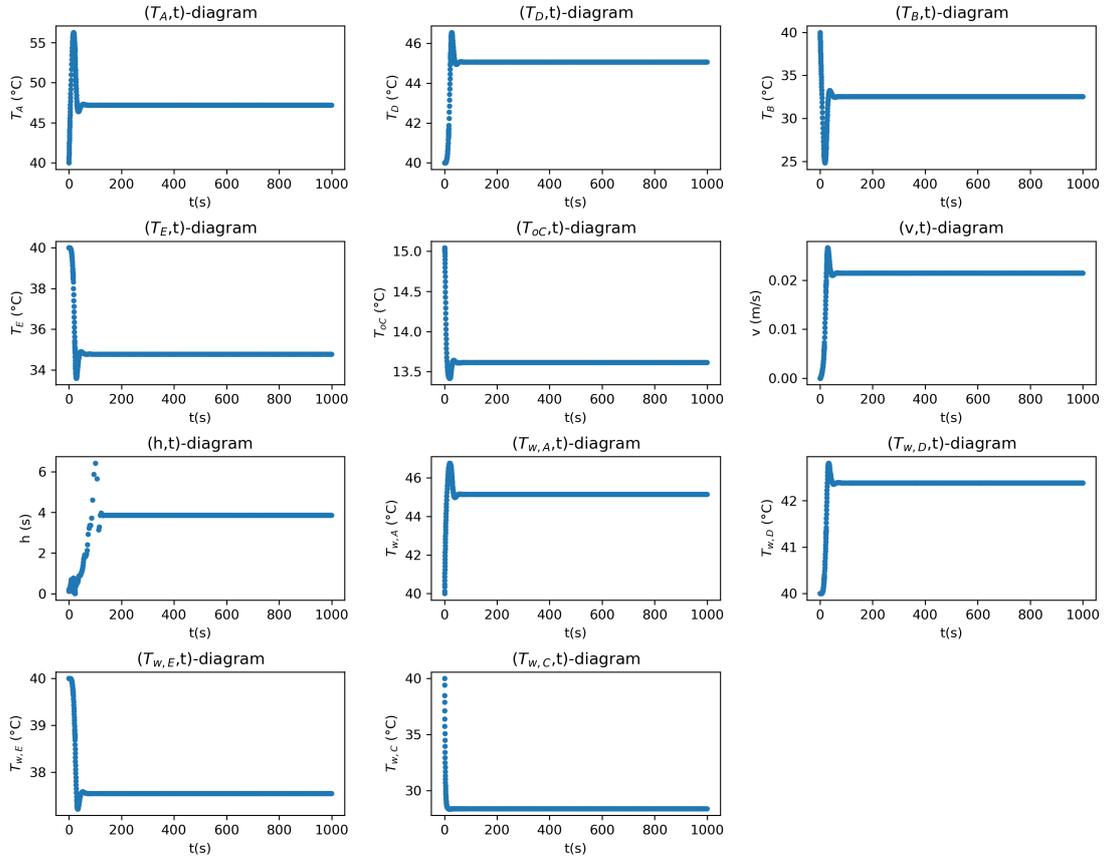


Figure 8-23: The results for research question 1e: Using the script allowing for negative velocities, simulating the $\alpha = 2^\circ$ situation.

Adaptive RK	363 steps, 1.45 seconds, $\delta_1 = 10^{-6} \text{C/s}$, $\delta_2 = 10^{-6} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 2^\circ$, $l = 0.20 \text{ m}$, $stf = 0$.
SS results	$T_A = 47.22^\circ\text{C}$, $T_D = 45.05^\circ\text{C}$, $T_B = 32.53^\circ\text{C}$, $T_E = 34.78^\circ\text{C}$, $T_{out,C} = 13.61^\circ\text{C}$, $v = 0.02149 \text{ m/s}$, $T_{w,A} = 45.14^\circ\text{C}$, $T_{w,D} = 42.39^\circ\text{C}$, $T_{w,E} = 37.55^\circ\text{C}$, $T_{w,C} = 28.39^\circ\text{C}$
$T_{max,t}$	56.28°C

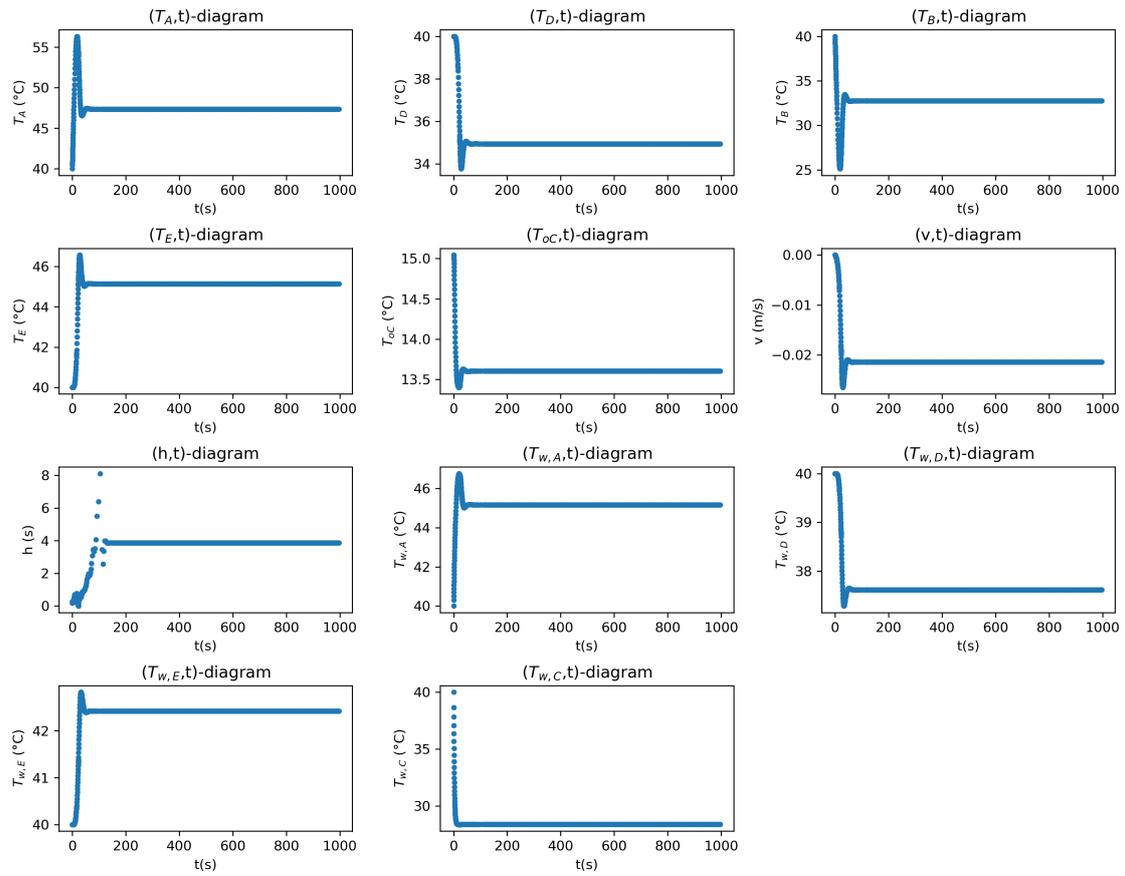


Figure 8-24: The results for research question 1e: Using the script allowing for negative velocities, simulating the $\alpha = -2^\circ$ situation.

Adaptive RK	456 steps, 1.75 seconds, $\delta_1 = 10^{-7} \text{ } ^\circ\text{C/s}$, $\delta_2 = 10^{-7} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, angle = -2° , $l = 0.20 \text{ m}$, $stf = 1$.
SS results	$T_A = 47.36^\circ\text{C}$, $T_D = 34.94^\circ\text{C}$, $T_B = 32.76^\circ\text{C}$, $T_E = 45.14^\circ\text{C}$, $T_{out,C} = 13.6^\circ\text{C}$, $v = -0.02140 \text{ m/s}$, $T_{w,A} = 45.17^\circ\text{C}$, $T_{w,D} = 37.61^\circ\text{C}$, $T_{w,E} = 42.42^\circ\text{C}$, $T_{w,C} = 28.39^\circ\text{C}$
$T_{max,t}$	56.31°C

8.1.6 Question 1f

Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspect affect the feasibility predictions for the $^{99}\text{Molybdenum}$ Producing Mini Loop? Gamma radiation exposure in the non-fission nodes.

In figure 8-25, the maximum temperature in all nodes throughout time is plotted against varying shielding transmission factors for the non-fission nodes C, D and E . The maximum node steady state temperature is also plotted against varying shielding transmission factors. A transmission factor of 0 means that shielding blocks all gamma-radiation in nodes C, D, E , in contrast to 1 when nodes C, D and E are exposed to as much gamma radiation as node A .

Clearly, shielding of gamma radiation is not necessary to maintain safe temperatures $T < 90^\circ\text{C}$: a maximum temperature difference of 3.3°C can be obtained from shielding. Cooling to the surrounding water is fundamental to this result. As gamma radiation exposure rises, the majority of the added heat leaks away to the surroundings.

The displayed profile in figure 8-25 can be understood from the two linear relations: the heat production in the relevant tube walls is a linear function of the shielding transmission factor; the heat flow from the wall to the node is linearly related to the wall temperature (equation (28)). As the wall temperatures $T_{w,D}(t)$, $T_{w,C}(t)$ and $T_{w,E}(t)$ increase linearly with the transmission factor, so does the heat flow from the wall to the node.

To conclude, simulating gamma radiation in the non-fission nodes worsens the feasibility prediction. The effect is small, however, due to a strong cooling power towards the surroundings and the heat exchanger.

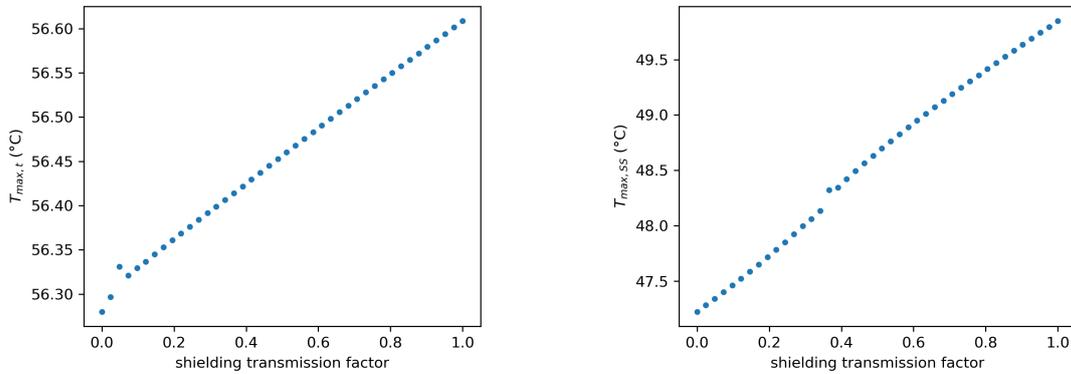


Figure 8-25: The results for research question 1f: The maximum temperature in all nodes throughout time and the maximum steady state temperature plotted against varying shielding transmission factors for nodes C, D, E . The other parameters were $T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01$ kg/s, $c = 310$ g/L, $\alpha = 2^\circ$, $l = 0.20$ m.

