A FINITE ELEMENT MODEL FOR THERMAL-HYDRAULIC CALCULATIONS ON THE U-BATTERY®

TU Delft

C.C.E.M. Orbons | 4480414

A Finite Element Model for Thermal-Hydraulic Calculations on the U-Battery®

by

C.C.E.M. Orbons

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday December 17, 2021 at 2:00 PM.

Student number:	4480414	
Project duration:	March, 2021 – December, 2021	
Thesis committee:	Dr.ir. D. Lathouwers,	TU Delft, supervisor
	Prof.dr.ir. J.L. Kloosterman,	TU Delft, supervisor
	Drs.ir. M. van Den Berg,	TU Delft, supervisor
	Dr. Z. Perkó,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

A Finite Element Model for Thermal-Hydraulic Calculations on the U-Battery®

C.C.E.M. Orbons

Abstract

The U-Battery® is a small modular reactor based on the design of the high temperature gas-cooled reactor. For further development and licensing of the U-Battery® an accurate multi-physics model is needed as a benchmark for neutronics and temperature calculations. In this thesis a time-dependent thermal-hydraulics code is designed for temperature calculations in the coolant and solids of the U-Battery® core. First, two separate steady-state finite element models were created. One for heat convection in the coolant channels based on the discontinuous Galerkin method and one for heat diffusion in the solids based on the symmetric interior penalty Galerkin method. These separate models were combined in an iterative model for the steady-state calculations of the U-Battery® core. The Crank-Nicolson method for time integration was applied to the coupled model for time-dependent calculations.

The separate codes were verified using analytically solvable problems and show a theoretical convergence to second-order accuracy using linear basis functions. The performance of the coupled system was investigated by simulating a single coolant channel surrounded by graphite over the height of the core. The solid materials were modeled using linear prismatic elements. When increasing the number of elements in the mesh the overall temperature decreases. This is caused by differences in the boundary surface area and volume of the channel when refining the mesh. Additionally, calculations were done on an insulated fuel block for the U-Battery® with 108 coolant channels and 216 fuel compacts. The steady-state temperature was calculated as well as the temperature during a transient for the hypothetical blockage of various coolant channels. The maximum temperature during these transients is within safety limits for damage to the TRISO fuel. Oscillations of the Crank-Nicolson method in the temperature are seen in the first time steps of the simulation. The variations in temperature are a few Kelvin only.

Acknowledgements

I would like to thank everyone who has been there for me during this thesis. First and foremost, I would like to thank my supervisors Danny, Jan Leen and Marc. Thank you Danny, for your daily supervision and sacrificing your time to lengthy code viewing sessions. Thank you Jan Leen for your guidance and enthusiasm during our weekly meetings on the project. Thank you Marc for being such a motivational supervisor. I really enjoyed our time working together on the project. I wish you all the best with the continuation of your PHD and hope you will have some time left for traveling the world.
I would also like to thank my friends and family for their support. Thank you to my study friends for motivating me from our first day as physics students in Delft and a special thanks to Rixt for sharing many free coffees. Thank you to my housemates for our relaxing 'vrijmibo's' and delicious dinners at the end of my long days and thank you to my old housemate Rens for our weekly climbing and study sessions. Thank you to my siblings Julia and Luc for their support. Finally, I would like to thank the main sponsor of my education and my most loyal supporter: My mother. Thank you mom.

Cleo Orbons
Delft, December 2021

Contents

1 Introduction

The global need for clean and reliable energy is urgent. Recently, nuclear power got renewed interest as a reliable source of carbon-free energy. In particular the potential of small modular reactors (SMRs) is investigated for industrial purposes. Due to their reduced size and nuclear power SMRs are simpler in operation, easier to build and safer compared to conventional reactors. SMRs are being designed in different countries and, very recently, in November 2021 the United Kingdom announced to invest half a billion pounds in the development and construction of SMRs [1] [2].
A design of the SMR that is currently under development is the U-Battery®. The U-Battery® is a high temperature gas-cooled reactor (HTGR) designed to have an output power of 20MWth and a lifetime between 5 and 10 years without refueling. The name of the design refers to batteries to emphasize the scalable, long-lasting design of the reactor [3]. For developing and licensing of the U-Battery® it is necessary to have an accurate multi-physics model that could provide a benchmark for simplified and faster models. Since the power production in nuclear fuel is dependent on the fuel's temperature, a multi-physics model should combine the complex neutronics in the nuclear reactor with thermal-hydraulics calculations in the fuel, moderator, coolant and reflector. This thesis will focus on the latter of these requirements: providing a thermal-hydraulics code for simulation of the temperature inside the U-Battery® reactor core. Before elaborating the goals and outline of this thesis the U-Battery® design is briefly discussed and an overview of recent work on the thermal-hydraulics of the U-Battery® is given.

1.1. U-Battery® Design

The U-Battery® is a commercial SMR. It was designed to provide reliable, low carbon and cost effective energy to secluded areas or industries. The initial design feasibility of the U-Battery® was investigated by researchers of the universities of Delft and Manchester in 2011 and is based on the mature design of the HTGR [3]. Additionally, the cost-effectiveness of building and mass production and the logistics needed for operation and plant layout were investigated [3]. In the final report for this research two designs where proposed: a 20MWth and a 10 MWth design. Both designs were based on the design of an HTGR moderated with graphite and cooled with high pressurized helium gas. The outlet temperature of HTGRs is higher compared to conventional light water reactors that use water as coolant. The higher outlet temperature of HTGRs leads to a higher conversion efficiency of thermal energy to electricity and makes the reactors better applicable for the use of heat in industrial processes. For the fuel of the reactor different fuel types were investigated including UO_2-TRISO particles

Table 1.1: Basic design parameters for the U-Battery® [3].

Parameter	Value
Reactor type	Block-Type HTGR
Thermal Power	20 MWth
Coolant	Helium
Moderator	Graphite
Pressure	4.0 MPa
Inlet Temperature	250 °C
Outlet Temperature	750 °C
Core diameter, height	$\leq$ 3.7m, 3.2m
Fuel type	UO_2, TRISO coated fuel

and variations of uranium-thorium fuel blocks [4], [5], [6]. In this thesis the temperature will be modeled inside the reactor core based on the original 2011 20MWth design with TRISO fuel particles. The used design parameters for the U-Battery® are given in Table 1.1. Although the U-Battery® is a complete package of a power plant, this thesis will focus on the components inside the reactor pressure vessel (RPV) and more specifically on the core of the reactor including the fuel and reflector zones. A schematic overview of the RPV and the internals is given in Figure 1.1 [3].

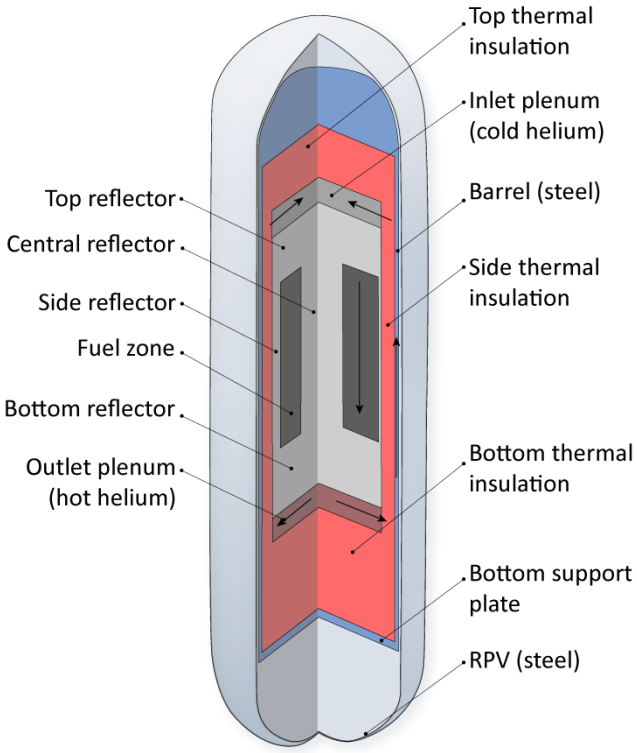
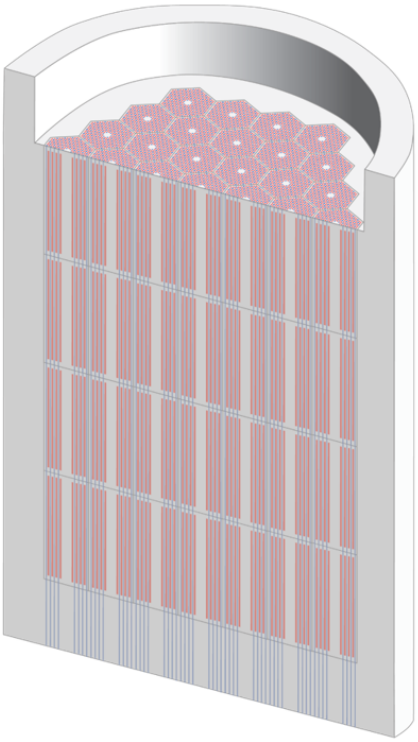


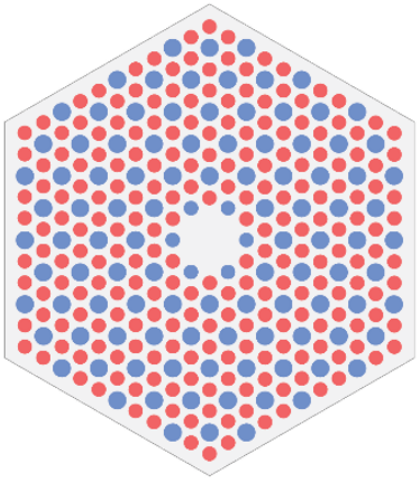
Figure 1.1: RPV layout for the U-Battery® [3].

The prismatic core of the U-Battery® is made up of the fuel and reflector zones. The fuel zone consists of hexagonal fuel blocks with channels for fuel, coolant and control rods. There are 4 layers of 37 of these fuel blocks. These fuel blocks are surrounded by a graphite reflector

as shown in Figure 1.2a. The cross-section of a single fuel block is shown in Figure 1.2b. This is a 36cm wide hexagonal block of graphite with 216 fuel channels and 108 coolant channels with diameters of 1.27 cm and 1.588 cm respectively [3]. The graphite acts as a moderator to slow down neutrons to thermal energy levels. The coolant channels are filled with helium gas that is pumped downwards through the channels under high pressure. The fuel of the U-Battery® consist of TRISO particles with low-enriched UO_2. TRISO particles are small spheres of UO_2 coated with three different carbide layers that can withstand temperatures up to 1600 °C without damage. The particles are packed in small cylindrical fuel compacts mixed with graphite. These compacts are stacked in fuel rods [3].



(a) U-Battery® core design with a 4x37 fuel block layout surrounded by a graphite reflector.



(b) Fuel block cross-section for the U-Battery®. Red dots are fuel compacts with a diameter of 1.27cm, blue dots are coolant channels with a diameter of 1.588cm. The six cool channels in the middle of the fuel block have a diameter of 1.27cm.

Figure 1.2: Core and fuel block layout for the U-Battery®.

1.2. Recent work on this topic

As mentioned earlier, the U-Battery® is designed as a small version of the helium cooled HTGR with a prismatic block core [3]. HTGRs with prismatic cores have been investigated since the 1940's and the first reactors were build in the sixties in Europe and the US. These early HTGRs were graphite moderated and used pressurized CO_2 as coolant due to the high costs of helium [2], [7]. Many research has been done on the heat transport in these gas cooled reactors and [7] and [8] give a clear overview regarding the thermal-hydraulics calculations for pressurized helium and nuclear graphite. However, the HTGRs currently in use do not use TRISO fuel particles and operate at a lower temperature than the U-Battery® design.
For the U-Battery® specifically the thermal-hydraulic transient calculations were done by M. Ding [9], [10]. In this research heat transfer calculations were done using the steady-state coupled neutronic/thermal-hydraulic system DALTON/THERMIX. DALTON is a TU-Delft in-house code for 3D time-dependent neutron diffusion and THERMIX a 2D thermal-hydraulics

code [11]. In addition to general thermal-hydraulic evaluations at steady-state operation, calculations for the U-Battery® were performed on two transients [10]: a depressurized loss of forced cooling (DLOFC) event and a pressurized loss of forced cooling (PLOFC) event. In both events the cooling of the reactor is shut down and the fission reaction is stopped by dropping the control rods. In the DLOFC event the coolant flow in the entire reactor stops and the pressure drops to atmospheric pressure. Due to the low pressure the heat transfer through natural convection is negligible. In a PLOFC event the coolant flow stops as well, but the pressure is kept at operational pressure. In this event the natural convection in the coolant becomes the main form of heat transfer [3], [9], [10].
The thermal-hydraulic calculations performed by M. Ding form a basis for safety analyses and the general temperature profile inside the U-Battery®. However, for modelling the best possible estimate of the temperature inside the U-Battery®, the calculations of the THERMIX code need to be improved. THERMIX is a two-dimensional code, modeling the reactor as a slab in the radial and axial direction. In general, this a reasonable assumption since the reactor core is for a large part cylindrical symmetric. However, due to the edges of the prismatic blocks the reactor is not entirely cylindrical and for investigating non-symmetrical events a three-dimensional code is needed. Additionally, THERMIX models the core as a single homogeneous medium with a constant porosity to account for the coolant channels. In doing so the geometry of the coolant channels, fuel channels and graphite is lost. The THERMIX code is thus not applicable to transients in zoomed in parts of the reactor and non-symmetrical events where for instance certain coolant channels are blocked.

1.3. Research goal and report structure

The analysis of Ding's research leads to the goal of this thesis: to provide a new basis code for thermal-hydraulic simulations for the U-Battery®. The main goal of this code is accuracy, providing a best-estimate calculation for the temperature in the reactor core and coolant. The code will provide a three-dimensional finite element model of the U-Battery® for transient calculations on the temperature based on a given power input. The goal of this code is to be used in a combined neutronic/thermal-hydraulic system where the output of the thermal-hydraulics code will be used to update the neutron cross-sections and power production in a neutronics code. The combined system will function as a benchmark for calculations on the U-Battery®. For the scope of this thesis the power input is assumed to be known. Since the temperatures for the coolant and solids (fuel, moderator, reflector) are defined by different governing equations, two separate models were created; one for the coolant and one for the solids. Eventually, these codes are combined in an iterative coupled system and time dependence is taken into account using a finite difference integration method.
The structure of this thesis is as follows: The following chapter will discuss the steady-state heat transfer models for coolant and solid temperatures. Starting with a one-dimensional model for the coolant using the discontinuous Galerkin finite element method for heat convection problems. Next, the solid temperature is calculated using the symmetric interior penalty Galerkin method for three-dimensional diffusion problems. This chapter finishes with an overview of the implementation of linear prismatic elements for meshing of the U-Battery® core. The third chapter explains the coupling of the two steady-state codes in an iterative and time-dependent code for the complete thermal-hydraulics system. Chapter four begins with the verification of the two steady-state models from chapter two followed by the time-dependent thermal-hydraulic code from chapter three. In the fifth chapter the working of the code is showcased for a fuel block composition of the U-Battery®. The steady-state operation and a transient for an asymmetrical blockage of different coolant channels is simulated given a standard power profile. Finally, conclusions and recommendations are drawn up in chapter six.