8.1.7 Question 1g

Extending the simplified stationary simulation to a transient model, to what extent does incorporation of the following aspect affect the feasibility predictions for the $^{99}\text{Molybdenum}$ Producing Mini Loop? Adjusted size for placement inside the DLDR?

As becomes clear from the $(T_{max,t}, t)$ -diagram of figure 8-26, varying the tube length does not bring temperatures above 65°C : the feasibility at $c = 310 \text{ g/L}$ is not hampered. Since it follows from equation (2) that $l = 0.11\text{m}$ is the largest shape that fits inside the DLDR for an angle $\alpha = 2^\circ$, this is chosen as the length for the design.

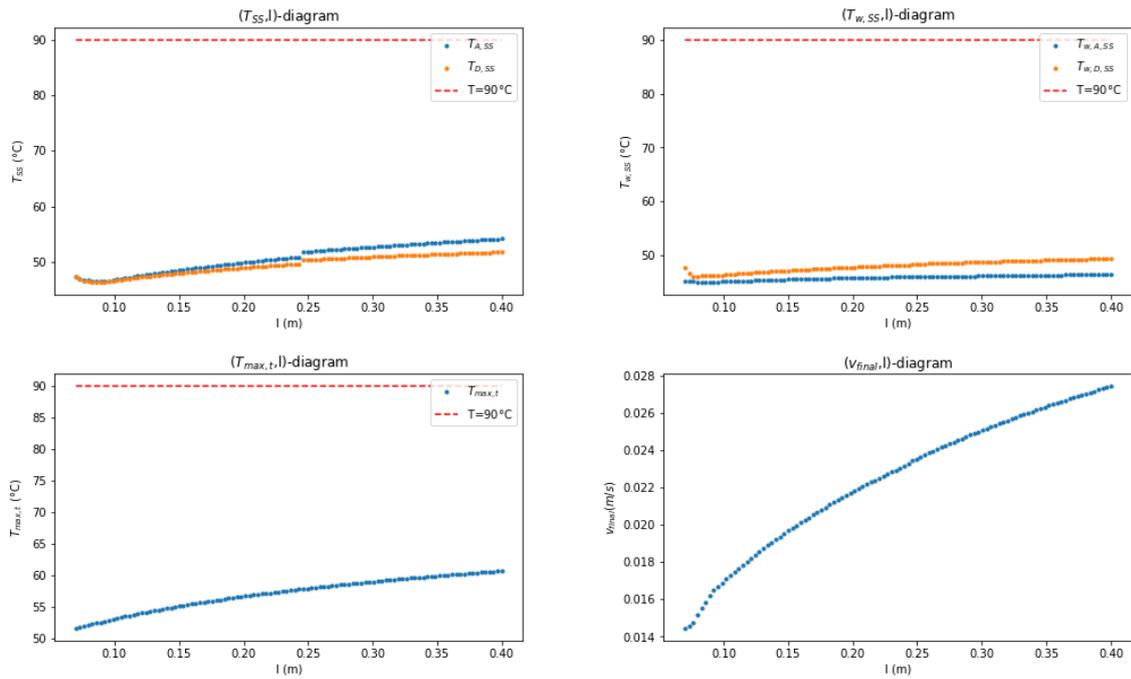


Figure 8-26: The results for research question 1g: Plotting steady state node temperatures, steady state wall temperatures, maximum node temperature throughout time and velocity against varying length for $l > 0.07\text{m}$. The other parameters were $T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 2^\circ$, shielding transmission factor = 1.

8.1.8 Optimized model

Simulating 5 nodes, incorporating gamma radiation and heat exchange with the surrounding water in the internal energy balance over the tube wall, and allowing for positive and negative velocities, have further ameliorated feasibility predictions compared to the simplified model (figure 8-16).

To ensure a positive velocity while still maintaining a large volume fitting in the DLDR¹⁴, the angle is set to $\alpha = 2^\circ$ and the length to $l=0.11$ m. Gamma radiation shielding is not necessary and is even challenging to implement, as outlined in section 3.4. All shielding transmission factors are therefore set to 1.

The results of the final model are shown in figure 8-27. The steady state values of this simulation are used as initial conditions for emergency simulations in research questions 2a-d.

¹⁴An increasingly large angle restricts the MPML length l , as becomes clear from equation (2).

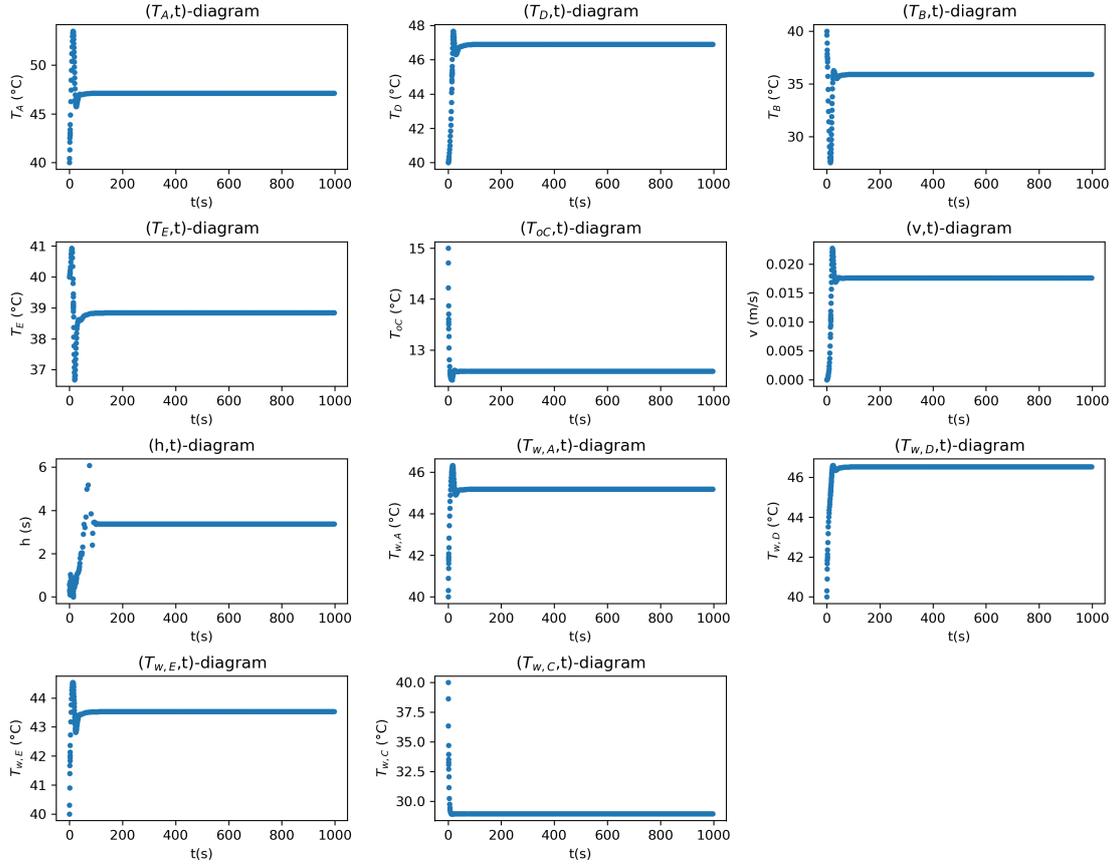


Figure 8-27: Plotting temperature and velocity development for the optimized model including all extensions from research question 1.

Adaptive RK	358 steps, 1.26 seconds, $\delta_1 = 10^{-5} \text{ }^\circ\text{C/s}$, $\delta_2 = 10^{-5} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\alpha = 2^\circ$, $l = 0.11 \text{ m}$, $stf = 1$.
SS results	$T_A = 47.13^\circ\text{C}$, $T_D = 46.9^\circ\text{C}$, $T_B = 35.91^\circ\text{C}$, $T_E = 38.84^\circ\text{C}$, $T_{out,C} = 12.59^\circ\text{C}$, $v = 0.01758 \text{ m/s}$, $T_{w,A} = 45.18^\circ\text{C}$, $T_{w,D} = 46.54^\circ\text{C}$, $T_{w,E} = 43.53^\circ\text{C}$, $T_{w,C} = 28.96^\circ\text{C}$.
$T_{max,t}$	53.48°C

8.2 Safety

8.2.1 Question 2a

Applying the extended simulation, how does the following emergency case affect the safety of the MPML? A defect cooling pump causes the cooling to solely rely on heat flow to the surrounding water.

To simulate the effects of a defect cooling pump, the steady state results of the optimized model 8-27 are used as initial conditions, while deactivating the cooling pump as outlined in section 1.5.

Cooling solely relies on heat flow to the surrounding water now. As depicted in for instance the (T_A, t) - and $(T_{out,C}, t)$ -diagrams of figure 8-28, temperatures increase from steady state conditions, especially in node C . Due to heat exchange with the surrounding at $T_s = 40^\circ\text{C}$, however, the wall of node C remains below 46°C and node C does not become warmer than 83.1°C . Since the temperature difference between nodes D and E diminishes to 4°C , natural convection becomes weak and the velocity drops.

To conclude, the heat exchanger does surprisingly not seem to be necessary for safe steady state operation since temperatures remain safely below 84°C . The MPML remains safe.