2

Models for steady-state heat transfer

In a nuclear reactor the transfer of heat is essential for the operation of the reactor. Fission reactions in the fuel rods produce heat. This heat is dissipated from the reactor core by the coolant. In the U-Battery® heat is produced in the TRISO Uranium fuel kernels. From here, the heat is conducted to the solid graphite moderator surrounding the fuel rods. From the graphite the heat is transferred to the helium gas flowing through the coolant channels. This heated gas is used to generate electricity or is used as process heat. The differences in temperature over the core of the reactor are large. The temperature can vary from 250 °C at the inlet up to 1000 °C in the fuel rods [3], [10]. This chapter gives the theoretical framework and model overview for the steady-state temperature calculations on the U-Battery®. This includes two finite element codes; one code for the heat convection in the coolant channels and one for the heat diffusion in the solids inside the core.

2.1. Model for coolant temperature

The code for the coolant temperature is based on a one-dimensional convection problem over the height of the coolant channels. This section will start with showing how the convection problem over the channel height leads to a partial differential equation for the temperature. Following this, the finite element discontinuous Galerkin (DG) method is used to solving this partial differential equation. At last, an overview is given of the final algorithm used for solving the DG matrix system.

2.1.1. Heat convection in coolant channels

Pressurized helium gas is pumped through the coolant channels of the U-Battery®. When entering the core of the U-Battery® at the top of the core the colder helium is heated by the warmer walls of the channels. After exiting the core at the bottom the temperature of the helium gas will have increased with 500 degrees on average [3]. Since the helium itself is moving, this process is a combination of heat diffusion and heat advection. The combination of these two means of transfer will be referred to as heat convection. There are two types of convection: natural and forced. Natural convection is caused by differences in the material density. Forced convection is caused by external forces. Since the helium gas in the U-Battery® is pumped through the reactor under high pressure, this flow is considered to be forced.

To find the governing equations for the temperature of the coolant, first a single coolant channel is considered. In this channel the helium enters at the top and is forced to flow downwards under a constant pressure of 40MPa and a constant mass flow rate ϕ_m in [g/s]. It is assumed that there is sufficient mixing in the helium such that the temperature variations along the radial

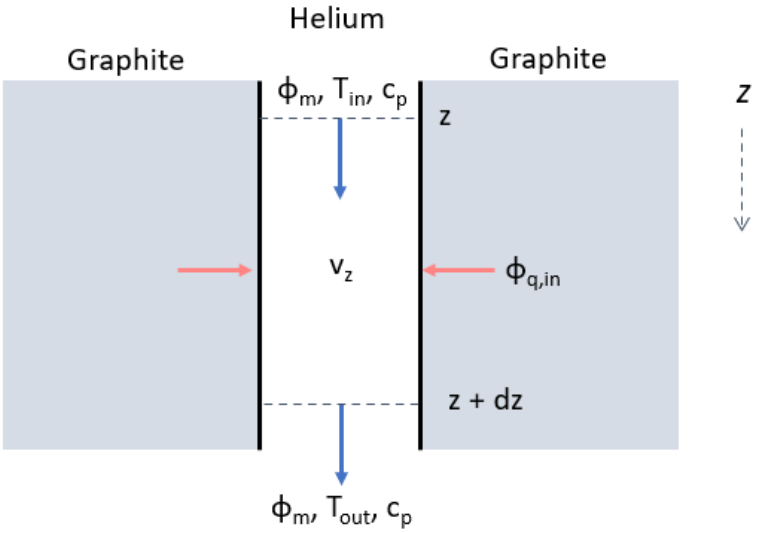


Figure 2.1: Schematic overview of heat transfer in a single coolant channel over a control height $[z, z + dz]$

direction are neglected. A volume inside this channel is considered over a control height on the interval $[z, z+dz]$. The axial direction z is oriented downwards since the helium flows from the top of the core to the bottom. This situation is shown in Figure 2.1 along with the relevant properties of the helium and graphite. The energy balance over the control volume is then given by:

$$\phi_m c_p \left[T(z + dz) - T(z)\right] = \phi_{q,in} \, . \tag{2.1}$$

Here ϕ_m is the mass flux in [g/s], c_p is the heat capacity of the helium in [J/K/g] and $\phi_{q,in}$ is the inward heat in [W] [7], [12]. For the coolant channels of the U-Battery® $\phi_{q,in}$ is the inflow of heat from the warmer graphite moderator to the colder helium gas. The amount of incoming energy is dependent on the temperature difference between the graphite and the helium at the boundary of the channel. The heat flux from the graphite to the helium is described by Newton's law of cooling. This law gives the heat flux as the difference in temperature ΔT multiplied by the area A and the heat transfer coefficient γ [12]:

$$\phi_q = \gamma A \Delta T \, . \tag{2.2}$$

For the described control volume in the single coolant channel, the area in the heat flux equation is given by the circumference of the coolant channel multiplied with the control height dz. For a channel with diameter D_{He} this gives: $A = \pi dz D_{He}$. ΔT is calculated as the difference between the averaged graphite temperature over the circumference of the channel, $\langle T_w(z) \rangle$, and the helium temperature at height z. The value of the heat transfer coefficient γ is dependent on the type of flow in the core. Table 2.1 and Table 2.2 summarize the properties of the flow and the properties of helium at the operational pressure and in and outlet temperature of the coolant. For $Re > 3000$, the Gnielinski equation can be used to compute the heat transfer coefficient in a tube [13]:

$$\gamma = Nu \frac{\lambda_{He}}{D_{He}} = \frac{\frac{f_D}{8}(Re - 1000)Pr}{1 + 12.7\sqrt{\frac{f_D}{8}}(Pr^{2/3} - 1)} \frac{\lambda_{He}}{D_{He}} \, . \tag{2.3}$$

The Darcy friction factor f_D in this formula is calculated using the Koo correlation [7];

$$f_D = 4f = 4\left(0.0014 + 0.125 Re^{-0.32}\right) \, , \tag{2.4}$$

Table 2.1: Properties for the coolant flow of the U-Battery® [3], [7].

parameter	value	units	meaning
N_{cool}	3240	-	Number of coolant channels
D_{He}	1.588	cm	Coolant channel diameter
$\phi_{m,tot}$	7640	g/s	Total mass flux
ϕ_m	2.36	g/s	Coolant channel mass flux
T_{in}	522	K	Inlet temperature
T_{out}	1022	K	Outlet temperature
P	40	Mpa	Helium pressure

Table 2.2: Relevant properties for heat transfer between the helium in the coolant channels of the U-Battery® and the graphite in the moderator. All calculations were based on $T_{in} = 522K$ and $T_{out} = 1022K$ and with a constant pressure of 40Mpa [14].

parameter	value at T_{in}	value at T_{out}	units	meaning
ρ	0.0037	0.0019	g/cm^3	Helium density
c_p	5.195	5.195	J/g/K	Helium specific heat capacity
μ	3.04 $\cdot 10^{-4}$	4.78 $\cdot 10^{-4}$	g/cm/s	Helium dynamic viscosity
v	324	631	cm/s	Flow velocity
λ	0.0023	0.0037	W/cm K	Helium thermal conductivity
Re	6448	4023	-	Reynolds number
Pr	0.70	0.70	-	Prandtl number
f_D	0.036	0.041	-	Darcy friction factor
Nu	21	13	-	Nusselt number
γ	0.030	0.031	W/cm^2 K	Heat transfer coefficient

which is valid for $3000 < Re < 3000000$. Using these relations for the heat transfer coefficient gives the heat flux $\phi_{q,in}$. Substituting this new found term for $\phi_{q,in}$ in equation 2.1 gives the equation for the helium temperature:

$$\phi_m c_p \frac{dT(z)}{dz} = \pi D_{He} \gamma [\langle T_w(z) \rangle - T(z)] \, . \tag{2.5}$$

This equation is a linear first-order differential equation for the temperature over the height of the channel. The next section will explain the numerical method for solving this equation.

2.1.2. Discontinuous Galerkin finite element method

To find the solution to the partial differential equation for the convection problem the discontinuous Galerkin finite element method is used. The discontinuous Galerkin (DG) method is a class of finite element methods first introduced by Reed and Hill in 1973 as a solution to the neutron transport equation. It has since been widely used in other areas such as computational fluid dynamics. The method is stable, accurate and can easily be applied to different types of meshes. Similar to other finite element methods, in the DG method the interval over which the solution is approximated is divided into smaller intervals. On each of these intervals the solution is approximated by the sum of basis functions and coefficients. The main difference between the DG method and conventional finite element methods is that the DG method is only piecewise-continuous. DG allows for discontinuities in the solution between elements. These discontinuities are the strength of the DG method. In this section a DG system for solving the one-dimensional heat convection equation in 2.5 is be derived [15], [16].

As was shown in the previous section, the first-order differential problem for the temperature in the coolant channels is described by:

$$\phi_m c_p \frac{dT(z)}{dz} = \pi D \gamma [\langle T_w(z) \rangle - T(z)], \quad \text{with} \quad z \in (0, H), \quad T(0) = T_{inlet} \, . \tag{2.6}$$

The helium enters the core with a constant temperature T_{inlet}. Similar to the control volume in the previous section the positive z-direction is oriented downwards. The height at the top of the core is 0 and the height at the bottom of the core is H. When flowing through the core the helium is heated by the graphite walls. Solving this problem following the standard DG method for solving partial differential equations requires three steps [15]:

1. introducing the discontinuous approximation
2. deriving the Galerkin weak formulation
3. applying the boundary conditions for the specific problem.

First, the discontinuous approximation for the problem in 2.6 is introduced. This is done by partitioning the one-dimensional domain as shown in Figure 2.2. The domain $z \in (0, H)$ is partitioned in a set of N elements e_k. Here $k = 1, 2 \ldots, N$ and each element has a size Δz_k. The boundaries of element k are given by $z_1^k = z_{k-1/2}^k$ and $z_2^k = z_{k+1/2}^k$. On a single element the solution T is approximated by u which is a summation of I linear basis functions $h_i^k(z)$, (where $i = 1, 2, \ldots, I$) multiplied by expansion coefficients T_j^k. For this problem two first-order polynomials are used as basis functions. These basis functions and the approximation for the temperature are written as:

$$h_1^k(z) = 1 - \frac{z - z_1^k}{\Delta z^k}, \quad h_2^k(z) = 1 + \frac{z - z_2^k}{\Delta z^k}, \qquad T(z) \approx u = \sum_{k=1}^{N} \sum_{j=1}^{2} T_j^k h_j^k(z) \, . \tag{2.7}$$

This equation gives the temperature as a summation of functions in a discretized domain. On each element of this domain the temperature is a weighted sum of the basis functions.
Now that the discontinuous approximation for T is introduced, the second step is to derive the

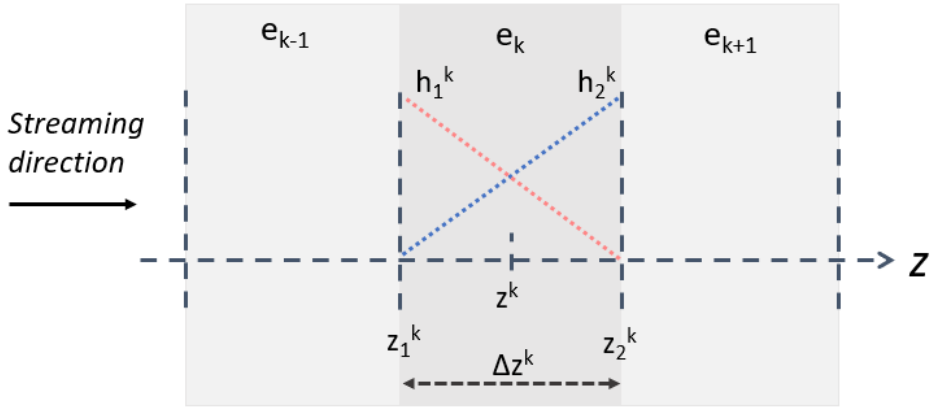


Figure 2.2: Schematic overview of an element z^k with the boundaries of the element and basis functions h_i^k.

Galerkin weak formulation of problem 2.6. In general, the weak formulation for DG problems in a function space V_k is given by the Galerkin equation:

$$\text{Find } u \in V_k \text{ such that } \; a(u, v) = L(v) \;\; \forall v \in V_k \, , \tag{2.8}$$

where $a(u,v)$ is called the bilinear form and $L(v)$ is the linear form. Here u is the solution to the problem and v is a test function sharing the same function space V_k on the element k. Equation 2.6 is rewritten to the DG weak formulation by substituting T with the approximation u, multiplying the equation with the test function v and integrating over an element e_k. This leads to the weak form:

$$\phi_m c_p \int_{e_k} \frac{duv}{dz} - \phi_m c_p \int_{e_k} u\frac{dv}{dz} = \pi D\gamma \int_{e_k} [\langle T_w \rangle - u]\, v \,, \tag{2.9}$$

here the left side of the equation was integrated by parts. Thus in the DG formalism the bilinear and linear forms are given by:

$$a(u,v) = \phi_m c_p \int_{e_k} \frac{duv}{dz} - \phi_m c_p \int_{e_k} u\frac{dv}{dz} + \pi D\gamma \int_{e_k} uv \,, \qquad L(v) = \pi D\gamma \int_{e_k} \langle T_w \rangle\, v \,. \tag{2.10}$$

After defining the approximation space and determining the weak form, the last step is applying the boundary and streaming conditions. The first integral in the bilinear form can be worked out analytically. Using the boundary condition for the inlet temperature and for the streaming property gives:

$$\phi_m c_p \int_{e_k} \frac{duv}{dz} = \phi_m c_p \left[u(z_2^k)v(z_2^k) - u(z_1^k)v(z_1^k)\right] = \begin{cases} \phi_m c_p \left[u(z_2^k)v(z_2^k) - T_{inlet}v(z_1^k)\right], & \text{if } k = 1 \\ \phi_m c_p \left[u(z_2^k)v(z_2^k) - u(z_2^{k-1})v(z_1^k)\right], & \text{if } k \neq 1 \,. \end{cases} \tag{2.11}$$

Now a system according to the formalism in 2.8 is derived. For each element e_k the temperature can be found by working out 2.10. This is done by substituting the temperature approximation in the weak form and using the basis functions h_i^k for the test function v:

$$\begin{aligned} \phi_m c_p T_2^k h_i^k(z_2^k) \;-\; \phi_m c_p \sum_{j=1}^{2} T_j^k \int_{e_k} h_j^k \frac{dh_i^k}{dz} \;+\; \pi D\gamma \sum_{j=1}^{2} T_j^k \int_{e_k} h_i^k h_j^k \;=& \\ \phi_m c_p T_2^{k-1} h_i^k(z_1^k) \;+\; \pi D\gamma \int_{e_k} \langle T_w \rangle\, h_i^k \;,& \end{aligned} \tag{2.12}$$

where $T_2^{k-1} = T_{inlet}$ if $k = 1$. Equation 2.12 is a system of two equations with two unknowns on each local element e_k; the expansion coefficients T_1^k and T_2^k. Solving this system for the expansion coefficients gives the solution for the approximated temperature u.
When implementing this matrix system in a computational code the terms in 2.12 need to be computed. The first term is the out-streaming term of thermal energy at the right-side of the element and can be written out by filling in the basis function. The second term describes the streaming of energy due to convection and is called the convection matrix. This term can be worked out by taking the integral over the basis functions:

$$\int_{e_k} h_j^k \frac{dh_i^k}{dz} = \begin{cases} -1/2, & \text{if } i = 1 \\ +1/2, & \text{if } i = 2 \,. \end{cases} \tag{2.13}$$

The third integral is called the mass matrix and is evaluated by filling in the basis functions as well:

$$\int_{e_k} h_i^k h_j^k = \begin{cases} \Delta z^k/3 & \text{if } i = j \\ \Delta z^k/6 & \text{if } i \neq j \,. \end{cases} \tag{2.14}$$

The fourth and fifth term are source vectors. The first source vector accounts for the in-streaming from the previous element. Since the inlet temperature of the previous element

is known, the system is solved from element 1 to N. This way the expansion coefficient of T_2^{k-1} is known when solving for element k. The last integral accounts for external sources due the heat exchange with the walls. This integral is worked out manually using second-order Gauss quadrature for a linear element with quadrature points $x_1^k = \Delta z^k \left(\frac{1-1/3\sqrt{3}}{2} + z_1^k\right)$ and $x_2^k = \Delta z^k \left(\frac{1+1/3\sqrt{3}}{2} + z_1^k\right)$ and quadrature weights $w_1 = w_2 = \frac{\Delta z^k}{2}$:

$$\pi D\gamma \int_{e_k} T_w h_i^k = \pi D\gamma \frac{\Delta z^k}{2} \begin{bmatrix} T_w(x_1)h_1^k(x_1) + T_w(x_2)h_1^k(x_2) \\ T_w(x_1)h_2^k(x_1) + T_w(x_2)h_2^k(x_2) \end{bmatrix} = \pi D\gamma \frac{\Delta z^k}{2} \begin{bmatrix} T_w^{k,1} \\ T_w^{k,2} \end{bmatrix} \quad . \tag{2.15}$$

Now that all integrals are evaluated the system in equation 2.12 combines to the local matrix system:

$$\begin{bmatrix} 2\alpha^k + \frac{1}{2} & \alpha^k + \frac{1}{2} \\ \alpha^k - \frac{1}{2} & 2\alpha^k + \frac{1}{2} \end{bmatrix} \begin{bmatrix} T_1^k \\ T_2^k \end{bmatrix} = 3\alpha^k \begin{bmatrix} T_w^{k,1} \\ T_w^{k,2} \end{bmatrix} + \begin{bmatrix} T_2^{k-1} \\ 0 \end{bmatrix} \quad , \tag{2.16}$$

where $\alpha^k = \frac{\pi D\gamma \Delta z^k}{6\phi_m c_p}$.

2.1.3. Algorithm overview

The system in Equation 2.16 solves for the temperature coefficients of a single element. To get the complete temperature in the coolant this system is solved for all elements. Due to the streaming in the channels the temperature in an element is dependent on the temperature in the previous element. To account for this, the element matrices are solved in sequential order starting with the element at the top of the core, where the temperature of the previous element is the inlet temperature. The complete U-Battery® has over 3000 coolant channels. Since the influence of the coolant channels on each other will be accounted for through the heat conduction with the graphite, the coolant channels in this code are modeled as isolated systems. The code for creating and solving the element matrices for the coolant channels is first written in Python and later in Fortran90. Figure 2.3 shows the algorithm used for solving the coolant temperature. The output of the code contains the temperature expansion coefficients T_i^k and the temperature at the quadrature points.

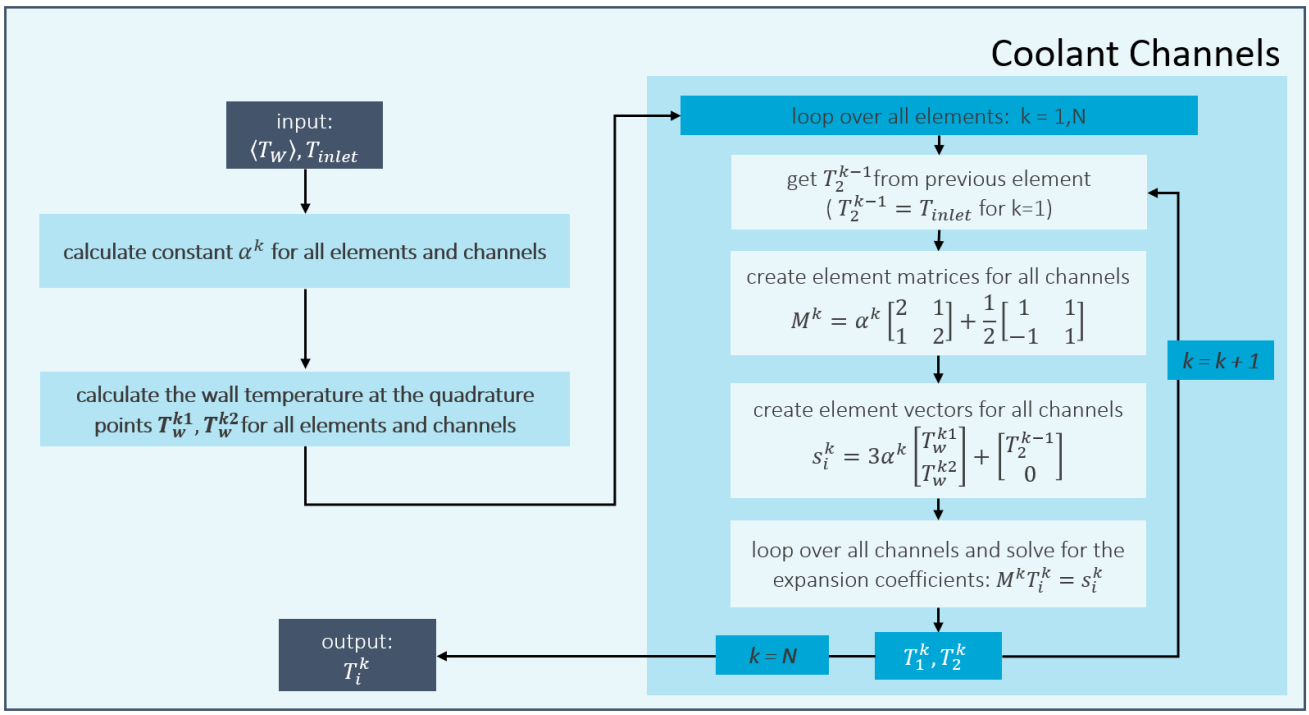


Figure 2.3: Schematic algorithm overview of the code for the coolant channel temperature.

2.2. Model for solid temperature

The second code for the thermal-hydraulic calculations for the U-Battery® consists of a model for the solid temperatures in the core. The core of the reactor consists of prismatic fuel blocks surrounded by a graphite reflector. The fuel blocks are solid graphite blocks with holes for coolant and fuel. The fuel consist of TRISO fuel particles combined in fuel compacts. For simplicity the stacked fuel compacts in the fuel channels are modeled as uniform fuel rods [3], [10]. To explain the code for the solid temperature, the first part of this section discusses the concept of heat diffusion. Next, the symmetric interior penalty Galerkin method for solving the steady-state heat equation is derived. For implementation of this method in a Fortran90 code, parts of the TU Delft in-house code PHANTOM-SN for neutron transport were adapted to be applicable to heat diffusion instead of neutron diffusion. The adjustments of this code and the final algorithm for solving the steady-state solid temperature are given in the third part of this section.

2.2.1. Heat diffusion through solid material

In solids the transfer of heat is called heat conduction or heat diffusion. In the U-Battery® the solid materials are simplified to two types: the fuel inside the fuel channels and the graphite of the moderator and reflector. In these materials the temperature is described by the time-dependent diffusion equation:

$$\rho c_p \frac{\partial T(\vec{r},t)}{\partial t} = \nabla \cdot \lambda(T) \nabla T(\vec{r},t) + p(\vec{r},t) \; . \tag{2.17}$$

This equation describes how the time rate of change of the temperature of a solid with density ρ in [g/cm] and specific heat capacity c_p in [J/g/K] is given by the second spatial derivative of the temperature, the material thermal conductivity $\lambda(\vec{r},T)$ in [W/cm/K] and the heat production $p(\vec{r},t)$ in [W/cm^3] [7], [12].
When considering a single surface with a surface normal vector $\hat{n}$, the thermal energy through

that surface is called the heat flux ϕ_q'' in [W/cm^2]. For heat diffusion problems this heat flux is given by *Fourier's Law* [12] :

$$\phi_q'' = -\lambda(\vec{r}, T)\nabla T(\vec{r}, t) \cdot \hat{n} \, . \tag{2.18}$$

Fourier's law essentially states that thermal energy will flow from places with higher temperatures to places with lower temperature. The thermal conductivity λ indicates how diffusive a material is. In a material with a high λ heat will distribute evenly more quickly than in materials with a lower λ.
The moderator and reflector of the U-Battery® are made of graphite. Graphite is a crystalline structure of carbon atoms. The temperature dependence of the thermal conductivity is caused by vibrations in the crystalline lattice. Since the crystalline structure is different for different types of graphite, the thermal conductivity is dependent on the type of graphite used. In addition, since the temperatures in the U-Battery® have large variations, the variations in the thermal conductivity are large as well. Figure 2.4 gives an example of the variation of the thermal conductivity for G-346 Isotropic graphite [17]. A type of graphite that is tested as moderator in gas-cooled nuclear reactors. For this thesis the thermal conductivity is assumed to have a constant value of 1.33 W/cm/K, identical to the value used in the initial calculations on the U-Battery® by Ding et al. (2011) [3] .

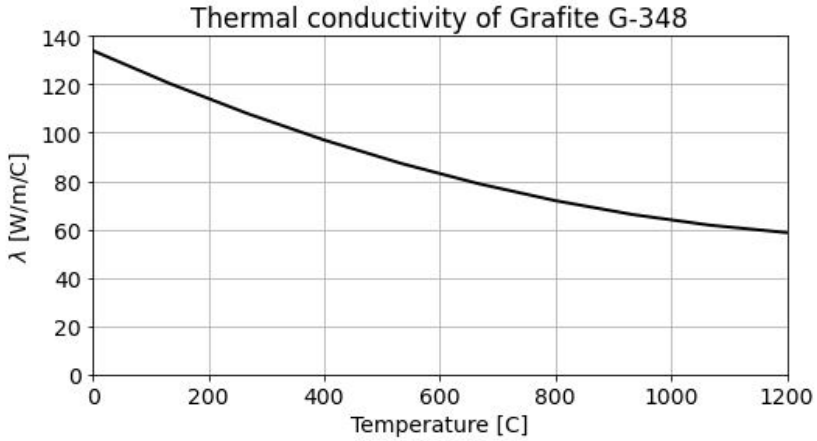


Figure 2.4: Temperature dependency of the thermal conductivity for G-348 isotropic graphite as found by McEligot et al. (2016) [17].

2.2.2. Symmetric Interior Penalty Galerkin Method

To obtain a code for the temperature in the solids of the core the diffusion equation (2.17) needs to be solved. The diffusion equation is a second-order elliptical problem. For higher-order problems the solution of the conventional the Discontinuous Galerkin method tends to become unstable. To ensure the stability of the solution the Symmetric Interior Penalty Galerkin (SIPG) method was introduced. The SIPG method applies a penalty to jumps at interior boundaries. This reduces the sizes of the interior jumps. This chapter focuses on the steady-state solution for the solid temperature. The SIPG method is worked out for the steady-state diffusion problem [16], [18]:

$$\begin{aligned} -\nabla \cdot (\lambda \nabla T) &= p, \quad \text{in } \Omega \\ \lambda \, (\nabla T \cdot \hat{n}) &= g_N, \quad \text{on } \Gamma_N \, . \end{aligned} \tag{2.19}$$

Here the domain Ω is bounded by a single boundary Γ_N. On this boundary Neumann boundary condition g_N applies. The spatial temperature T is the solution of the problem. p is the spatial power production. To solve this system the steps of the DG method are followed once more.

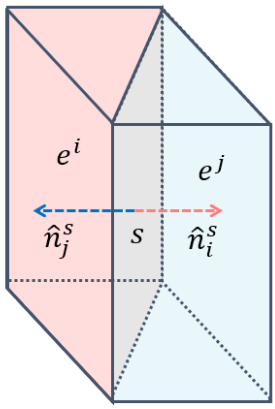


Figure 2.5: Surface boundary s between two three-dimensional elements e^i and e^j with outward normal surface vectors $\hat{n}_i^s$ and $\hat{n}_j^s$.