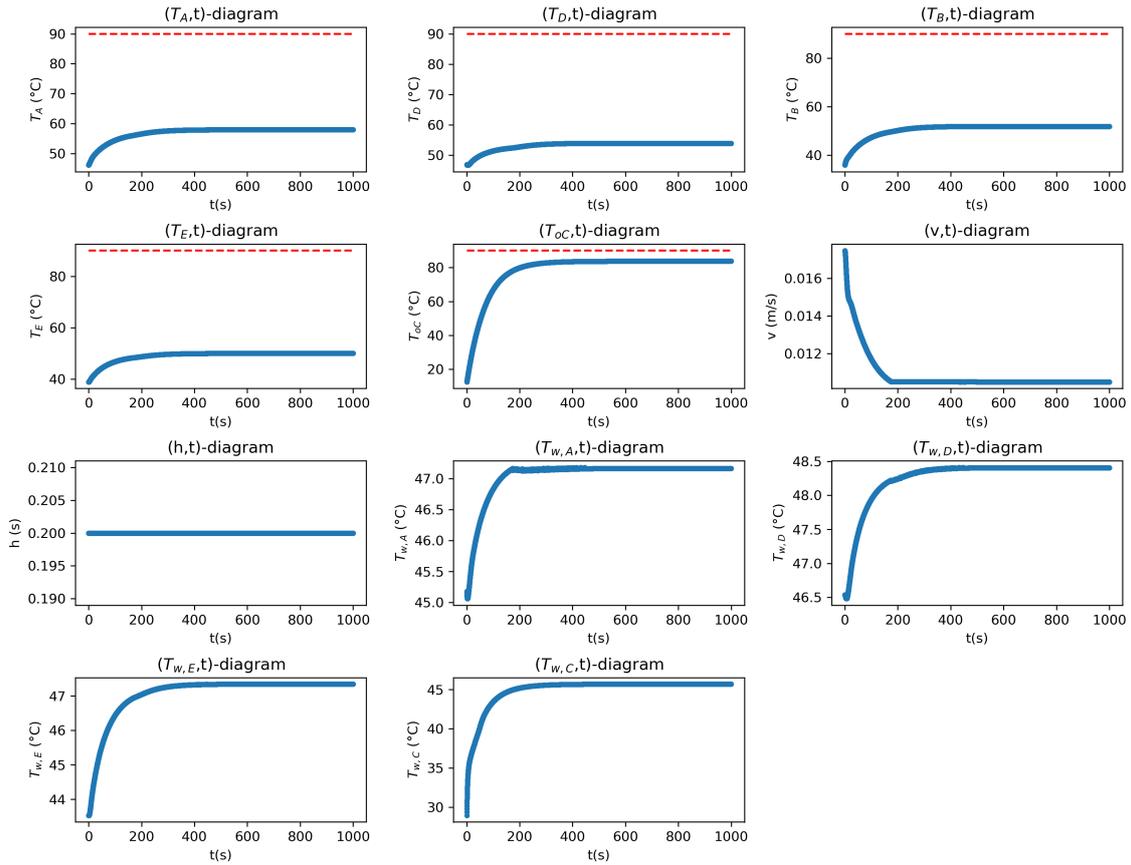


Figure 8-28: The results for research question 2a: Simulating temperature and velocity development starting from steady state conditions and defect cooling pump at $t=0$.

Adaptive RK	$N = 5000$ steps, $h=0.2$ s, 3.57 seconds,
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 10^{-20}$ kg/s, $c = 310$ g/L, angle = 2° , $l = 0.11$ m, $stf = 1$
SS results	$T_A = 57.97^\circ\text{C}$, $T_D = 53.93^\circ\text{C}$, $T_B = 51.91^\circ\text{C}$, $T_E = 49.98^\circ\text{C}$, $T_{out,C} = 83.76^\circ\text{C}$, $v = 0.0105$ m/s, $T_{w,A} = 47.16^\circ\text{C}$, $T_{w,D} = 48.41^\circ\text{C}$, $T_{w,E} = 47.34^\circ\text{C}$, $T_{w,C} = 45.68^\circ\text{C}$
$T_{max,t}$	83.76°C

8.2.2 Question 2b

Applying the extended simulation, how does the following emergency case affect the safety of the MPML? An empty DLDR causes the cooling to solely rely on the heat exchanger.

As becomes clear from the results in figure 8-29, temperatures remain safe in the emergency case that an empty DLDR causes the cooling to solely rely on active cooling by the heat exchanger.

At $t=0$ the DLDR is considered empty. Firstly the three wall temperatures $T_{w,A}$, $T_{w,D}$ and $T_{w,E}$ start rising since the walls keep being heated by gamma radiation while heat can no longer escape to the surrounding water. Simultaneously, the corresponding node temperatures T_A , T_D and T_E absorb heat from the walls. A stationary situation is reached after 400 seconds.

Since the wall of node C was heated by the surrounding water for $t < 0$, the temperatures $T_{w,C}$ and $T_{out,C}$ start declining now that this source of heat has vanished. Shortly after, the node C temperature $T_{out,C}$ slightly increases again as the heat convection coming from node D grows.

Even though all the node temperatures remain safely below 90°C , the wall temperature node D becomes $T_{w,D} = 87.71^\circ\text{C}$. This poses no danger, however, since it is only the fluids that are required to remain below 90°C .

To conclude, heat exchange with the surrounding DLDR water is not needed for safe steady state operation since temperatures remain safely below 81°C . The MPML remains safe.

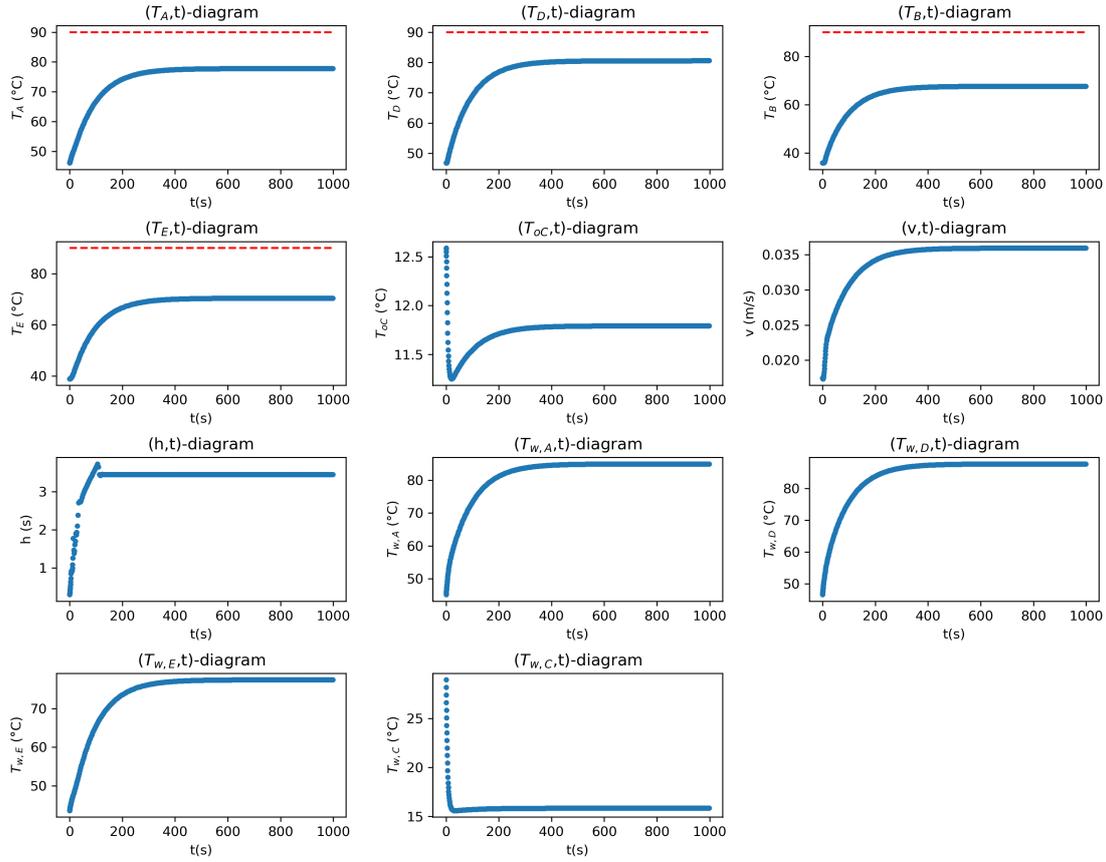


Figure 8-29: The results for research question 2b: Simulating the emergency case in which an empty DLDR prevents heat exchange with the surroundings.

Adaptive RK	314 steps, 1.31 seconds, $\delta_1 = 10^{-5} \text{ }^\circ\text{C/s}$, $\delta_2 = 10^{-5} \text{ m/s}^2$
Parameters	$T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01 \text{ kg/s}$, $c = 310 \text{ g/L}$, $\text{angle} = 2^\circ$, $l = 0.11 \text{ m}$, $stf = 1$
SS results	$T_A = 77.81^\circ\text{C}$, $T_D = 80.58^\circ\text{C}$, $T_B = 67.63^\circ\text{C}$, $T_E = 70.4^\circ\text{C}$, $T_{out,C} = 11.79^\circ\text{C}$, $v = 0.03596 \text{ m/s}$, $T_{w,A} = 84.95^\circ\text{C}$, $T_{w,D} = 87.71^\circ\text{C}$, $T_{w,E} = 77.53^\circ\text{C}$, $T_{w,C} = 15.83^\circ\text{C}$
$T_{max,t}$	80.58°C

8.2.3 Question 2c

Applying the extended simulation, how does the following emergency case affect the safety of the MPML? A defect cooling pump and an empty DLDR cause accumulation of heat in the loop.

In this section the very rare emergency scenario is simulated in which the DLDR is suddenly empty and the cooling pump becomes defect simultaneously, starting from steady state flow (figure 8-27). The results are shown in figure 8-30. As outlined in section 6 the adaptive method is only beneficial if the system reaches a steady state equilibrium. That is not the case here. Therefore non-adaptive Runge-Kutta is applied.

The loop is no longer cooled, leading to accumulation of heat. As depicted in 8-30 the node that firstly rises to the critical temperature 90°C , is the uranyl nitrate solution in node *A*. Starting from the instantaneous event at $t = 0$, a period of hundred seconds remains to block the neutron and gamma flux or extract the uranyl nitrate solution out of the MPML.

To conclude, a defect cooling pump and an empty DLDR require a shut down within 100 seconds. If this is feasible, then the safety of the setup remains intact.