The domain on which the temperature T is sought and the basis functions are now three-dimensional. Instead of an interval over the line z, this domain is partitioned by assuming a three-dimensional mesh M made up of N elements e^k. The partitioning of the mesh M is explained more elaborate in section 2.2.3. The interior boundary Γ_i is defined as the set of interior edges or faces of M. For two neighboring elements e^i and e^j, sharing the same face s, the outward normal of the surface for each element is given by $\hat{n}_s^i$ and $\hat{n}_s^j$ as is shown in Figure 2.5. If the basis functions on the elements are given by h^i and h^j respectively, the basis function has two traces on the boundary surface s. The average of the basis function on the element e^i, $\{h^i\}$, and the jump $[h^i]$ on the face s are defined as [15], [16], [18]:

$$\{h^i\} = \frac{1}{2}\left(h^i + h^j\right) \qquad [h^i] = (h^i \hat{n}_s^i - h^j \hat{n}_s^j)\,. \tag{2.20}$$

On the boundary surfaces of the domain the elements do not have a neighbor and the definition of the jump and average is given by:

$$\{h^i\} = h^i \qquad [h^i] = h^i \hat{n}_s^i\,. \tag{2.21}$$

Similar to the convectional DG method, the solution T is approximated by u on the domain Ω. Where u is a summation of three-dimensional polynomial basis functions. For the diffusion problem prismatic elements are used with six basis functions on each element k: h_i^k , (with $i = 1, 2, \ldots, 6$). The expansion coefficients for the temperature corresponding to the basis functions are T_j^k. The approximation for the temperature is thus written as:

$$T(r) \approx u(r) = \sum_{k=1}^{K} \sum_{j=1}^{6} T_j^k h_j^k(r)\,. \tag{2.22}$$

The three-dimensional basis functions $h_j^k(r)$ for the prismatic elements are dependent on the configuration of the mesh M on each element. In section 2.2.3 the basis functions and other implications of the prismatic elements for the mesh of the U-Battery® core are explained in more detail. The weak formulation of the diffusion problem is given by the Galerkin equation on the function space V:

$$\text{Find } u \in V \text{ such that } \; a(u, v) = L(v) \;\; \forall v \in V\,. \tag{2.23}$$

In the DG method the bilinear and linear forms were derived directly from the partial differential equation. In the SIPG method the DG bilinear form $a(u, v)$ and linear form $L(v)$ for the diffusion

problem in Equation 2.19 are given by:

$$a(u,v) = \sum_{e\in M}\int_{e_k} \lambda\nabla v\cdot\nabla u + \sum_{s\in\Gamma_l}\int_s (-\{\lambda\nabla v\cdot\hat{n}_s\}[u] - \{\lambda\nabla u\cdot\hat{n}_s\}[v] + \sigma_0[u][v]) \ ,$$
$$L(v) = \sum_{e\in M}\int_{e_k} pv + \sum_{s\in\Gamma_N}\int_s vg_N \ . \tag{2.24}$$

The last term in $a(u,v)$ is called the jump term. This jump term is a bilinear operator introduced to penalize the jump of the functions on the interior element boundaries Γ_l. The constant σ_0 is called the penalty parameter. This parameter determines how much the interior jump is penalized. The value for the parameter penalty can vary for each element. The choice of the penalty parameter will be discussed in section 2.2.4 [16], [18].

Next, the Neumann boundary conditions on Γ_N are implemented in the weak form. In heat diffusion the boundary conditions are given by the gradient of the temperature. Two types of boundary gradients are considered. First, when the gradient is zero, no heat flows in or out of the surface. This can be interpreted as an insulating boundary and will be referred to as Γ_0. On this boundary $g_N = 0$ thus the contribution of this boundary drops out of the weak form. In the other situation, Γ_N, the gradient is given by the inward heat flux. This heat flux is proportional to the difference between the temperatures of the solid on the boundary surface and the temperature at the other side of the surface. For the case of a boundary between the coolant with T_{He} and the graphite the heat flux is given by the same relation as the heat flux in the code for the coolant channels [7]:

$$\phi_q = \gamma(T(r_s) - T_{He}(z)) \ , \tag{2.25}$$

where γ is the heat transfer coefficient for the helium-graphite boundary as shown in section 2.1.1. Working out the sum over the Neumann boundary for this condition gives:

$$\sum_{s\in\Gamma_N}\int_s vg_N = \gamma\int_{de_k\in\Gamma_N}(T_{He}v - uv) \ . \tag{2.26}$$

Implementing the temperature approximation for u and the three-dimensional basis functions for v gives again a solvable system for the temperature in the U-Battery®:

$$L_{ij}T_i = s_i \tag{2.27}$$

The entries for the SIPG matrix L_{ij} and the SIPG source vector s_i are given by:

$$L_{ij} = \sum_k\int_{e_k}\lambda\nabla h_i^k\cdot\nabla h_j^k + \sum_{s_k\in\Gamma_l}\int_{s_k}\left(-\lambda\{\nabla h_i^k\cdot\hat{n}_s\}[h_j^k] - \lambda\{\nabla h_j^k\cdot\hat{n}_s\}[h_i^k] + \sigma_0[h_i^k][h_j^k]\right)\ldots$$
$$\ldots + \gamma\sum_{s_k\in\Gamma_{Ni}}\int_{s_k} h_i^k h_j^k \ , \tag{2.28}$$
$$s_i = \sum_k\int_{e_k} ph_i^k + \gamma\sum_{s_k\in\Gamma_{Ni}}\int_{s_k} T_{He}h_i^k \ .$$

When using 6 basis functions and N elements the SIPG matrix in Equation 2.28 has the dimension (6N) x (6N). This matrix is a block matrix consisting of N 6x6 matrices on each row and column. On the diagonal of the SIPG matrix the contributions of the own basis functions on each element A^k are listed. Additionally, as can be seen in the definitions of the average and jump terms over the basis functions, each element is affected by the basis functions of its neighbors and vice versa. Since prismatic elements have at most five neighbors (elements on the boundary of the domain will have less neighbors), each row and column of the SIPG matrix will have at most $1 + 5 = 6$ smaller 6x6 matrices as entries. If the 6x6 matrix B^{kl} is the influence of element l on the element k, an example of the SIPG matrix L can be visualized as follows:

$$L = \begin{bmatrix} A^1 & B^{12} & B^{13} & & 0 & B^{1N} \\ B^{21} & A^2 & 0 & \ddots & & 0 \\ B^{31} & 0 & A^3 & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & & \ddots & \ddots & A^{N-1} & 0 \\ B^{N1} & 0 & & 0 & 0 & A^N \end{bmatrix},$$

Zero entries correspond to elements that are not neighbors of each other and are thus empty. The maximum number of non-zero entries for the SIPG matrix for a prismatic mesh of N elements is: $N \cdot (6^2 + 5 \cdot 6^2) = N6^3 = 216N$ [16], [18].

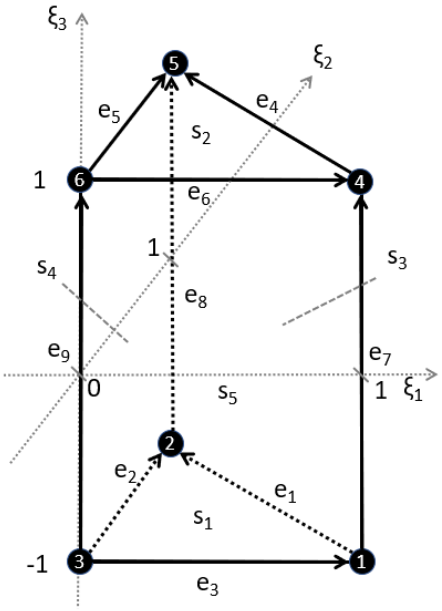


Figure 2.6: Reference prism for a single element showing the vertices, edges e and surfaces s.

2.2.3. Prismatic elements for domain partitioning

As was shown previously, in finite element methods the domain is partitioned in a mesh. Basis functions on this mesh need to be defined. For one-dimensional problems this partitioning was quite straightforward. However, the temperature of the solids is solved by a three-dimensional diffusion problem, so a three-dimensional partitioning of the U-Battery® is needed. Because of the geometry of the U-Battery® it was chosen to work with a triangular prismatic mesh: prisms are triangular in the x and y- direction, which makes meshing circular geometries easier compared to for instance square elements. Prismatic elements are linear in the z-direction, which

allows for easier coupling of the one-dimensional coolant temperature solution to the solid temperature calculations.

For creating the solid temperature code parts of the PHANTOM-SN code for neutron transport were adapted. This finite element code is already able to work with three-dimensional hexahedrons and tetrahedrons but not yet triangular prisms. Therefore a new element type was added to the shape function library of the PHANTOM-SN code. This adjustment has a few implications: First, the master element for the mesh and the basis functions on this master element have to be defined. Next, since the code needs to solve volume integrals over the master element, quadrature sets for the prismatic element are added to the *quadrature* module. At last the the penalty parameter σ_0 for the jump terms of the SIPG method are defined for the specific case of the prismatic elements.
When meshing a geometry, the master element is transposed and transformed to fit the desired shape. For calculations, each element is mapped to the master element. The master element for the triangular prism K_P is defined on the coordinates ζ_1, ζ_2 and ζ_3. Since the PHANTOM-SN code already includes two-dimensional triangle elements K_t, the master prism is defined as the outer product of the triangle element in the ζ_1 and ζ_2 directions and the one-dimensional line element K_l in the ζ_3 direction: $K_P = K_t \otimes K_l$. The domain is $[0 \leq \zeta_1, \zeta_2 \leq 1]$, $[\zeta_1 + \zeta_2 = 1]$ and $[-1 \leq \zeta_3 \leq 1]$. The reference structure is shown in Figure 2.6 and consists of: 6 nodes, v_1, v_2,...,v_6, 9 edges: e_1,e_2,...,e_9 and 5 surface: s_1,s_2,...,s_5 [15], [19].
The six nodal shape functions h_{v_i} corresponding to each node v_i are found by multiplying the affine coordinates of the two-dimensional triangle λ_1^t, λ_2^t and λ_3^t in ζ_1 and ζ_2 with the affine coordinates for the line element, λ_4^l and λ_5^l, in ζ_3:

$$
\begin{aligned}
h_{v1} &= \lambda_4^l \lambda_1^t = \frac{1}{2}(1-\zeta_3)\,\zeta_1 \,, \\
h_{v2} &= \lambda_4^l \lambda_2^t = \frac{1}{2}(1-\zeta_3)\,\zeta_2 \,, \\
h_{v3} &= \lambda_4^l \lambda_3^t = \frac{1}{2}(1-\zeta_3)\,(1-\zeta_1-\zeta_2) \,, \\
h_{v4} &= \lambda_5^l \lambda_1^t = \frac{1}{2}(1+\zeta_3)\,\zeta_1 \,, \\
h_{v5} &= \lambda_5^l \lambda_2^t = \frac{1}{2}(1+\zeta_3)\,\zeta_2 \,, \\
h_{v6} &= \lambda_5^l \lambda_3^t = \frac{1}{2}(1+\zeta_3)\,(1-\zeta_1-\zeta_2) \,.
\end{aligned}
\tag{2.29}
$$

For the calculations in this thesis linear basis functions are used. It can be shown that each nodal shape functions h_{vi} has the value of 1 in its corresponding node and a value of 0 in the other nodes. Additionally, the sum of all the basis functions always adds up to 1. When the shape functions are integrated over the reference prism their sum is one as well. The basis functions h_i^k for the mesh of the core are given by the six nodal shape functions transformed to fit on the element k [15], [19].
In the PHANTOM-SN code the volume and face integrals over the elements are solved numerically using Gauss quadrature. This technique approximates an integral over a volume as the weighted sum of the function in non-evenly spaced quadrature points. To solve the integrals on the prismatic elements, Gauss quadrature is needed for three-dimensional prisms. Again, the combination of the quadrature for the triangle and line element is used to calculate the quadrature points and weights for the master triangular prism. On the one dimensional line element K_l the integration on of a function $g(\zeta_3)$ with weights $w_{l,j}$ and M quadrature points ζ_3^j

Table 2.3: First and second-order quadrature sets for the one-dimensional line, two-dimensional triangle and three dimensional triangular prismatic element.

Line			
order	M	points	weights
1	1	0	2
2	2	$-\sqrt{\frac{1}{3}}\ \sqrt{\frac{1}{3}}$	1 1
Triangle			
order	N	points	weights
1	1	$\left(\frac{1}{3},\frac{1}{3}\right)$	$\frac{1}{2}$
2	3	$\left(\frac{1}{6},\frac{1}{6}\right)\left(\frac{1}{6},\frac{2}{3}\right)\left(\frac{2}{3},\frac{2}{3}\right)$	$\frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}$
Prism			
order	N*M	points	weights
1	1	$\left(\frac{1}{3},\frac{1}{3},0\right)$	1
2	6	$\left(\frac{1}{6},\frac{1}{6},-\sqrt{\frac{1}{3}}\right)\left(\frac{1}{6},\frac{2}{3},-\sqrt{\frac{1}{3}}\right)\left(\frac{2}{3},\frac{2}{3},-\sqrt{\frac{1}{3}}\right)\left(\frac{1}{6},\frac{1}{6},\sqrt{\frac{1}{3}}\right)\left(\frac{1}{6},\frac{2}{3},\sqrt{\frac{1}{3}}\right)\left(\frac{2}{3},\frac{2}{3},\sqrt{\frac{1}{3}}\right)$	$\frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}\ \frac{1}{6}$

is given by:

$$\int_{K_l} g(\zeta_3)d(\zeta_3) = \sum_{j=0}^{N} w_{l,j} f(\zeta_3^j)\ . \tag{2.30}$$

On the triangle K_t the integration on of a function $f(\zeta_1,\zeta_2)$ over N quadrature points is given by the weights $w_{t,i}$ and quadrature points ζ_1^i and ζ_2^i:

$$\int^{K_t} f(\zeta_1,\zeta_2)d(\zeta_1)d(\zeta_2) = \sum_{j=0}^{N} w_{t,j} f(\zeta_1^i,\zeta_2^i)\ . \tag{2.31}$$

Combining these gives the volume integral over the prismatic element K_P for a certain function $h(\zeta_1,\zeta_2,\zeta_3)$:

$$\int_{K_P} h(\zeta_1,\zeta_2,\zeta_3)d\zeta_1 d\zeta_2 d\zeta_3 = \sum_{i=1}^{N}\sum_{j=1}^{M} w_{t,i} w_{l,j} h(\zeta_1^i,\zeta_2^i,\zeta_3^j)\ . \tag{2.32}$$

For the basis functions of the prismatic elements the maximum order is 1. Thus second-order Gauss quadrature is sufficient to calculate the volume integrals. The first and second-order quadrature points and weights for the prismatic element are given in Table 2.3. The sum of all quadrature weights equals the volume of the element. Since the volume of the reference prism is one, the sum of the quadrature weights should add up to one as well [15].

2.2.4. SIPG Penalty parameter

The penalty parameters σ_0 for diffusion problems is dependent on the geometry of the element and the diffusion coefficient λ. The PHANTOM-SN code included calculations for the penalty parameters for different element types. Here σ_0 is given by the diffusion length scale divided by the length scale of the element multiplied by a factor dependent on the geometry of the element's surface. The jump parameter is calculated over the interior boundaries of the elements. Since prismatic elements consist of two boundary area types, triangles and quads, this factor is different on the different boundary surfaces. The penalty parameter on each surface

is calculated according to:

$$\sigma_0^{triangle} = \frac{max\,(\lambda)}{L_{elem}}\,(p+1)^2\ , \tag{2.33}$$

$$\sigma_0^{quad} = \frac{max\,(\lambda)}{L_{elem}}\frac{1}{2}\,(p+1)\,(p+2)\ , \tag{2.34}$$

where p is the polynomial order and L_{elem} is the volume of the element divided by the surface area. The diffusion length scale is taken as the maximum value of the thermal conductivity λ of the element and the neighboring element [20].

2.2.5. Algorithm overview

To calculate the steady-state temperature in the solid materials the steady-state diffusion equation in section 2.2.1 is solved on a prismatic mesh using the SIPG method introduced in section 2.2.2. The code used to solve this system is an adaptation of the TU Delft in-house code PHANTOM-SN. This is a Fortran90 code for solving the neutron transport equation. One of the solvers in this code uses the SIPG method to solve the diffusion approximation for the neutron transport equation. This Diffusion Synthetic Acceleration (DSA) solver is copied alongside the necessary modules and adjusted to work with the prismatic elements as well as heat transport instead of neutron transport. As input the code needs the heat transfer coefficient, γ, the temperatures of the coolant channels and the power density in the solid. The temperature on the coolant boundaries is given as a function depending on z-coordinate T_{He}. The input for the power density is a function dependent on the spatial coordinates, $P(r)$. The core geometry is modelled using two solid materials: fuel and graphite, the conductivity, λ, of these materials is inputted as well.

After reading the input parameters, the code computes the SIPG matrix and source vector in Equation 2.28. The volume and surface integrals in the entries of the SIPG matrix and vector are all solved using second order Gauss quadrature. Next, this matrix system is solved for the temperature of the solids. Figure 2.7 gives a schematic overview of the code for solving the steady-state temperature.
The SIPG matrix has dimensions $(6N)$x$(6N)$, where N is the number of elements. For solving this matrix system the PETSc Krylov solver is used with a preconditioned conjugate gradient method for positive definite symmetric matrices [21]. This is an iterative solver that converges to a solution for the temperature. The tolerance of the solver can be adjusted and affects the accuracy of the solution.
In this code the temperature calculation for the coolant is not yet included. The next chapter will focus on the coupling of the separate steady-state codes for the coolant and solid temperatures.

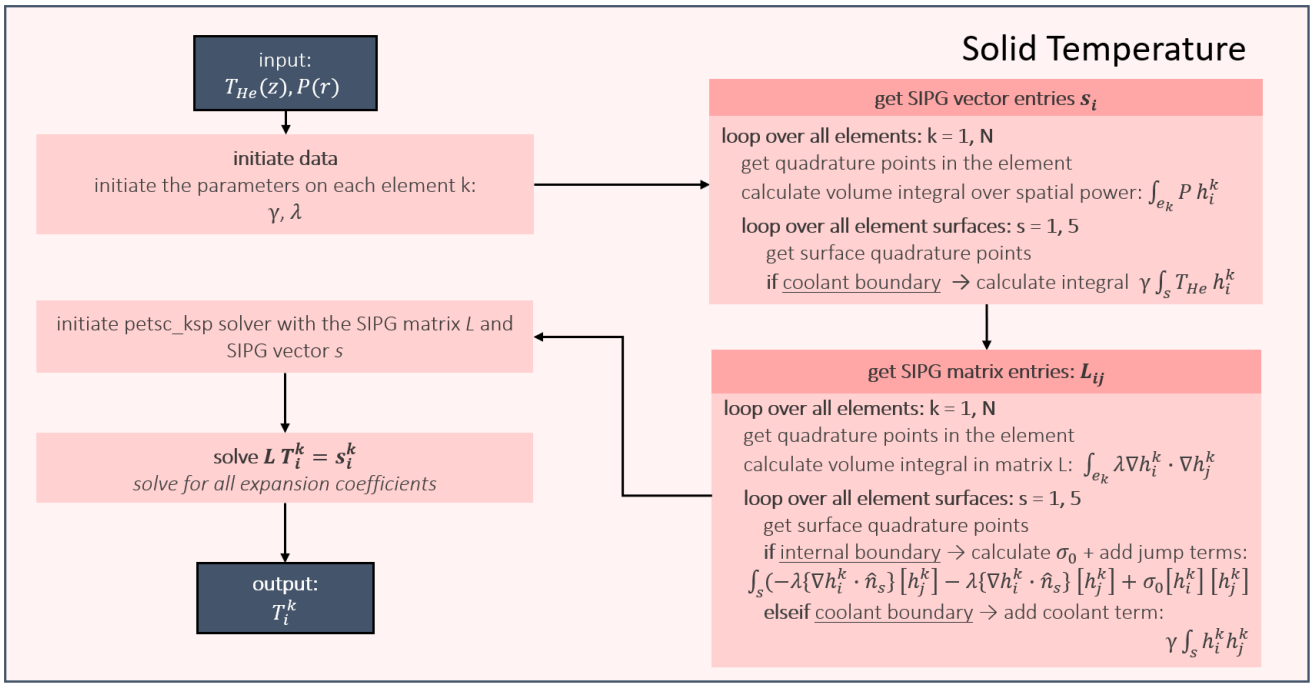


Figure 2.7: Schematic algorithm overview of the solid temperature module.

3

Coupled time-dependent thermal-hydraulics model

In the previous chapter two separate finite element models for steady-state temperature calculations in the coolant and solids of the U-Battery® core were explained. This chapter discusses the coupling of these codes into a complete thermal-hydraulic time-dependent model of the core by using the output of each code as input for the other. First, the creation of the one-dimensional mesh for the coolant channels from the three-dimensional mesh of the solids is explained. This is followed by an iterative model for steady-state calculations. Next, the Crank-Nicolson method is used to solve for the time-dependent core temperature. The last section of this chapter gives an overview of the model used for solving time-dependent problems for transient calculations of the core temperature.

3.1. Mapping between the models

The steady-state temperature models for the coolant and solid are dependent on each other. The codes intertwine at the coolant channel boundaries: The coolant model uses the temperature of the graphite at the channel walls as input and the solid model uses the coolant temperature in the channel as input. To be able to use the output of these codes as input for the other code, first a mapping between the geometries of the meshes is made. This is done by choosing the one-dimensional domain for the coolant channels according to the mesh of the solids. The mesh of the solids is a three-dimensional mesh consisting of linear prismatic elements. In this mesh coolant channels are left empty and the each cylindrical boundary surface with a coolant channel is labeled with the number of the coolant channel. As shown in Chapter 2.2.3 the prisms correspond to one-dimensional line elements in the z-direction. This property is used to generate linear elements for the coolant channels based on the prismatic mesh of the solids. In the Fortran90 code this is done as follows: First, each element in the three-dimensional solid mesh gets a new property corresponding to the height level of the element. Since the z-direction in the coolant model is oriented downwards, this is done from top to bottom, labeling the top layer as level 1. Next, the layers and their sizes are used as the one-dimensional domain for the coolant channel calculations. This means that the number of elements in the model for the coolant channels corresponds to the number of elements in the z-direction in the solid mesh. A schematic overview of how this is done is shown in Figure 3.1 for three elements. Figure 3.2 shows a cut in the z-plane of the three-dimensional mesh around a coolant channel.

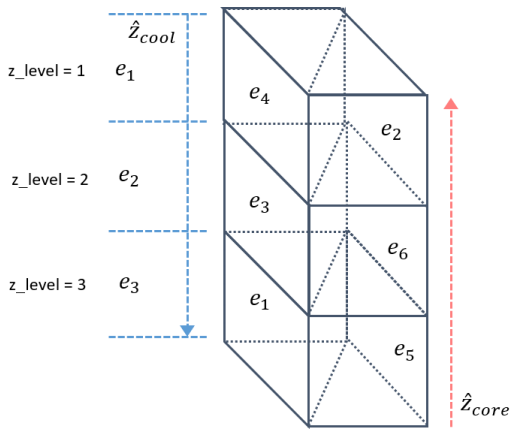


Figure 3.1: Schematic overview of the creation of the coolant channel mesh according to a prismatic mesh for the core. The elements e_k of the core mesh are number randomly.

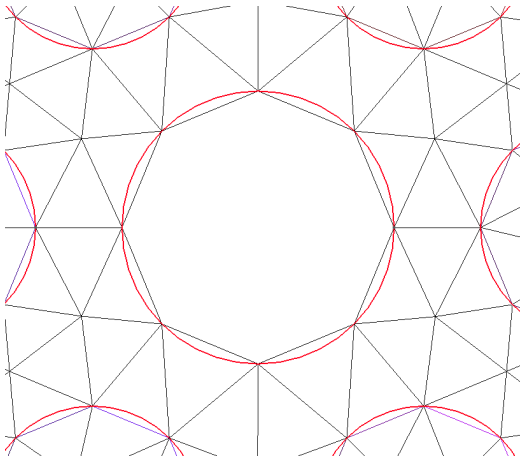


Figure 3.2: Cross-section of the mesh around a single coolant channel. The black lines show the mesh as created by GMSH [22]. The red lines are the channel boundaries in the actual U-Battery® geometry.

Now that the meshes for the coolant and solid models are conformal, the output of each code is used as input at the channel boundaries of the other code. In both codes the contribution of the other code is added to the source vector at the right-hand side of the DG and SIPG matrix systems. As was shown in section 2.1.2, the contribution of the solid temperature to the coolant model is given by:

$$\pi D \gamma \int_{e_k} \langle T_w \rangle \, h_i^k \, dz \,, \tag{3.1}$$

where $\langle T_w \rangle$ is the averaged solid temperature over the circumference of the coolant channel wall. This integral is calculated using second-order Gauss quadrature. So the averaged wall temperature at the quadrature points is needed as input. The solid model uses second-order quadrature as well. All coolant channel boundaries in the solid mesh are labelled with the number of the corresponding coolant channel. For each boundary surface in the solid mesh, the temperature at the quadrature points can be found. Figure 3.3 shows the mapping of the quadrature points on the two-dimensional boundary surfaces to quadrature points on the one-dimensional line elements of the coolant model. As is shown in Figure 3.2, each coolant channel is mapped by multiple boundary surfaces. The input $\langle T_w \rangle$ at each quadrature point is calculated by taking the average of all corresponding quadrature points in the solid mesh. This is done using the algorithm in Figure 3.4a.
In the model for the solid temperature, the coolant temperature contributes to the source vector as well. As was shown in section 2.2.2, the coolant temperature is included taking the integral over the coolant channel surface according to:

$$\gamma \int_{s_k \in \Gamma_N} T_{He} h_i^k \, dA \,, \tag{3.2}$$

where T_{He} is the coolant temperature dependent on the height. Again this integral is solved using second-order Gauss quadrature, so the coolant temperature at the quadrature points of the boundary surfaces is needed. Again the mapping in Figure 3.3 is used. The coolant code gives as output the temperature at the quadrature points of the coolant elements for all channels. The algorithm in Figure 3.4b shows how this output is used to compute the source vector for the SIPG matrix system.

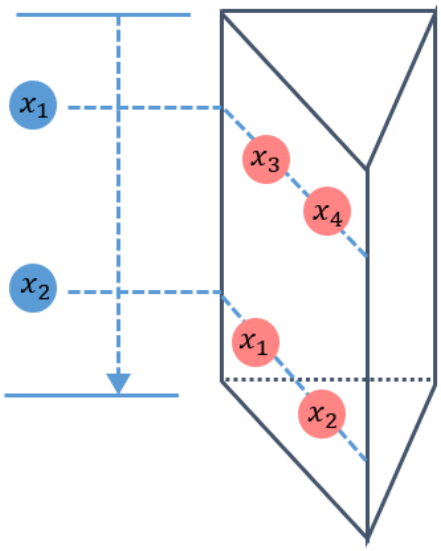


Figure 3.3: Mapping of the quadrature points on the two-dimensional boundary surfaces of the prismatic mesh (red dots) to the one-dimensional coolant mesh (blue dots).

```
Solid temperature → average wall temperature:

loop over all core elements k = 1,N
 get element height level = z
  loop over all surfaces s = 1, 5
   if surface  = coolant boundary
    get channel_no = c
    loop over all surface quad points x = 1,4
     get solid temperature at quad point x = T_solid
     if (x=3) or (x=4)
       → temp_wall (c, z, 1)  =+ T_solid
       → cnt(c, z, 1) =+ 1
     elseif (x=1) or (x=2)
       → temp_wall (c, z, 2)  =+ T_solid
       → cnt(c, z, 2) =+ 1

temp_wall (c, z, 1) = temp_wall (c, z, 1) / cnt (c, z, 1)
temp_wall (c, z, 2) = temp_wall (c, z, 2) / cnt (c, z, 2)
```

(a) Solid temperature as input for the coolant code. The averaged wall temperature is calculated using the solid temperature at the quadrature points of the boundary surface.

```
Coolant temperature → source vector in solid model:

loop over all core elements k = 1,N
 get element height level = z
  loop over all surfaces s = 1, 5
   if surface  = coolant boundary
    get channel_no = c
    loop over all surface quad points x = 1,4
     get quad weight*quad det = quot
     get element dg shape functions at x = fun_dg
     if (x=3) or (x=4)
       → T_He = coolant temp at quad point 1 (z, c, 1)
     elseif (x=1) or (x=2)
       → T_He = coolant temp at quad point 2 (z, c, 2)
     loop over all nodes of the element:  node = 1,6
       get position of node in SIPG matrix = id
       src(id) = src(id) + quot*fun_dg*gamma*T_He
```

(b) Coolant temperature as input for the solid code. The source integral over the boundary, src, is calculated. Gamma is the heat transfer coefficient.

Figure 3.4: Algorithms for the coupling of the solid and coolant temperature.