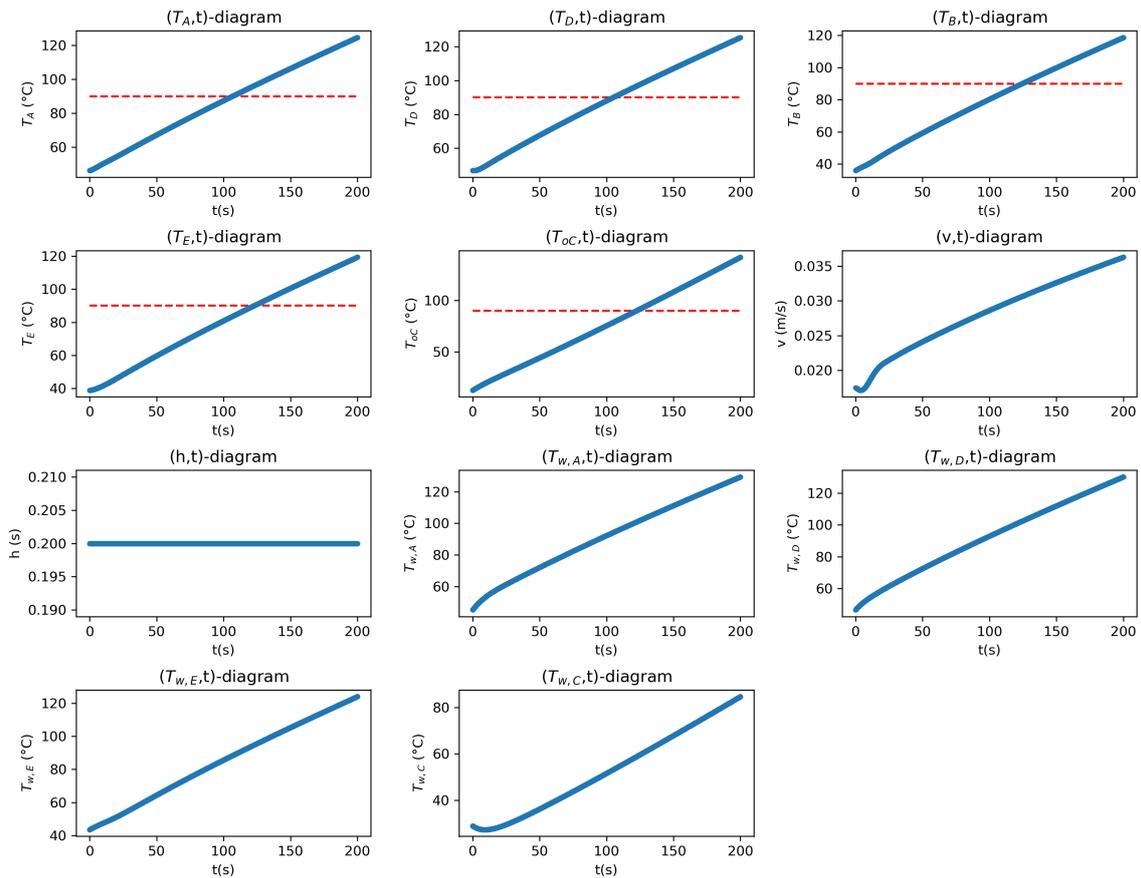


Figure 8-30: The results for research question 2c: An empty DLDR and defect cooling pump cause heat accumulation in the loop. The parameters were $T_{in,C} = 10^{\circ}\text{C}$, $\phi_{m,C} = 10^{-20}$ kg/s, $c = 310$ g/L, angle = 2° , $l = 0.11$ m, shielding transmission factor = 1. Non-adaptive Runge Kutta method is applied since the result does not become stationary.

8.2.4 Question 2d

Applying the extended simulation, how does the following emergency case affect the safety of the MPML? An increased neutron flux causes heat production by fission to rise.

In the emergency case that the reactor power increases, for instance by cold water flowing through the reactor core, an increased neutron and gamma flux stimulate the heat production by fission and gamma radiation. From 8-31 it is clear that temperatures remain safely below $T < 90^\circ\text{C}$, even for the extremely unlikely case that the neutron and gamma flux would become six times larger. The latter situation is physically not viable in the Hoger Onderwijs Reactor, which makes an increased reactor power a harmless situation.

To conclude, an increased reactor power is no emergency case and does not affect the safety of the MPML.

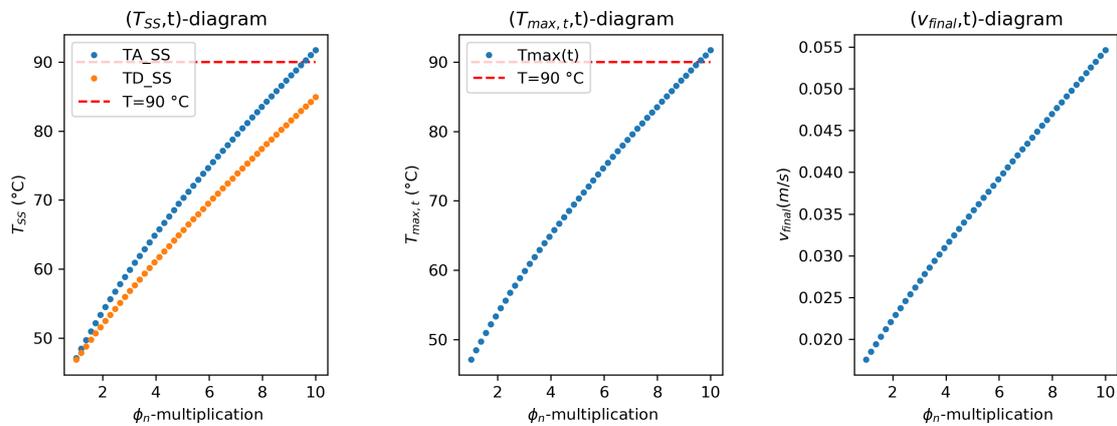


Figure 8-31: The results for research question 2d: Plotting steady state temperatures, maximum node temperatures throughout time and velocity against the multiplication factor for increased gamma and neutron flux. A factor of 2 denotes doubled heat production by fission and gamma radiation. The parameters were $T_{in,C} = 10^\circ\text{C}$, $\phi_{m,C} = 0.01$ kg/s, $c = 310$ g/L, angle = 2° , $l = 0.11$ m, shielding transmission factor = 1.

9 Conclusions

This research focused on correcting and extending the stationary simulation of Dresen [2019] in order to investigate feasibility and safety aspects.

Before extending the simulation, the stationary model of Dresen [2019] was corrected. This significantly improved feasibility predictions: allowing for a higher concentration 310 g/L instead of 236 g/L, while requiring a significantly lower cooling power. This result was confirmed by both the corrected stationary Matlab simulation of Dresen [2019] and the transient Python simulation of this research.

Extensions

Extending the benchmark to a simulation including the vertical nodes unveiled the appearance of a temperature peak in the first 100 seconds. Starting from the benchmark and adding an extension step by step, the effect of every extra extension on the steady state and peak temperature is shown in table 7. It is concluded that extension a) is necessary to discover the temperature peak and extensions b), c) and d) are necessary for lowering it. For steady state temperatures, the only significant change was extension c). Extensions f) and g) only had a small effect on the feasibility and e) confirmed that the inclination angle forces the flow into either positive or negative direction.

Research question	Effect on peak temperature	Effect on maximum SS temperature
a) Inclusion of the two vertical nodes in the simulation	+45.3°C	0.0°C
b) Gamma radiation exposure in the tube wall	-18.4°C	0.0°C
c) Heat exchange with surrounding water	-21.4°C	-14.5°C
d) The inclination angle $\alpha = 2^\circ$ of the near-horizontal nodes	-10.9°C	-0.2°C
e) Positive and negative flow directions	0.0°C	$\pm 0.2^\circ\text{C}$
f) Gamma radiation exposure in the non-fission nodes	+0.3°C	+3.3°C
g) Shortening the length from $l=0.20$ m to $l=0.11$ m	-3.1°C	-2.8°C

Table 7: The effect of different extensions on the peak and maximum steady state temperature.

Important to note, is that the extensions were added cumulatively, meaning that all extensions a-d were for instance also incorporated in the simulation of question e. The results of table 7 are therefore not absolute: without cooling towards the surrounding water, for instance, it follows from question 2b) that removing the gamma shielding does lead to significantly higher temperatures.

Feasibility

The optimized model fits inside the DLDR with a length $l = 0.11$ m and an inclination angle of 2° . At $c = 310$ g/L, $\phi_{m,C} = 0.01$ kg/s and $T_{in,C} = 10^\circ$ the maximum steady state temperature and peak temperature are respectively 47.1°C and 53.5°C . Since 310 g/L is the maximum concentration possible for effective $^{99}\text{Molybdenum}$ extraction and $T < 90^\circ\text{C}$ is the feasibility requirement, the corrections and extensions have led to promising results.

Safety

Applying the optimized model to emergency cases showed that safe temperatures can be maintained even in the extreme conditions of a defect cooling pump, an empty DLDR or an increased reactor power. If the first two occur simultaneously, a period of 100 seconds is available to safely shut down the system.

10 Shortcomings and recommendations

Shortcomings

1. Heat exchange with the surrounding water is more complex than assumed here for four reasons. Firstly, natural convection is simulated as though the loop is placed in a water pool without the DLDR. In practise the natural convection flow within the DLDR could be more complex, for instance by upward flow in the middle and downward flow along the sides of the DLDR. And the heat flow out of the DLDR could cause an upward natural convection around the DLDR. Secondly, figure 3-11 shows that the DLDR is surrounded by two boxes and the reactor reactor core, which could decrease heat transfer to the pool. Thirdly, the DLDR fluid flows slowly which increases cooling of the loop. Lastly, heating caused by neutron blocking is not taken into account. As the shielding should not confine natural convection in the DLDR, the shielding is applied close to the loop (figure 3-11). This could reduce cooling towards the surroundings.
2. The corners of the loop have been omitted from the simulation. Including them in the internal energy balance increases accuracy but does not lead to significant changes since the heat production and heat exchange is small in these sections due to their small volume. Including them in the momentum balance slightly increases friction, thereby lowering the velocity. A slightly decreased velocity does not lead to significant temperature differences, however, as depicted in figure 8-26.
3. Solving the wall temperature of node B would enable simulation of gamma exposure in this node, as would it more accurately predict heat flow in the heat exchanger.
4. Natural convection around vertical cylinders is not taken into account for $Gr < 10^8$. Instead, conduction is assumed in this range. If natural convection were included, however, cooling towards the surroundings would slightly increase for small lengths $l < 7\text{mm}$ (figure 8-26).

To what extent do these shortcomings affect the results? Shortcoming 2) brings no significant temperature changes and 3) should have no significant effect either, as gamma exposure to the other nodes had no significant effect already. Shortcoming 4) makes the simulation a conservative estimate of the temperatures for low Grashof numbers. If natural convection is included in this region, temperatures would further decrease.

Shortcoming 1) could in the worst case scenario mean that the cooling towards the surroundings is heavily overestimated. From research question 2b), however, it follows that heat exchange with the DLDR water is not necessary for safe temperatures so the feasibility of $c = 310 \text{ g/L}$ remains true even for this worst case. Emergency cases would be worsened, although not to an existential threat: from question 2c) it follows that a short period is available to shut down the reactor if the cooling pump should suddenly stop. A safe handling of emergency cases should therefore still be possible.

Recommendations

It is recommended to further optimize the current simulation and investigate various other aspects of the loop. Optimizations include:

1. Improving upon the most important shortcomings by more accurately simulating the heat exchange with the surrounding water and by solving the wall temperature of node B throughout time.

2. Dividing the loop into more nodes. This would in the most simple form consist of adding a temperature variable and an internal energy balance for every extra node. Doing so would increase the accuracy of the assumption $T_{node} = T_{out,node}$.
3. While the velocity is assumed to be the same for every node in this research, the fluid flow is in practise expected to accelerate and decelerate in the vertical nodes. It is therefore recommended to add a velocity variable and a transient momentum balance for every node.
4. Maximizing the volume: the length of the vertical and horizontal tubes was shortened from 0.2 to 0.11 m in this research. Increasing the horizontal tube length until the loop volume is 0.5L would maximize $^{99}\text{Molybdenum}$ production.