3.2. Iterative model for steady-state calculations

Using the mapping between the two codes, the steady-state temperature of the coupled system is calculated by looping over multiple iterations between the coolant and solid. An initial temperature for both coolant and solids is given as input. At the first iteration the coolant temperature is calculated from the initial solid temperature, after which the solid temperature is calculated using the updated coolant temperature. This process is repeated for a given number of iterations or until the temperature difference between two iterations is below a given tolerance. This model is used to calculate steady-state solutions for constant power density and parameters. For transient analysis a time-dependent model of the U-Battery® is needed.

3.3. Crank-Nicolson method for time dependence

To compute temperature variations over time for changing boundary conditions or varying power densities, the combined model should be made time-dependent. The previous codes dealt with solutions to the steady-state diffusion equation. To include time-dependence in the model the Crank-Nicolson method is used to solve the time-dependent diffusion equation. The Crank-Nicolson (CN) method is a finite difference scheme for time integration. For diffusion problems CN is unconditionally stable and second-order accurate [23]. The solutions of the codes for the steady-state coolant and solid calculations can be used to iterate over discrete time steps. This section will discuss the CN method and the derivation of the new matrix system for the temperature as well as the time-scale considered for the U-Battery®.
As was shown in section 2.2.1, the time-dependent diffusion problem that needs to be solved for the solid temperature is described by [7], [15]:

$$\begin{aligned} \rho c_p \frac{\partial T}{\partial t} &= \lambda \nabla^2 T(r,t) + p(r,t)\ , \\ -\lambda(\nabla T \cdot \hat{n}) &= g_N(r,t), \ \text{ on } \Gamma_N\ , \\ T(r,t) &= T_0(r), \ \text{ at } t = 0\ . \end{aligned} \tag{3.3}$$

Here $g_N(r,t)$ is the Neumann boundary condition on the boundary surfaces and $T_0(r)$ is the initial solid temperature at $t = 0$. For the steady-state model for the solid temperature in section 2.2 a matrix system was derived using the SIPG method. Here the space-discretization of the temperature was given by T_i. The temperature is solved using the steady-state diffusion SIPG matrix L and the source vector s_i: $LT_i = s_i$.
The time-dependent diffusion problem in Equation 3.3 is identical to the steady-state diffusion problem with the addition of a linear term for the time-derivative of the temperature. It can be shown that the new matrix system for the time-dependent temperature is solved by [15]:

$$\rho c_p M \frac{\partial T_i}{\partial t} = -LT_i + s_i\ . \tag{3.4}$$

The derivation for this equation is given in Appendix A.1. The matrix M is the finite element mass matrix, whose coefficients are given by: $M_{ij} = \int_\Omega h_i^k h_j^k$ [15].

For the discretization of the time, the implicit CN finite difference method is used. The time is discretized in time steps n with time step size Δt. For equation 3.4 the implementation of the time discretization leads to [15]:

$$\rho c_p\ M \frac{T_i^{n+1} - T_i^n}{\Delta t} = \theta\left(-LT_i^{n+1} + s_i^{n+1}\right) + (\theta - 1)\left(-LT_i^n + s_i^n\right)\ , \tag{3.5}$$

which is the Theta method for time integration. In the case of CN it is used that $\theta = \frac{1}{2}$. In this problem the power density and boundary conditions are known, thus the source vector s_i^n is known at all time-steps. Using that $\theta = \frac{1}{2}$ a new matrix system can be written down that solves for the temperature at the next time-step T_i^{n+1}:

$$\left(\frac{\rho c_p}{\Delta t} M + \frac{1}{2} L\right) T_i^{n+1} = \left(\frac{\rho c_p}{\Delta t} M - \frac{1}{2} L\right) T_i^n + \frac{1}{2}\left(s_i^{n+1} + s_i^n\right)\ . \tag{3.6}$$

Using the same PETSc solver as in section 2.2.5 the temperature at T^{n+1} is solved from the matrix on the left-hand side and the terms on the right-hand side.

The CN scheme is second-order accurate in time. However, for larger time steps the solution of the CN method tends to oscillate. Generally, the time steps Δt for a system with spacial steps Δx and thermal diffusivity $\alpha = \frac{\lambda}{\rho c_p}$ should satisfy [23]:

$$\alpha \frac{\Delta t}{\Delta x^2} < \frac{1}{2}\,, \tag{3.7}$$

to avoid oscillations in the solution. The material parameters of graphite in the U-Battery® lead to a diffusivity of $\alpha = \frac{\lambda}{\rho c_p} = \frac{1.33}{1.8 \cdot 0.72} \approx 1$. The distance between the coolant and fuel channels in the U-Battery® is around 0.5cm, the element sizes in the model will be between 0.5cm and 1 cm in the radial directions. Using this it can be calculated that the maximum time step size before oscillations occur will be around $\frac{0.5^2}{2} = 0.125$.

3.4. Algorithm overview

The matrix system in Equation 3.6 solves for the time-dependent solid temperature and is implemented in the code for the core temperature. The time is integrated using equally spaced time steps n of size Δt. A benefit of the CN system is that the matrices M and L only have to be computed once. For the initial temperatures of the solid and coolant the steady-state solution from the previous section can be used as input. At each iteration the temperature of the system at the next time-step $n+1$ is solved from the temperature at n and the source vectors at n and $n+1$. The source vector s_i^n is dependent on the (time-dependent) power density over the core and on the coolant temperature at the time step T_{He}^n. Here, for the calculation of s^{n+1}, a problem arises: to calculate T_{solid}^{n+1}, T_{He}^{n+1} is needed. But for calculating T_{He}^{n+1}, T_{solid}^{n+1} is needed. To solve this problem the helium temperature at $n+1$ is first calculated by extrapolating the temperature of the solid. This approximation for the solid temperature at $n+1$ is made using the extrapolation of the temperature, T_{solid}^{ext}, over n and $n-1$ according to:

$$T_{solid}^{n+1} \approx T_{solid}^{ext} = T_{solid}^{n} + \left(T_{solid}^{n} - T_{solid}^{n-1}\right) = 2T_{solid}^{n} - T_{solid}^{n-1}\,. \tag{3.8}$$

For the first time-step the initial temperature T_0 is taken as the temperature at $n-1$. Figure 3.5 gives an overview of the time-depended scheme implemented in the code.

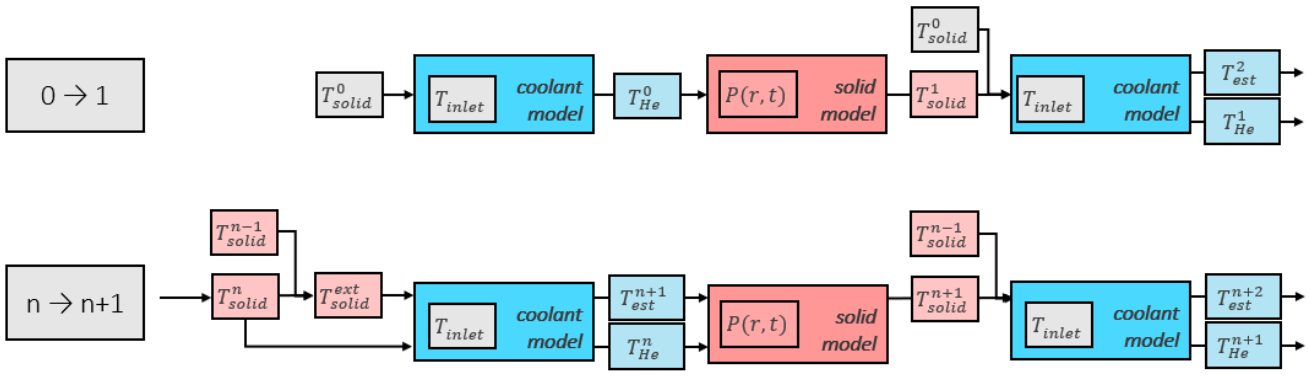


Figure 3.5: Schematic overview of the implementation of the CN algorithm for time integration for the first time step and time step n to $n+1$.

4

Model Verification

This chapter describes the verification of the two steady-state codes and the coupled thermal-hydraulic code followed by a short discussion at the end of each section. The codes are verified separately beginning with the one-dimensional DG method for the coolant temperature. This code is analysed using a python script to implement the method and to compare the results to an analytical solution. The second verification is that of the implementation of the prismatic elements in the PHANTOM-SN code for diffusion problems. Since the prismatic elements were implemented before adjusting the code from neutron transport to heat diffusion, the verification is done on a manufactured neutron transport problem on a prismatic mesh. Next, the coupling of the codes for the coolant and solid in steady-state is analysed for a simplified problem on a conductive tube with a single coolant channel on the inside. Finally, the time dependence of the model is investigated by modeling the same tube with a time-dependent problem.

4.1. Coolant temperature

To verify the one-dimensional DG method for the coolant temperature the results from the code described in section 2.1 are compared to an analytical solution for the steady-state convection problem. In addition, the convergence of the error is checked as well as the conservation of energy. For the analytic solution a single coolant channel is considered with a constant temperature at the walls, $T_w(z) = T_w$, over the entire height H of the core. The temperature profile of the coolant is then given by [12] :

$$T_{\text{analytical}}(z) = T_w - (T_w - T_{\text{in}})\, e^{\frac{-\pi D \gamma (H-z)}{c_p \phi m}}, \tag{4.1}$$

The parameters for and their meaning are shown in Table 4.1. The results of the code for a single coolant channel with a constant wall temperature of T_w = 1000 K are shown in Figure 4.1 for 4 and 10 elements as well as the analytical solution. For the solution with four elements the discontinuous jumps in the DG method are visible. For higher numbers of elements the solution appears to follow the analytical solution.

To test the deviation from the analytical solution as a function of element size the convergence of the normalized root mean square (NRMS) error is plotted in Figure 4.2. The NRMS error is given by:

$$E_{NRMS} = \sqrt{\frac{\int (T_{\text{analytical}} - T)^2 dz}{\int T^2_{\text{analytical}}\, dz}}. \tag{4.2}$$

Table 4.1: Parameters and their value for simulation of the coolant channels.

parameter	value	units	meaning
D	1.588	cm	channel diameter
H	320	cm	height
c_p	5.195	J/g/K	heat capacity
ϕ_m	2.36	g/s	mass flux
γ	0.03	W/cm^2/K	heat transfer coefficient
T_{in}	250	K	inlet temperature
T_w	1000	K	wall temperature

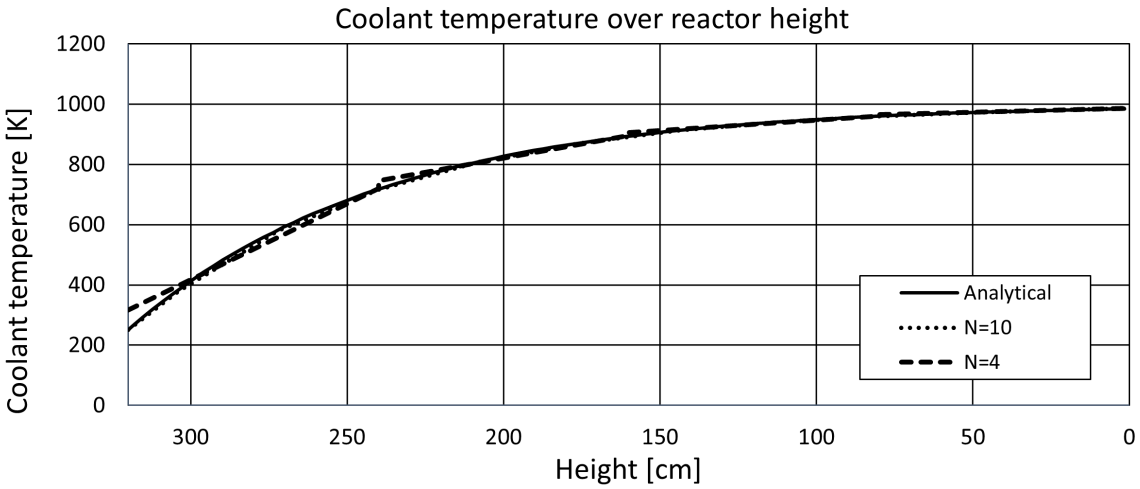


Figure 4.1: Comparison of the coolant temperature with $N = 4$ and $N = 10$ elements against the analytical solution with a constant wall temperature of 1000 K.

Since the basis functions are linear, the convergence should theoretically satisfy a power law of order two. As can be seen in Figure 4.2, the fit for the convergence follows a power law of order 2.00. Figure 4.2 shows that for more than 10 elements the NRMS error is below 10^{-5}.

Finally, the conservation of energy over the height of the coolant channel is checked. For the energy in the system to be conserved, the difference in in and outlet temperature of the helium should be proportional to the integral over the boundary using the DG solution for the temperature. The conservation of energy is calculated by:

$$T_{\text{out}} - T_{\text{in}} = \alpha \int_0^H \big(T_g - T(z)\big)\, dz\,, \tag{4.3}$$

with $\alpha = \frac{\pi D \gamma}{\phi_m c_p}$. To satisfy conservation of energy, the difference between the left and right hand side, ΔE is calculated. The values of ΔE for different element sizes are given in table 4.2. It can be seen that even for a smaller numbers of elements the error is of the order 10^{-13}, the computational precision.

The computation times of the simulations are less than 1 second for calculations of up to 1000 elements. In these results the temperature profile for a single coolant channel is given. However, the code is able to calculate multiple coolant channels simultaneously. The U-Battery® has a total of 3996 coolant channels. Calculating 3996 coolant channels with 40 elements each took the Python code 31 seconds.

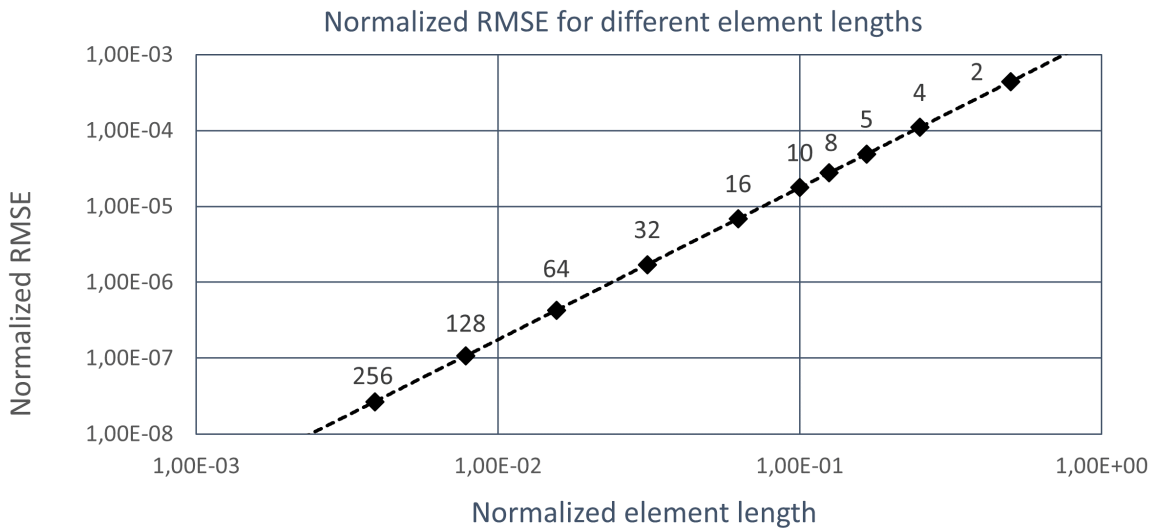


Figure 4.2: Normalized root-mean-square error for the coolant channel temperature calculation using the DG method for different numbers of elements and element lengths normalized to the height H. The dotted trend line is the function $y = ax^b$ with b = 2.00.

Table 4.2: Difference in conservation of energy for different numbers of elements.

N	**10**	**16**	**32**	**320**
$\lvert\Delta E\rvert$	1.1 10^{-13}	1.0 10^{-13}	4.5 10^{-13}	1.8 10^{-13}

Discussion

The results for the coolant channel model follow the analytical solution and show a theoretical convergence to second-order accuracy for the linear basis functions in one dimension as well as conservation of energy over the solution. The implemented DG code can calculate the steady-state temperature profiles for all coolant channels of the U-Battery® in less than 40 seconds when provided with a temperature profile for the graphite at the walls. The tests were run locally using Python. The expected computation time after implementation in Fortran90 and running the on the cluster is negligible compared to the more complex diffusion calculations for the solids.

Table 4.3: Root-mean-square error for the outputted flux with the manufactured source term for different numbers of elements and effective element lengths.

N	**168**	**340**	**564**	**826**	**1328**	**2100**	**3744**	**9420**	**22640**	**75480**
L_{eff}	0.182	0.143	0.121	0.106	0.091	0.078	0.064	0.047	0.035	0.024
$RMSE$	0.0079	0.0049	0.0035	0.0027	0.0021	0.0014	0.0010	$5.5{\cdot}10^{-4}$	$3.1{\cdot}10^{-4}$	$1.4{\cdot}10^{-4}$

4.2. Implementation of prismatic elements

To verify the implementation of the prismatic elements in the PHANTOM-SN code the method of manufactured solutions is used to calculate the neutron flux density for a neutron diffusion problem. In this method the source term is calculated for a known flux profile. This source can be implemented in the code. The error of the code is then given as the normalized root-mean-square difference between the analytical flux profile and the flux profile as outputted by the code. The neutron diffusion equation for a non-fissionable and non-absorbing medium is given by [24]:

$$S\,(x,y,z) = -D\nabla^2\phi\,(x,y,z)\ . \tag{4.4}$$

Here S is the source function, D is the neutron diffusion coefficient and ϕ is the scalar neutron flux density. The diffusion equation is solved on a three-dimensional box structure on the domain $[-a \leq x,y,z \leq a]$. As boundary conditions the vacuum boundary conditions are used such that the inward flux $J^- = 0$ [24]:

$$J^-(\vec{r}_s) = 0 = \frac{1}{4}\phi(\vec{r}_s) + \frac{D}{2}\nabla\phi(\vec{r}_s)\cdot\hat{n}_s\ , \tag{4.5}$$

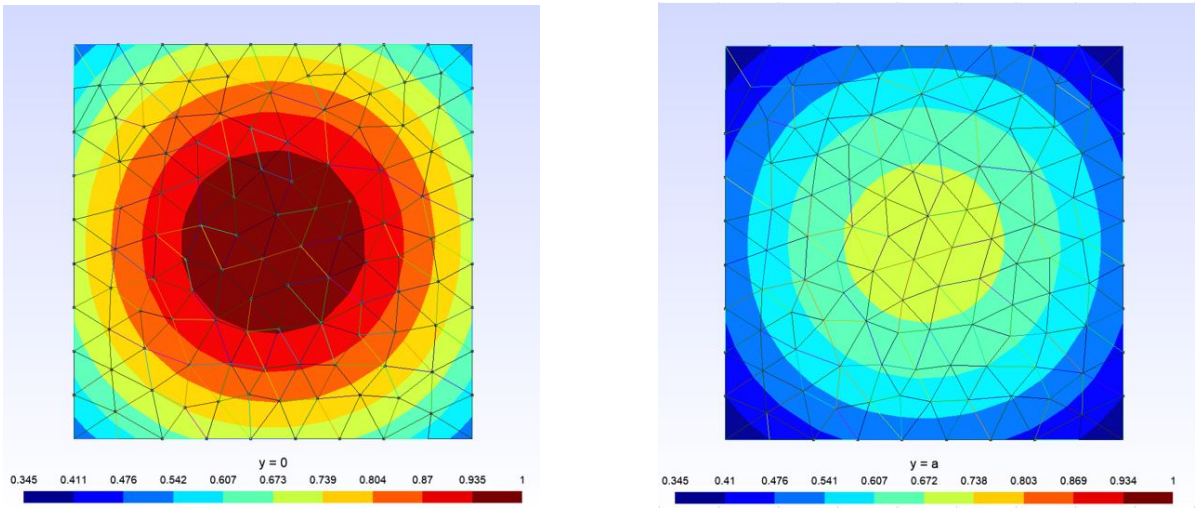
where $\vec{r}_s$ is the boundary of the box defined by $|x| = a$, $|y| = a$, or $|z| = a$ and $\hat{n}_s$ is the outward normal vector of the boundary surface. When the diffusion coefficient is taken to be unity and the dimensions of the box are $a = \frac{\pi}{2}$, it can be shown that the problem is analytically solved by the flux profile ϕ_M according to:

$$\phi_M = \cos\left(\frac{x}{2}\right)\cos\left(\frac{y}{2}\right)\cos\left(\frac{z}{2}\right)\ . \tag{4.6}$$

The manufactured source function for this problem is derived by filling in the diffusion equation and is given by:

$$S\,(x,y,z) = \frac{3}{4}\cos\left(\frac{x}{2}\right)\cos\left(\frac{y}{2}\right)\cos\left(\frac{z}{2}\right)\ . \tag{4.7}$$

This manufactured source is implemented in the Fortran90 PHANTOM-SN code and the flux profile is first solved using the SIPG method on a prismatic mesh as described in section 2.2.3 consisting of 2100 elements. Figure 4.3 shows two two-dimensional profiles of the solution on the planes $y = 0$ (through the center of the box) and $y = a$ (the boundary surface of the box). Next, the root-mean-squared error with respect to the manufactured solution is calculated using a meshes with increasing numbers of elements. To confirm that the error follows the theoretical quadratic convergence for the linear basis functions the effective element size was calculated as $L_{eff} = \sqrt[3]{V/N}$. Where $V = 8a^2$ is the volume of the box and N is the total number of elements in the mesh. The results are listed in table 4.3. These results are plotted on a double logarithmic scale in Figure 4.4. The trend line through the data points is a fitted power series of the shape $y = ax^b$ with $b = 1.98$.



(a) y = 0 (b) y = a

Figure 4.3: Solution for the neutron flux on a prismatic mesh as seen from the center plain and the boundary surface of the box. The images where made in GMSH [22].

Discussion

First-order prismatic elements were implemented in the existing PHANTOM-SN code for neutron transport. The results of the SIPG method for a neutron diffusion problem on a box follow the analytical solution using a prismatic mesh. The NRMS error in Figure 4.4 approaches the theoretical value of two. This convergence is however not exactly 2.0. The difference between the two largest meshes follows a power law of 2.01. The origin of these small deviations could be the use of the effective element length. Not all elements in the prismatic mesh are equally shaped and the elements themselves have differently sized edges. This could lead to a small inaccuracy in the effective element length.

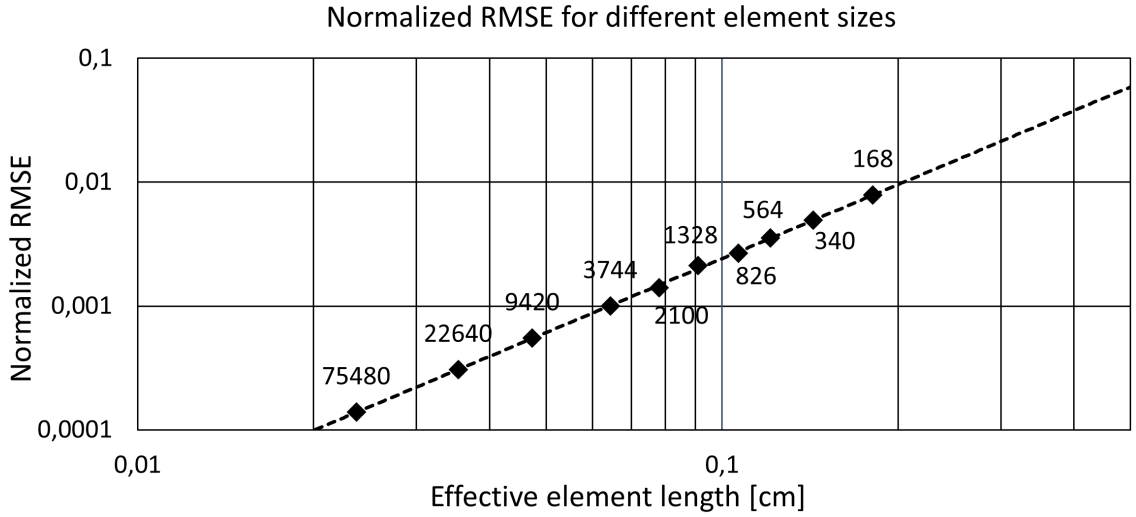


Figure 4.4: Normalized RMS error for prismatic meshes with varies effective element lengths on a double-logarithmic scale. The numbers correspond to the total number of elements in the mesh. The dotted trend line is the function $y = ax^b$ with b = 1.98.

Table 4.4: Parameters for simulation of a conductive graphite tube with a coolant channel with a fixed temperature on the inside.

parameter	value	units	meaning
D_i	1.588	cm	inner tube diameter
D_o	6.0	cm	outer tube diameter
H	320	cm	tube height
T_{inlet}	250	K	coolant inlet temperature
T_0	-	K	initial graphite temperature
p	1.5	W/cm^3	power density
γ	0.03	W/cm^2/K	heat transfer coefficient
ρ	1.8	g/cm^3	graphite density
c_p	0.72	J/g/K	graphite heat capacity
λ	1.33	W/cm/K	graphite thermal conductivity

4.3. Coupling of coolant and solid models

The DG model for the coolant temperature and the SIPG model for the solid temperatures are combined in a time-dependent coupled code as described in section 3.1. This model can be used to either calculate steady-state or time-dependent solutions using the Crank-Nicolson method for time integration. First, the steady-state coupling of the models is investigated. The next section focuses on the time-dependent behaviour of the coupled system. Both tests are done on a graphite tube with a constant power density in the solid. The inner surface of the tube has a boundary with a coolant channel. Through this boundary heat is exchanged. The parameters used in the code are shown in Table 4.4.

In the first test for the steady-state system the temperature over the entire tube is kept constant. A problem identical to that in section 4.1 is simulated and similar results are expected with only a single calculation of the coolant channel temperature. The results are shown in Figure 4.5 for simulations with 4 and 10 prisms over the height of the tube along with the analytical solution. The results are similar to the analytical solution.

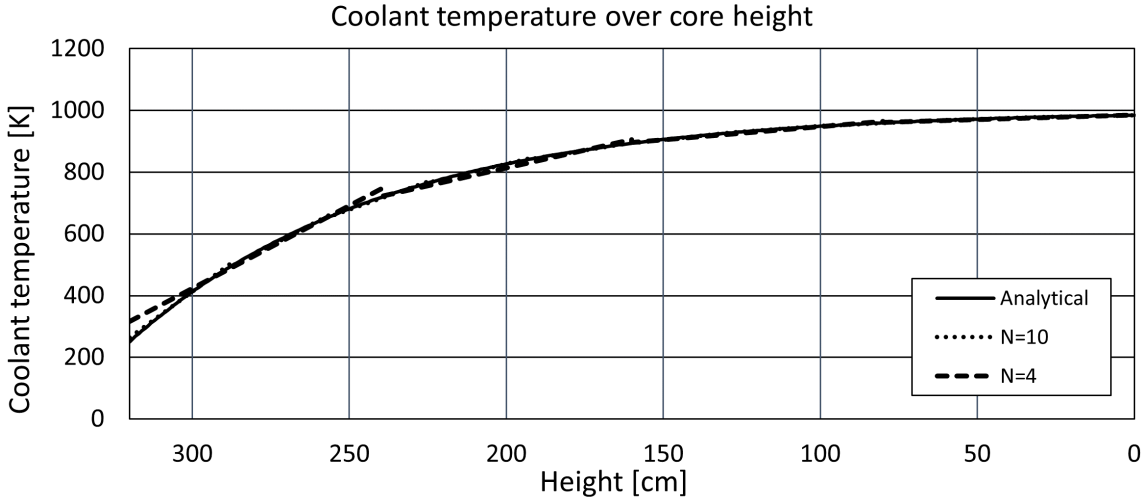


Figure 4.5: Comparison of the coolant temperature with $N = 4$ and $N = 10$ elements against the analytical solution with a constant wall temperature of 1000 K.