In addition, examination of the following aspects is recommended:

1. The flowing DLDR water could function as a heat exchanger besides filtering contamination. This heat exchange setup as designed by Pendse [2018] is displayed in figure 10-32.
2. Investigating the possibility for neutron exposure on a vertical node gives perspective for a Multi-MPML setup. If possible, it would enable placement of multiple loops in a row.
3. Investigating the effect of placement farther from the reactor core additionally gives perspective for a larger setup. A lower neutron flux is expected to reduce the reaction rate.
4. Examining the pressure is necessary for safety, because a uranyl nitrate solution exposed to radiation might trigger the formation of gasses such as NO_x , NO , H_2 and O_2 . Besides, additional margin could be reached by elevating the pressure and thereby increasing the boiling point of the aqueous solution.
5. Elgin [2014] derived that the uranyl nitrate concentration is limited by the extraction of $^{99}\text{Molybdenum}$: the maximum possible value is 310 g/L. Since temperatures remain below 55°C for this concentration, extraction of molybdenum is considered the limiting factor for the uranyl nitrate concentration. Extraction of $^{99}\text{Molybdenum}$ therefore needs further exploration to investigate whether even higher concentrations are feasible.
6. As outlined by Pendse [2018], the neutron flux considered is based on Huisman [2016] and does not consider any interaction of neutrons with water. The current MPML design suggests roughly 90 mm of water between the reactor core and the loop. It is therefore recommended to incorporate this interaction in order to more accurately predict the neutron flux and the reaction rate.

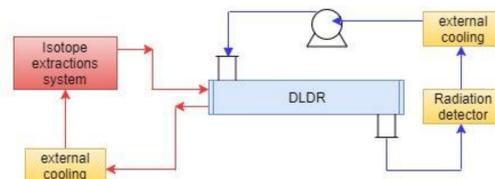


Figure 10-32: The cooling system as designed by Pendse [2018].

References

- Luuk R. Dresen. The mpml: 99-molybdenum producing mini loop, driven by natural circulation. Technical report, Delft University of Technology, 2019.
- Mushtaq Ahmad. Molybdenum-99/technetium-99m management: race against time. *Annals of nuclear medicine*, 25(9):677–679, 2011.
- Supply and demand of medical radioisotopes, 2021. URL <https://www.oecd-nea.org/>.
- K. Elgin. A study of the feasibility of 99mo production inside the tu delft hoger onderwijs reactor. Technical report, Delft University of Technology, 2014.
- Jurriaan A. R. Huisman. Heat transfer of the mo-99 research loop: Optimising heat transfer of the closed reactor loop aimed at the efficient production of molybdenum-99. Technical report, Delft University of Technology, 2016.
- C.C. Pothoven. Recirculation of the mo-99 research loop, predicting the long-term behaviour of the the loop content. Technical report, Delft University of Technology, 2016.
- S. Pendse. Cooling system design for 99mo research loop. Technical report, Delft University of Technology, 2018.
- LPBM Janssen and MMCG Warmoeskerken. *Transport phenomena data companion*. E. Arnold, 1987.
- WE Grant, WJ Darch, ST Bowden, and WJ Jones. The surface tension and viscosity of solutions of uranyl salts. *The Journal of Physical Chemistry*, 52(7):1227–1236, 1948.
- Not all the rings in space go around saturn, 2014. URL <https://www.atimetals.com/>.
- Harry Van den Akker and Robert Mudde. *Transport Phenomena: The Art of Balancing*. Delft Academic Press/VSSD, 2014.
- Dr. ir. Rohde. *Nuclear Reactor Thermal Hydraulics (AP3641)*. 2014.
- William Murray Deen. Analysis of transport phenomena. 1998.
- Vasilis Bellos, Ioannis Nalbantis, and George Tsakiris. Friction modeling of flood flow simulations. *Journal of Hydraulic Engineering*, 144, 2018.
- Amir Akbari, Erfan Mohammadian, Seyed Ali Alavi Fazel, Mehdi Shanbedi, Mahtab Bahreini, Milad Heidari, Parham Babakhani Dehkordi, and Siti Nurliyana Che Mohamed Hussein. Natural convection from the outside surface of an inclined cylinder in pure liquids at low flux. *ACS omega*, 4(4):7038–7046, April 2019. ISSN 2470-1343. doi: 10.1021/acsomega.9b00176.
- Mohamed Ali and Shereef Sadek. *Free Convection Heat Transfer from Different Objects*. 06 2018. ISBN 978-1-78923-264-6. doi: 10.5772/intechopen.75427.
- R.M. Fand, E.W. Morris, and M. Lum. Natural convection heat transfer from horizontal cylinders to air, water and silicone oils for rayleigh numbers between 3×10^2 and 2×10^7 . *International Journal of Heat and Mass Transfer*, 20(11):1173 – 1184, 1977. ISSN 0017-9310. doi: [https://doi.org/10.1016/0017-9310\(77\)90126-0](https://doi.org/10.1016/0017-9310(77)90126-0). URL <http://www.sciencedirect.com/science/article/pii/0017931077901260>.

Stuart W Churchill and Humbert HS Chu. Correlating equations for laminar and turbulent free convection from a horizontal cylinder. *International journal of heat and mass transfer*, 18(9): 1049–1053, 1975.

Lin Xian, Guangming Jiang, and Hongxing Yu. Natural convective heat transfer from a heated slender vertical tube in a cylindrical tank. *NURETH-16, Chicago, IL, August 30 - September 4*, 2015.

Czeslaw O. Popiel. Free convection heat transfer from vertical slender cylinders: A review. *Heat Transfer Engineering*, 29(6):521–536, 2008.

Mark Newman. *Computational physics*. CreateSpace Independent Publ., 2013.

H.W. Hoogstraten. *Collegedictaat toegepaste wiskunde II numerieke methoden*. 1985.

Appendices

A Conduction around a cylinder

Heat transfer between the loop and the surroundings is described by natural convection. An exception occurs when the temperature difference between the loop and the surroundings is small. If the temperature difference does not cause a buoyancy force large enough to conquer the viscous force, then natural convection does not occur. Heat transfer between a cylinder and a stationary surrounding is then described by conduction. The Nusselt number for this situation can be found by recognizing and combining the following:

- Heat flow in radial direction is assumed to be constant, since the radius is small compared to the length $l \gg r$:

$$A \cdot \phi_q'' = K, \quad (69)$$

in which $\phi_q'' [J/(m^2s)]$ is the heat flux and $A = 2\pi rl$ is the surface area between cylinder and the water. K is a constant.

- Heat flux by conduction is described by Fourier's law:

$$\phi_q'' = -\lambda \frac{\partial T}{\partial r}, \quad (70)$$

where $\lambda [W/(mK)]$ is the thermal conductivity.

- At the outer radius of the cylinder, the temperature is equal to the wall temperature of the cylinder. Far from the cylinder (for instance at a 10 meters distance), the temperature is assumed to be equal to the water temperature of the pool.

$$\begin{aligned} r = R_{out} & : T = T_w \\ r = 10m & : T = 40^\circ C \end{aligned}$$

- Heat flux is in general given by:

$$\phi_q'' = h\Delta T. \quad (71)$$

Substituting equation (70) into (69), integrating and solving for the given boundary conditions and comparing to equation (19) substituted into (71), gives $Nu = 0.26$.

B Consequences for the momentum balance

In this section, the effect of the assumption $\rho_{ref,j} = \rho_{ref,avg} = (\rho_{ref,A} + \rho_{ref,D} + \rho_{ref,B} + \rho_{ref,E})/4$ is investigated.¹⁵

It is first noted that $\rho_{ref,A} = \rho_{ref,D} = \rho_{ref,B} = \rho_{ref,E}$ is not only an assumption but is exactly true if the simulation starts from isothermal initial conditions and the Boussinesq approximation is applied.

The effect of the approximation is calculated for the case that the simulation is initiated from non-isothermal conditions.

¹⁵'j' denotes the nodes A,D,B and E and $\rho_{ref,j} = \rho_{ref,A} = \rho_{ref,D} = \rho_{ref,B} = \rho_{ref,E}$.

With a coolant temperature $T_{in,C} = 30^\circ C$ and the maximum temperature $T_{max} = 90^\circ C$ as assumed by Dresen [2019], steady state temperatures in nodes A, D, B and E could in a worst case be respectively equal to 90, 70, 50 and $30^\circ C$. This scenario is given the name 'I' and is defined by the following reference densities [kg/m^3] (Janssen and Warmoeskerken [1987]):

$$\rho_{0,I}(\vec{r}) = \begin{pmatrix} 965.34 \\ 977.93 \\ 988.07 \\ 995.68 \end{pmatrix}, \quad \vec{r} = \begin{pmatrix} r_A \\ r_D \\ r_B \\ r_E \end{pmatrix}.$$

The average reference density is now equal to $\rho_{ref,avg} = 981.76kg/m^3$. In order to investigate the effect of the assumption on situation I, this average reference density $\rho_{ref,avg} = 981.76kg/m^3$ is used to approximate two other hypothetical situations named 'II' and 'III' that are defined by their reference densities [kg/m^3]:

$$\rho_{0,II}(\vec{r}) = \begin{pmatrix} 965.34 \\ 965.34 \\ 965.34 \\ 965.34 \end{pmatrix}, \quad \rho_{0,III}(\vec{r}) = \begin{pmatrix} 995.68 \\ 995.68 \\ 995.68 \\ 995.68 \end{pmatrix}.$$

Neglecting the inclination angle, the total momentum balance over the loop (equation (60)) can be rewritten as follows:

$$\frac{d}{dt}(\langle v \rangle(t)) = \frac{g(\rho_E(t) - \rho_D(t))}{4 \cdot \rho_{ref}} - \frac{f \cdot \frac{1}{2} \langle v(t) \rangle^2 \cdot 2\pi r l}{4 \cdot \pi r^2 l}, \quad (B.1)$$

in which only the first term is significantly affected by the choice of reference density. It is found that the error in this term that is caused by using $\rho_{ref,j} = \rho_{ref,avg} = 981.76kg/m^3$ to approximate situations II and III are respectively 1.66% and 1.42% compared to using the true densities.

$\rho_{ref,j} = \rho_{ref,avg} = 981.76kg/m^3$ is a better approximation for case I than for II and III, since it is the average of the densities of case I. When approximating case I using $\rho_{ref,j} = \rho_{ref,avg} = 981.76kg/m^3$, the error in the first term caused by this approximation is therefore even smaller than 1.66%.

The simulation of this research resembles case I but has smaller temperature and density differences. Therefore the error in the first term of equation (B.1) becomes even smaller for this research, and the assumption is accurate to apply.

C Python code

```
1  ###Applying Adaptive Runge-Kutta to the MPML
2  #Allowing for negative velocities
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import time
6  start_time = time.time()
7
8
9
10 ###Parameters
11 c = 310 #[g/L]
12 T_inC = 10
13 phi_mC = 0.01
14 l = 0.11
15 shielding_tf = 1
16 angle = 2
17
18
19
20 ###Material Properties
21 #Properties of uranyl nitrate solution
22 rho0 = 1330.6
23 cp = 2905.5
24 Lambda = 0.665
25 a = Lambda/(rho0*cp)
26 beta = 0.000523
27 M = 394.04
28 c_mol = c/M
29 def mu(T,c_mol):
30     mu0 = mu_w(T)
31     A=-0.1687
32     B=0.7904
33     return mu0*(1+A*np.sqrt(c_mol)+B*c_mol)
34
35 def Pr(T,c_mol):
36     return mu(T,c_mol)*cp/Lambda
37
38 def rho(T):
39     rho_ref0 = 1330.6
40     T_ref0 = 60
41     return rho_ref0 - rho_ref0*beta*(T-T_ref0)
42
43 #Properties of zircaloy
44 rho_t = 6.55e3
45 eps = 1.5e-6 #effective (not relative) roughness
46 Lambda_t = 21.5
47 cp_t = 285
48
49 #Properties of the coolant water
50 rho_w = 999.73
51 cp_w = 4203
52 Lambda_w = 0.574
53 a_w = 0.138e-6
54 def mu_w(T):
55     A=1.1709
56     B=0.001827
57     C=89.93
58     mu_20 = 1.0020e-3
59     return mu_20 *10**((A*(20-T)-B*(T-20)**2)/(T+C))
```

```

60 def Pr_w(T):
61     return mu_w(T)*cp_w/Lambda_w
62
63 #Properties of surrounding
64 T_s = 40
65 Lambda_s = 0.6274
66 rho_s = 992.95
67 cp_s = 4183.3
68 nu_s = mu_w(T_s)/rho_w
69 Pr_s = mu_w(T_s)*cp_s/Lambda_s
70 beta_s = 3.85e-4
71 g = 9.81
72
73
74
75 ###Design properties
76 #Properties of loop
77 d = 0.002 #wall thickness
78 radius = 0.003 #inner tube radius
79 D = 2*radius #inner tube diameter
80 D_out = 2*radius + 2*d #outer tube diameter
81 Radius = 0.008 #inner tube radius of node C
82 D_C = 2*Radius #inner tube diameter of node C
83 D_C_out = 2*Radius + 2*d #outer tube diameter of node C
84 D_h = D_C-D-2*d #wetted perimeter of node C
85
86 #Properties of the heat exchanger
87 A_C = np.pi*(Radius**2-(radius+d)**2) #crosssectional area node C
88 V_C = A_C*l #volume of node C
89 v_C = phi_mC/(rho_w*A_C) #coolant velocity
90
91 #Node volume
92 V_j = np.pi*radius**2*l
93 def V_jt(radius):
94     return np.pi*l*(d**2+2*d*radius)
95
96
97
98 ###Functions
99 #Fission
100 Ef = 192*1.60217662e-13 #192 MeV to Joules
101 sigma = 583e-28 #583 barn
102 enr = 0.1975 #enrichment
103 N_A = 6.022e23
104 phi_n = 3.5e16 #[1/(m^2 s)]
105 Q = V_j*Ef*sigma*enr*(c*1000)*N_A*phi_n/M #concentration*1000: g/L to g/m^3
106
107 #Gamma heating
108 u = 300 #300 W/kg tube material
109 def Pg(radius):
110     return u*rho_t*V_jt(radius)
111
112 #Heat transfer coefficient for flow in a circular tube
113 def h1(v,T_j):
114     T = T_j
115     Re = rho*np.abs(v)*D/mu(T,c_mol)
116     Gz = a*l/(np.abs(v)*D**2)
117     if Re>1e4 and Pr(T,c_mol)>=0.7:
118         h1 = 0.027*Re**(0.8)*Pr(T,c_mol)**(0.33)*Lambda/D
119     elif Gz <= 0.05:
120         h1 = 1.62*Gz**(-1/3)*Lambda/D
121     elif Gz > 0.05: #Should be Gz>0.1

```

```

122     h1 = 3.66*Lambda/D
123     else:
124         raise Exception("h1 out of range")
125     return h1
126
127 #Heat transfer coefficient for flow in a concentric tube: heat exchanger
128 def h1_C(T_oC):
129     T = (T_inC+T_oC)/2
130     Re = rho_w*v_C*D_h/mu_w(T)
131     Gz = a_w*1/(v_C*D_h**2)
132     if Re>1e4 and Pr_w(T)>=0.7:
133         h1_C = 0.027*Re**(0.8)*Pr_w(T)**(0.33)*Lambda_w/D_h
134     elif Gz <= 0.05:
135         h1_C = 1.62*Gz**(-1/3)*Lambda_w/D_h
136     elif Gz > 0.05:
137         h1_C = 3.66*Lambda_w/D_h
138     else:
139         raise Exception("h1_C out of range")
140     return h1_C
141
142 #Heat transfer coefficient for natural convection above a heated horizontal
    cylinder
143 #Correlation from Churchill and Chu
144 def h3_hor(T_w_j,D):
145     Gr = g*beta_s*(np.abs(T_w_j-T_s))*D**3/(nu_s**2)
146     Ra = Gr*Pr_s
147     h3hor = 1
148     if 1e-5 < Ra < 1e12:
149         h3hor = (0.6+ 0.387*Ra**(1/6))/( 1+ 0.559**(9/16)/Pr_s)**(8/27) )**2 *
    Lambda_s/D
150     elif Ra == 0:
151         h3hor=0
152     else:
153         raise Exception("h3_hor out of range: T_w_j = {}, Gr = {}, Ra = {}".format(
    T_w_j,Gr,Ra))
154     return h3hor
155
156 #Heat transfer coefficient for natural convection around a heated vertical cylinder
157 #Correlation by Xian, Jiang and Yu
158 def h3_ver(T_w_j,D):
159     Gr = g*beta_s*(np.abs(T_w_j-T_s))*1**3/(nu_s**2)
160     Ra = Gr*Pr_s
161     if Ra < 1e8: #conduction
162         Nu = 0.26
163     elif 1e8 < Ra < 1.45e14: #laminar and turbulent natural convection
164         A = np.log10(Ra**(0.25)*D/1)
165         Nu = Ra**(0.25) * 10**(0.090 - 0.449*A + 0.107*A**2 + 0.065*A**3)
166     else:
167         raise Exception("h3_ver out of range, Gr = {}, Ra = {}".format(Gr,Ra))
168     return Nu*Lambda_s/1
169
170 """
171 #Heat transfer coefficient for natural convection around a heated vertical cylinder
172 #Correlation by Le fevre
173 def h3_ver(T_w_j,D):
174     DeltaT = np.abs(T_w_j-T_s)
175     Gr = g*beta_s*(DeltaT)*1**3/(nu_s**2)
176     Ra = Gr*Pr_s
177     if Gr < 1e8: #conduction
178         Nu = 0.26
179     elif 1e8 < Gr < 4e9: #laminar natural convection

```

```

180     Nu = 4/3 * Ra**(0.25) * (7*Pr_s/(100+105*Pr_s))*0.25 + 4/35*(272+315*Pr_s)
      /((64+63*Pr_s) * 1/D
181     else:
182         raise Exception("h3_ver out of range, Gr = {}, Ra = {}".format(Gr,Ra))
183     return Nu*Lambda_s/l
184 """
185
186 #Darcy-Weisbach friction factor for uranyl nitrate solution
187 def fr(v,T_j):
188     Re = rho0*np.abs(v)*D/mu(T_j,c_mol)
189     if Re < 0:
190         raise Exception("Re<0 in friction factor")
191     A = 1/(1+(Re/2712)**(8.4))
192     B = 1/(1+(Re*eps/(150*D))**(1.8))
193     ffr = (64/Re)**A *(0.75*np.log(Re/5.37))**(2*B*(A-1)) *(0.88*np.log(6.82*D/eps)
      )**( 2*(A-1)*(1-B))
194     return ffr
195
196 #Darcy-Weisbach friction factor for water
197 def fr_C(v_C,T_oC):
198     T = (T_inC+T_oC)/2
199     Re = rho_w*v_C*D_h/mu_w(T)
200     A = 1/(1+(Re/2712)**(8.4))
201     B = 1/(1+(Re*eps/(150*D_h))**(1.8))
202     ffrC = (64/Re)**A *(0.75*np.log(Re/5.37))**(2*B*(A-1)) *(0.88*np.log(6.82*D_h/
      eps))**( 2*(A-1)*(1-B))
203     return ffrC
204
205 #heat flow towards the Dldr from node j
206 def phi_qDj(v,h3_j,T_j,T_w_j):
207     h1_j = h1(v,T_j)
208     return 2*np.pi*radius*1*h1_j* ( T_w_j - T_j )
209
210 #heat flow towards from Dldr to node C
211 def phi_qDC(T_oC,T_w_C):
212     T_Cavg = (T_inC + T_oC)/2
213     return 2*np.pi*Radius*1*h1_C(T_oC)* ( T_w_C - T_Cavg )
214
215 #heat flow from node B to C: heat exchanger
216 def phi_qH(T_B,T_D,T_oC,T_E,v):
217     if v >= 0:
218         if T_oC>T_D:
219             raise Exception("T_oC > T_D: value for phi_qH does not exist")
220             T_avg = (T_D + T_B)/2
221             U = 1/(1/h1(v,T_avg) + d/Lambda_t + 1/h1_C(T_oC))
222             A_H = 2*np.pi*radius*1 #surface heat exchanger
223             phi_qH = U*A_H*(T_B-T_inC-T_D+T_oC)/np.log((T_B-T_inC)/(T_D-T_oC))
224         else:
225             if T_oC>T_B:
226                 raise Exception("T_oC > T_B: value for phi_qH does not exist")
227                 T_avg = (T_E + T_B)/2
228                 U = 1/(1/h1(v,T_avg) + d/Lambda_t + 1/h1_C(T_oC))
229                 A_H = 2*np.pi*radius*1
230                 phi_qH = U*A_H*(T_E-T_inC-T_B+T_oC)/np.log((T_E-T_inC)/(T_B-T_oC))
231             return phi_qH
232
233
234
235 ###Set of ODE's: f
236 def f(r,t):
237     ##Variables
238     T_A = r[0]

```

```

239 T_D = r[1]
240 T_B = r[2]
241 T_E = r[3]
242 T_oC = r[4]
243 v = r[5]
244 T_w_A = r[6]
245 T_w_D = r[7]
246 T_w_E = r[8]
247 T_w_C = r[9]
248 ##Internal energy balance nodes j=A,D,B,E
249 h3_A = h3_hor(T_w_A,D_out)
250 h3_D = h3_ver(T_w_D,D_out)
251 h3_E = h3_ver(T_w_E,D_out)
252 h3_C = h3_hor(T_w_C,D_C_out)
253 if v>=0:
254     f_A = ( rho0*np.pi*radius**2*v*cp*(T_E-T_A) + Q + phi_qDj(v,h3_A,T_A,T_w_A)
255     )/ (V_j*cp*rho0)
256     f_D = ( rho0*np.pi*radius**2*v*cp*(T_A-T_D) + phi_qDj(v,h3_D,T_D,T_w_D) )/
257     (V_j*cp*rho0)
258     f_B = ( rho0*np.pi*radius**2*v*cp*(T_D-T_B) - phi_qH(T_B,T_D,T_oC,T_E,v) )/
259     (V_j*cp*rho0)
260     f_E = ( rho0*np.pi*radius**2*v*cp*(T_B-T_E) + phi_qDj(v,h3_E,T_E,T_w_E) )/
261     (V_j*cp*rho0)
262     ##Int EB node C
263     f_oC = (rho_w*np.pi*(Radius**2-(radius+d)**2)*v_C*cp_w*(T_inC-T_oC) +
264     phi_qH(T_B,T_D,T_oC,T_E,v) + phi_qDC(T_oC,T_w_C) )/(rho_w*V_C*cp_w)
265     ##Momentum function
266     T_avg = (T_A + T_D + T_B + T_E)/4
267     f_v = ( np.pi*radius**2*1*g*(rho(T_E)-rho(T_D) + (rho(T_B)-rho(T_A))*np.sin
268     (np.deg2rad(angle))) - fr(v,T_avg)*1/2*v**2 *2*np.pi*radius*1*rho0 )/(4*rho0*
269     V_j)
270     ##Int EB tube walls for nodes A,D,E,C
271     f_Tw_A = ( 2*np.pi*radius*1*h1(v,T_A)*(T_A - T_w_A) - 2*np.pi*(radius+d)*1*
272     h3_A*(T_w_A - T_s) + Pg(radius) )/(rho_t*V_jt(radius)*cp_t)
273     f_Tw_D = ( 2*np.pi*radius*1*h1(v,T_A)*(T_D - T_w_D) - 2*np.pi*(radius+d)*1*
274     h3_D*(T_w_D - T_s) + shielding_tf*Pg(radius) )/(rho_t*V_jt(radius)*cp_t)
275     f_Tw_E = ( 2*np.pi*radius*1*h1(v,T_E)*(T_E - T_w_E) - 2*np.pi*(radius+d)*1*
276     h3_E*(T_w_E - T_s) + shielding_tf*Pg(radius) )/(rho_t*V_jt(radius)*cp_t)
277     f_Tw_C = ( 2*np.pi*Radius*1*h1_C(T_oC)*((T_inC+T_oC)/2 - T_w_C) - 2*np.pi*(
278     Radius+d)*1*h3_C*(T_w_C - T_s) + shielding_tf*Pg(Radius) )/(rho_t*V_jt(Radius)*
279     cp_t)
280 else:
281     f_A = ( rho0*np.pi*radius**2*np.abs(v)*cp*(T_D-T_A) + Q + phi_qDj(v,h3_A,
282     T_A,T_w_A) )/ (V_j*cp*rho0)
283     f_E = ( rho0*np.pi*radius**2*np.abs(v)*cp*(T_A-T_E) + phi_qDj(v,h3_E,T_E,
284     T_w_E) )/ (V_j*cp*rho0)
285     f_B = ( rho0*np.pi*radius**2*np.abs(v)*cp*(T_E-T_B) - phi_qH(T_B,T_D,T_oC,
286     T_E,v) )/ (V_j*cp*rho0)
287     f_D = ( rho0*np.pi*radius**2*np.abs(v)*cp*(T_B-T_D) + phi_qDj(v,h3_D,T_D,
288     T_w_D) )/ (V_j*cp*rho0)
289     ##Int EB node C
290     f_oC = (rho_w*np.pi*(Radius**2-(radius+d)**2)*v_C*cp_w*(T_inC-T_oC) +
291     phi_qH(T_B,T_D,T_oC,T_E,v) + phi_qDC(T_oC,T_w_C) )/(rho_w*V_C*cp_w)
292     ##Momentum function
293     T_avg = (T_A + T_D + T_B + T_E)/4
294     f_v = -( np.pi*radius**2*1*g*(-rho(T_E)+rho(T_D) + (-rho(T_B)+rho(T_A))*np.
295     sin(np.deg2rad(angle))) - fr(v,T_avg)*1/2*np.abs(v)**2 *2*np.pi*radius*1*rho0
296     )/(4*rho0*V_j)
297     ##Int EB tube walls for nodes A,D,E,C
298     f_Tw_A = ( 2*np.pi*radius*1*h1(v,T_A)*(T_A - T_w_A) - 2*np.pi*(radius+d)*1*
299     h3_A*(T_w_A - T_s) + Pg(radius) )/(rho_t*V_jt(radius)*cp_t)

```

```

280     f_Tw_D = ( 2*np.pi*radius*1*h1(v,T_A)*(T_D - T_w_D) - 2*np.pi*(radius+d)*1*
h3_D*(T_w_D - T_s) + shielding_tf*Pg(radius) )/(rho_t*V_jt(radius)*cp_t)
281     f_Tw_E = ( 2*np.pi*radius*1*h1(v,T_E)*(T_E - T_w_E) - 2*np.pi*(radius+d)*1*
h3_E*(T_w_E - T_s) + shielding_tf*Pg(radius) )/(rho_t*V_jt(radius)*cp_t)
282     f_Tw_C = ( 2*np.pi*Radius*1*h1_C(T_oC)*((T_inC+T_oC)/2 - T_w_C) - 2*np.pi*(
Radius+d)*1*h3_C*(T_w_C - T_s) + shielding_tf*Pg(Radius) )/(rho_t*V_jt(Radius)*
cp_t)
283     return np.array([f_A,f_D,f_B,f_E,f_oC,f_v,f_Tw_A,f_Tw_D,f_Tw_E,f_Tw_C],float)
284
285
286
287     ###Initiate time array, varriable arrays, initial conditions and accuracy per unit
time
288     #Time interval and initial step size hin
289     t_initial = 0.0;
290     t_final = 1000.0;
291     hin = 0.2
292     h=hin
293     t_adaptive = []
294     t_adaptive.append(t_initial)
295     N = 0
296
297     #required accuracy per unit time.
298     #note: higher delta is required for longer running time.
299     # running 0.001 degrees error per second gives 1 error per 1000 seconds
300     delta0 = delta1 = delta2 = delta3 = delta4 = 0.00001 #target accuracy in degrees
celcius per second
301     delta5 = 0.00001 #target accuracy in m/s per second
302
303     #Initializing xpoints, vpoints and r with initial values
304     T_Apoints = []
305     T_Dpoints = []
306     T_Bpoints = []
307     T_Epoints = []
308     T_oCpoints = []
309     vpoints = []
310     T_w_Apoints = []
311     T_w_Dpoints = []
312     T_w_Epoints = []
313     T_w_Cpoints = []
314
315     #Initializing hpoints (not part of vector r)
316     hpoints = []
317     h_break = 1e-20 #break if h is lower than this value
318
319     #initial conditions:
320     r = np.array([40.0,40.0,40.0,40.0,15.0,1e-5,40.0,40.0,40.0,40.0],float) #isothermal
initial conditions
321     #r = np.array([46.13, 46.90, 35.91, 38.84, 12.59, 0.01748, 45.18, 46.54, 43.53,
28.96 ],float) #steady state initial conditions
322
323     #Initializing empty errays (necessary for the check in the if statement)
324     k1 = k2 = k3 = k4 = q1 = q2 = q3 = q4 = w1 = w2 = w3 = w4 = ri = r1 = r2 = np.zeros
(len(r))
325
326
327
328     ###Performing Adaptive Runge-Kutta
329     while t_adaptive[-1] < t_final:
330         t = t_adaptive[-1]
331         T_Apoints.append(r[0])
332         T_Dpoints.append(r[1])

```

```

333 T_Bpoints.append(r[2])
334 T_Epoints.append(r[3])
335 T_oCpoints.append(r[4])
336 vpoints.append(r[5])
337 T_w_Apoints.append(r[6])
338 T_w_Dpoints.append(r[7])
339 T_w_Epoints.append(r[8])
340 T_w_Cpoints.append(r[9])
341 r0 = r
342 try:
343     #perform one step of size h and calculate the new r values
344     k1 = h*f(r,t)
345     k2 = h*f(r+0.5*k1,t+0.5*h)
346     k3 = h*f(r+0.5*k2,t+0.5*h)
347     k4 = h*f(r+k3,t+h)
348     ri = r+(k1+2*k2+2*k3+k4)/6
349     if np.isnan(ri).any()==True:
350         raise Exception("ri contains non-numerical value")
351
352     #perform another step of size h and find our estimate r1 of r(t+2)
353     q1 = h*f(ri,t+h)
354     q2 = h*f(ri+0.5*q1,(t+h)+0.5*h)
355     q3 = h*f(ri+0.5*q2,(t+h)+0.5*h)
356     q4 = h*f(ri+q3,(t+h)+h)
357     r1 = ri + (q1+2*q2+2*q3+q4)/6
358     if np.isnan(r1).any()==True:
359         raise Exception("r1 second step contains non-numerical value")
360
361     #perform one step of size 2h and find our estimate r2 of r(t+2)
362     w1 = 2*h*f(r0,t)
363     w2 = 2*h*f(r0+0.5*w1,t+0.5*2*h)
364     w3 = 2*h*f(r0+0.5*w2,t+0.5*2*h)
365     w4 = 2*h*f(r0+w3,t+2*h)
366     r2 = r0 + (w1+2*w2+2*w3+w4)/6
367     if np.isnan(r2).any()==True:
368         raise Exception("r2 contains non-numerical value")
369
370     #calculate the new step size
371     ratio0 = 30*h*delta0 / np.abs(r1[0]-r2[0])
372     ratio1 = 30*h*delta1 / np.abs(r1[1]-r2[1])
373     ratio2 = 30*h*delta2 / np.abs(r1[2]-r2[2])
374     ratio3 = 30*h*delta3 / np.abs(r1[3]-r2[3])
375     ratio4 = 30*h*delta4 / np.abs(r1[4]-r2[4])
376     ratio5 = 30*h*delta5 / np.abs(r1[5]-r2[5])
377     ratio = min(ratio0,ratio1,ratio2,ratio3,ratio4,ratio5)
378     h_new = h*ratio**(1/4)
379     if h_new > 2*h:
380         h = 2*h
381     else:
382         h = h_new
383
384     #perform one step of size h_new
385     k1 = h*f(r,t)
386     k2 = h*f(r+0.5*k1,t+0.5*h)
387     k3 = h*f(r+0.5*k2,t+0.5*h)
388     k4 = h*f(r+k3,t+h)
389     r += (k1+2*k2+2*k3+k4)/6
390
391 except Exception as e:
392     ExceptionPresent = True
393     r = r0 #reset r
394     print("Exception was caught at t={}: ".format(round(t,2)))

```

```

395     print(e)
396     while True:
397         print("h = {}".format(h))
398         try:
399             h=0.7*h
400             k1 = h*f(r,t)
401             k2 = h*f(r+0.5*k1,t+0.5*h)
402             k3 = h*f(r+0.5*k2,t+0.5*h)
403             k4 = h*f(r+k3,t+h)
404             ri = r+(k1+2*k2+2*k3+k4)/6
405             #perform another step of size h and find our estimate r1 of r(t+2)
406             q1 = h*f(ri,t+h)
407             q2 = h*f(ri+0.5*q1,(t+h)+0.5*h)
408             q3 = h*f(ri+0.5*q2,(t+h)+0.5*h)
409             q4 = h*f(ri+q3,(t+h)+h)
410             r1 = ri + (q1+2*q2+2*q3+q4)/6
411             #perform one step of size 2h and find our estimate r2 of r(t+2)
412             w1 = 2*h*f(r0,t)
413             w2 = 2*h*f(r0+0.5*w1,t+0.5*2*h)
414             w3 = 2*h*f(r0+0.5*w2,t+0.5*2*h)
415             w4 = 2*h*f(r0+w3,t+2*h)
416             r2 = r0 + (w1+2*w2+2*w3+w4)/6
417
418             #calculate the new step size
419             ratio0 = 30*h*delta0 / np.abs(r1[0]-r2[0])
420             ratio1 = 30*h*delta1 / np.abs(r1[1]-r2[1])
421             ratio2 = 30*h*delta2 / np.abs(r1[2]-r2[2])
422             ratio3 = 30*h*delta3 / np.abs(r1[3]-r2[3])
423             ratio4 = 30*h*delta4 / np.abs(r1[4]-r2[4])
424             ratio5 = 30*h*delta5 / np.abs(r1[5]-r2[5])
425             ratio = min(ratio0,ratio1,ratio2,ratio3,ratio4,ratio5)
426             h_new = h*ratio**(1/4)
427
428             if h_new > h: #only allow h to become smaller now since we do not
want to undo the effect of h=0.7*h
429                 h = h
430             else:
431                 h = h_new
432
433             #perform one step of size h_new
434             k1 = h*f(r,t)
435             k2 = h*f(r+0.5*k1,t+0.5*h)
436             k3 = h*f(r+0.5*k2,t+0.5*h)
437             k4 = h*f(r+k3,t+h)
438             r += (k1+2*k2+2*k3+k4)/6
439
440         except:
441             h=h
442         finally:
443             if h<h_break:
444                 break
445             if r[5] > 0 and \
446                 r[4]+0.5*k1[4] <= r[1]+0.5*k1[1] and \
447                 r[4]+0.5*k2[4] <= r[1]+0.5*k2[1] and \
448                 r[4]+k3[4] <= r[1]+k3[1] and \
449                 ri[4]+0.5*q1[4] <= ri[1]+0.5*q1[1] and \
450                 ri[4]+0.5*q2[4] <= ri[1]+0.5*q2[1] and \
451                 ri[4]+q3[4] <= ri[1]+q3[1] and \
452                 r0[4]+0.5*w1[4] <= r0[1]+0.5*w1[1] and \
453                 r0[4]+0.5*w2[4] <= r0[1]+0.5*w2[1] and \
454                 r0[4]+w3[4] <= r0[1]+w3[1] and \
455                 r[4] <= r[1] and \

```

```

456         np.isnan((k4,q4,w4)).any() == False:
457             break
458         if r[5] < 0 and \
459             r[4]+0.5*k1[4]         <= r[2]+0.5*k1[2] and \
460             r[4]+0.5*k2[4]         <= r[2]+0.5*k2[2] and \
461             r[4]+k3[4]             <= r[2]+k3[2] and \
462             ri[4]+0.5*q1[4]        <= ri[2]+0.5*q1[2] and \
463             ri[4]+0.5*q2[4]        <= ri[2]+0.5*q2[2] and \
464             ri[4]+q3[4]            <= ri[2]+q3[2] and \
465             r0[4]+0.5*w1[4]        <= r0[2]+0.5*w1[2] and \
466             r0[4]+0.5*w2[4]        <= r0[2]+0.5*w2[2] and \
467             r0[4]+w3[4]            <= r0[2]+w3[2] and \
468             r[4]                   <= r[2] and \
469         np.isnan((k4,q4,w4)).any() == False:
470             break
471     finally:
472         #append time array
473         t_adaptive.append(t + h)
474         hpoints.append(h)
475         #continue to next time step
476         N += 1
477         if h<h_break:
478             print("h<",h_break)
479             break
480
481
482
483 ###plot
484 #Create time array
485 t = t_adaptive[0:len(t_adaptive)-1]
486 #Create figure
487 fig = plt.figure(figsize=(20,12))
488 #fig1
489 ax1 = fig.add_subplot(4, 3, 1)
490 ax1.set_xlabel("t(s)")
491 ax1.set_ylabel("$T_A$ ($\degree$C)")
492 ax1.set_title("$T_A$,t)-diagram")
493 ax1.plot(t,T_Apoints,'.')
494 #fig2
495 ax2 = fig.add_subplot(4, 3, 2)
496 ax2.set_xlabel("t(s)")
497 ax2.set_ylabel("$T_D$ ($\degree$C)")
498 ax2.set_title("$T_D$,t)-diagram")
499 ax2.plot(t,T_Dpoints,'.')
500 #fig3
501 ax3 = fig.add_subplot(4, 3, 3)
502 ax3.set_xlabel("t(s)")
503 ax3.set_ylabel("$T_B$ ($\degree$C)")
504 ax3.set_title("$T_B$,t)-diagram (N={} steps)".format(N))
505 ax3.plot(t,T_Bpoints,'.')
506 #fig4
507 ax4 = fig.add_subplot(4, 3, 4)
508 ax4.set_xlabel("t(s)")
509 ax4.set_ylabel("$T_E$ ($\degree$C)")
510 ax4.set_title("$T_E$,t)-diagram")
511 ax4.plot(t,T_Epoints,'.')
512 #fig5
513 ax5 = fig.add_subplot(4, 3, 5)
514 ax5.set_xlabel("t(s)")
515 ax5.set_ylabel("$T_{oC}$ ($\degree$C)")
516 ax5.set_title("$T_{oC}$,t)-diagram")
517 ax5.plot(t,T_oCpoints,'.')

```

```

518 #fig6
519 ax6 = fig.add_subplot(4, 3, 6)
520 ax6.set_xlabel("t(s)")
521 ax6.set_ylabel("v (m/s)")
522 ax6.set_title("(v,t)-diagram")
523 ax6.plot(t,vpoints,'.')
524 #fig7: step size throughout time
525 ax7 = fig.add_subplot(4, 3, 7)
526 ax7.set_xlabel("t(s)")
527 ax7.set_ylabel("h (s)")
528 ax7.set_title("(h,t)-diagram")
529 ax7.plot(t,hpoints,'.')
530 #fig8: wall temperature of node A
531 ax8 = fig.add_subplot(4, 3, 8)
532 ax8.set_xlabel("t(s)")
533 ax8.set_ylabel("$T_{w,A}$ ($\text{degreeC}$)")
534 ax8.set_title("$T_{w,A}$,t)-diagram")
535 ax8.plot(t,T_w_Apoints,'.')
536 #fig9: wall temperature of node D
537 ax9 = fig.add_subplot(4, 3, 9)
538 ax9.set_xlabel("t(s)")
539 ax9.set_ylabel("$T_{w,D}$ ($\text{degreeC}$)")
540 ax9.set_title("$T_{w,D}$,t)-diagram")
541 ax9.plot(t,T_w_Dpoints,'.')
542 #fig10: wall temperature of node E
543 ax10 = fig.add_subplot(4, 3, 10)
544 ax10.set_xlabel("t(s)")
545 ax10.set_ylabel("$T_{w,E}$ ($\text{degreeC}$)")
546 ax10.set_title("$T_{w,E}$,t)-diagram")
547 ax10.plot(t,T_w_Epoints,'.')
548 #fig11: wall temperature of node C
549 ax11 = fig.add_subplot(4, 3, 11)
550 ax11.set_xlabel("t(s)")
551 ax11.set_ylabel("$T_{w,C}$ ($\text{degreeC}$)")
552 ax11.set_title("$T_{w,C}$,t)-diagram")
553 ax11.plot(t,T_w_Cpoints,'.')
554
555 #adjust space between plots
556 plt.subplots_adjust(wspace=0.4,hspace=0.6)
557
558 ###Log
559 #find maximum bulk temperature
560 TmaxA = np.amax(T_Apoints)
561 TmaxD = np.amax(T_Dpoints)
562 TmaxB = np.amax(T_Bpoints)
563 TmaxE = np.amax(T_Epoints)
564 Tmax = max(TmaxA,TmaxD,TmaxB,TmaxE)
565
566 #log information
567 print("The adaptive Runge-Kutta calculation was initiated with a step size {},
    performed in {} steps and {} seconds, and using a target accuracy per unit time
    of {}, {}, {}, {}, {} degrees per second, {} m/s per second. The cooling
    conditions were T_inC = {} and phi_mC = {} kg/s and the concentration c = {} g/
    L. The final values are T_A = {}, T_D = {}, T_B = {}, T_E = {}, T_oC = {}, v =
    {} m/s. The maximum bulk temperature throughout time in all nodes was {}.".
    format(hin,N,round((time.time() - start_time),2),delta0,delta1,delta2,delta3,
    delta4,delta5,round(T_inC,2), round(phi_mC,2),c,round(T_Apoints[-1],2),round(
    T_Dpoints[-1],2),round(T_Bpoints[-1],2),round(T_Epoints[-1],2),round(T_oCpoints
    [-1],2),round(vpoints[-1],5),round(Tmax,2)))
568 print("The final wall temperatures were T_w_A = {}, T_w_D = {}. T_w_E = {}, T_w_C =
    {}".format(round(T_w_Apoints[-1],2),round(T_w_Dpoints[-1],2),round(T_w_Epoints
    [-1],2),round(T_w_Cpoints[-1],2) ))

```

```
569 print("The angle was {} degrees".format(angle))
```