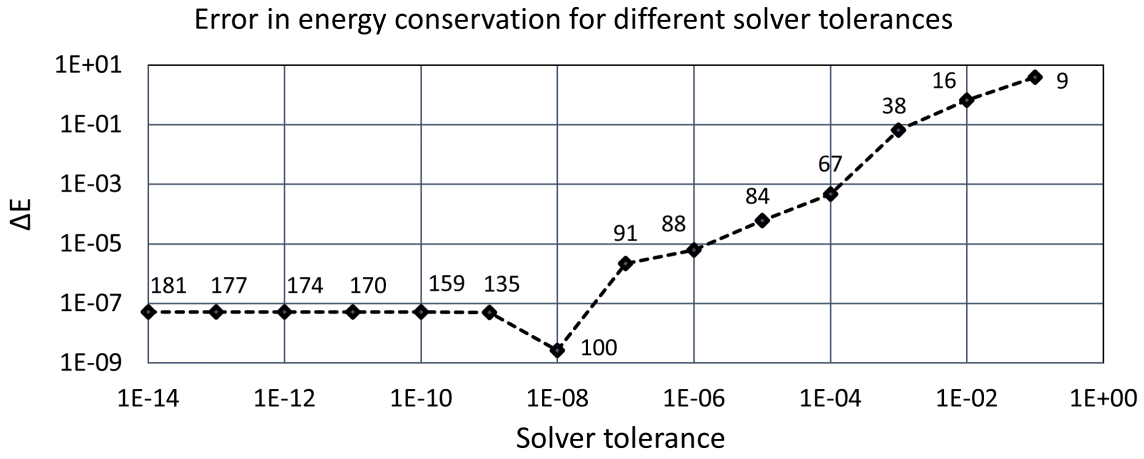


Figure 4.6: Differences in energy conservation for varying solver tolerances for the PETSc solver. The numbers depict the amount of loops necessary for the PETSc solver to reach convergence. All calculations were done on a mesh with 40 elements in the axial and 107 elements in the radial direction and with initial graphite temperature T=1000K.

For the second test of the coupled system, the walls of the tube are no longer kept at a constant temperature and the calculation of the solid temperature is taken into account. The power density in the graphite is kept constant. In the coupled system the code will iterate between calculations for the solid and coolant temperature. At each calculation of the solid temperature, the SIPG matrix is solved using the PETSc KSP solver. The tolerance of this solver affects the accuracy of the solution. Before analyzing the iterations in the coupled system, the PETSc KSP solver tolerance is investigated. This is done by adjusting the solver tolerance and computing the effect on the conservation of energy and the number of steps PETSc needs to solve for the SIPG system. For this problem with an insulated tube, constant power production and coolant channel boundary the energy conservation is given by:

$$p \int_{\Omega} dV \; = \; \int_{\Gamma_N} \gamma (T - T_{He}(z)) dA \; , \tag{4.8}$$

where $p \int_{\Omega} dV$ is the total heat production in the graphite and the integral over Γ_N is the integral over surface boundary of the channel. The total power production in the tube is calculated to be 12.5kW.
For a single calculation of the graphite temperature from the SIPG matrix the energy conservation is calculated with identical parameters and initial graphite temperature. Figure 4.6 shows a logarithmic plot of the solver tolerance against the energy difference along with the number of iterations the PETSc solver needed to converge. For solver tolerances above 10^{-8} the differences in energy remains roughly constant. Using the total power production inside the tube, the normalized energy difference is then in the order of the machine precision of 10^{-13}. In the rest of the results, the tolerance of the PETSc solver will be kept at 10^{-5} to prevent long simulation times.

In the following part the iterative calculation of the solid and coolant temperature is investigated. Iterations will continue until a 100 cycles have been calculated or until the change in the average graphite temperature is below 0.1 K compared to the previous iteration. This was done whilst varying three different parameters; the initial graphite temperature, the number of elements over the height of the tube and the refinement of the mesh around the coolant

channel. These tests were run for multiple meshes and the complete results can be found in Table A.1. in the appendix. An example of the solutions for the coolant and solid temperatures is shown in figures 4.7 and 4.8. These figures show the average temperatures over the height of the tube and a cross-section of the temperature profile in the plane where z=160 cm. The specific mesh that is used for the generation of these results has 40 elements in the axial direction and 107 elements in the radial direction. The effective element length is 8 cm in the axial and 0.22cm in the radial direction. From Figure 4.8 it can be seen that the temperature variation in the radial direction is around 5 K compared to a variation of almost 1000 K over the height.

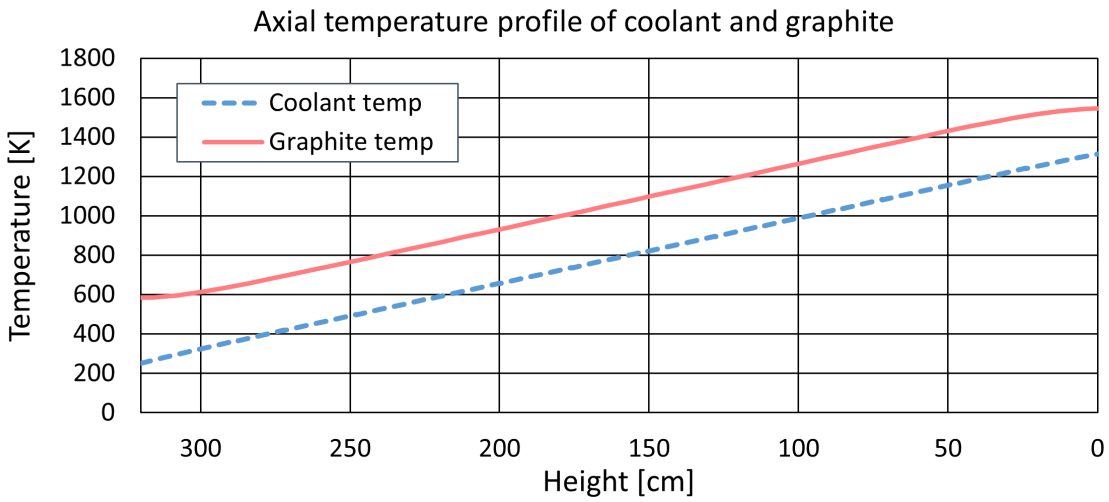


Figure 4.7: Temperature over the height of the core for the coolant and and the average graphite temperature per height for a mesh with 40 elements in the axial direction and 107 elements in the radial direction.

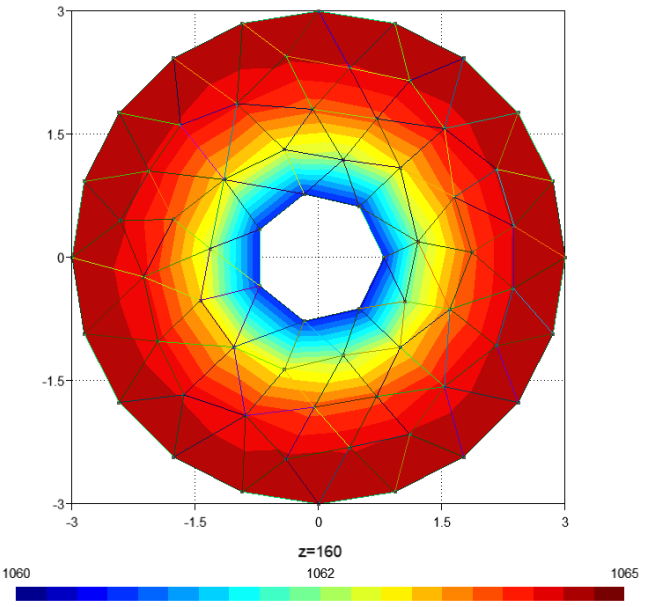


Figure 4.8: Graphite temperature profile for the plane z=160cm with 40 elements in the axial direction and 107 elements per height level [22].

To analyze the results from the runs in Table A.1. in the appendix, first the run time of the simulation is plotted for different numbers of elements in Figure 4.9. Here it can be seen that the computation time increases with the amount of elements. Choosing an initial temperature of 1000 K decreases the computation time compared to the less accurate guess of 0 K. What should be noted is that the number of iterations needed for the average graphite temperature to vary with less than 0.1 K was 14 times for all runs with 0 K and 13 times for all runs with 1000 K as initial temperature. However, the run time is mostly dependent on the number of elements. This can be explained with two reasons: 1) building the initial SIPG matrix takes longer for meshes with more elements. 2) At each iteration the PETSc KSP solver is used to calculate the solution to the SIPG matrix, the more elements the more iterations and time PETSc takes to converge.

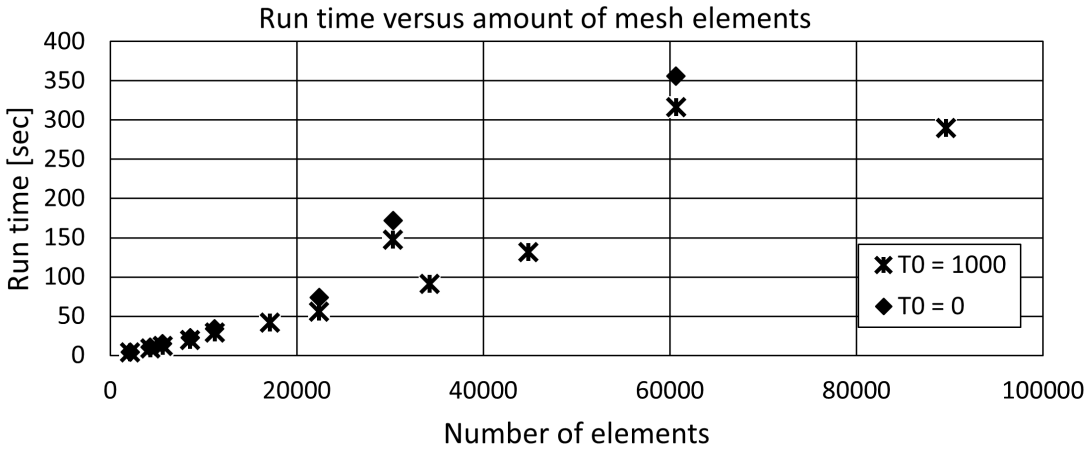


Figure 4.9: Run time compared to amount of elements in the mesh for runs with an initial graphite temperature of 0 K and 1000 K. Runs with 0 K as initial temperature all took 14 iterations before convergence and runs with 1000 K as initial temperature all took 13 loops.

The next parameters that are of interest are the number of elements over the height of the tube and the circle refinement around the channel. In Table 4.5 the maximum and average graphite temperatures and the outlet temperature of the helium is shown for different amounts of elements in the axial direction as well as for two different values for the circle refinement. The circle refinement is the amount of prismatic elements that is used to map the circumference of the channel. From Table 4.5 it can be seen that using more elements over the height increases the temperatures slightly. For 40 or more elements the temperature varies by less than 2 K. The outlet temperature of the helium gives only slight deviations of less than 1K, even for small numbers of elements.
The differences in temperature between the circle refinements of 10 and 7 in Table 4.5 are around 10 K. To investigate this behaviour further the temperatures are listed for different circle refinements in Table 4.6. All temperatures decrease the more elements are used to mesh the coolant channel. This can be explained from the fact that the actual coolant channel is a circle; the more elements used to mesh this circle, the more the meshing of the channel will approach the shape of a real circle. In Table 4.7 the relative differences in the circumference and cylinder volume are given for different amounts of elements around a channel. When meshing the channel two things change compared to the reference geometry: 1) the surface area of the channel decreases proportional to the change in circumference and 2) the volume of the channel decreases. Since the volume of the channel decreases, the volume of the

graphite slightly increases. Since the total power production scales with the graphite volume, the temperature in the solid increases, leading to a final increase in the coolant temperature.

Table 4.5: Maximum and average temperature of the graphite tube and outlet temperature of the coolant for different numbers of elements over the height of the core and meshes with 10 and 7 elements mapping the coolant channel circle. Temperatures are given in Kelvin.

	circle refinement = 10			**circle refinement = 7**		
N Height	Tmax	Tavg	Tout	Tmax	Tavg	Tout
20	1516.31	1056.57	1297.31	1527.41	1063.78	1306.53
40	1521.25	1054.41	1297.12	1532.37	1061,59	1306.52
80	1522.69	1053.85	1297.12	1533.82	1061,02	1306.51
160	1523.08	1053.72	1297.12	1534.21	1060,87	1306.51
320	1523.19	1053.67	1297.12	1534.32	1060,84	1306.51

Table 4.6: Maximum and average temperature of the graphite tube and outlet temperature of the coolant for different numbers of elements used to map the circle around the coolant channel.

Refinement	**Circle refinement**	**Tmax**	**Tavg**	**Tout**
1	7	1532.37	1061.59	1306.52
0.5	10	1521.24	1054.41	1297.12
0.2	20	1505.92	1044.51	1285.26
0.2	40	1500.99	1041.30	1280.12
0.2	60	1500.11	1040.72	1279.38

Table 4.7: Expected relative difference in the circumference and volumes of cylindrical objects for different numbers of elements used for mapping the circular surface compared to actual values.

# elements	**circumference**	**volume**
7	0.967	0.871
10	0.984	0.935
20	0.996	0.984
40	0.999	0.996
60	1.000	0.998

Discussion

For wall temperature, the temperature of the coolant follows the same analytic solution used in section 4.1, implicating a correct implementation of the DG method in the coupled code. The PETSc KSP solver tolerance decreases the error in the energy conservation calculation. Improvements stagnate for tolerances smaller than 10^{-8} when the normalized error is of the order 10^{-13}. Increasing the tolerance of the solver increases the number of steps for the solver to converge. The convergence of the solver together with the amount of elements in the mesh increases the total computation time. An accurate initial guess can decrease the computation time slightly, but since the code uses the temperature profile of the previous iteration as starting point for the solver this effect is small.

When taking into account heat production in the graphite and iterating between the solid and coolant temperatures, an increase in the number of elements over the height of the tube leads to a slight increase in the temperature. This increase converges the more elements are used. The outlet temperature of the coolant seems to be less susceptible to changes in the number of

elements than the graphite temperature. This can be explained from the one-dimensional DG results where small differences lead to accurate results. Since the temperature profile of the helium as shown in Figure 4.7 is linear, the linear basis functions will give accurate predictions for the outlet temperature.
The differences in temperature for different channel refinements in Table 4.6 are quite large. The temperatures when using 7 elements are around 2% larger compared to those with 60 elements. This is partially inline with the expected differences from Table 4.7: The total tube has a volume of $\pi \cdot 3^2 H$ cm^3 minus the volume of the channel. If the volume of the channel decreases with 13% in the case of 7 elements, this leads to an increase in graphite volume and power production of 1%. However, the temperature increase is 2%. The additional 1% can be explained by the difference in the outward heat flux. The heat flux out of the tube scales with the area of the channel and the temperature difference between graphite and coolant. Using 7 elements compared to 60, the total area decreases by 3.3%. Since the total total heat flux needs to remain constant, the temperature difference between the graphite and coolant will increase, which can be seen in Table 4.6. Using 20 or more elements for the circle refinement will decrease the error to less than 1%. However, using a circle refinement of 20 elements requires an extremely fine mesh with a very large amount of elements which would not benefit the computation time.
In the tests for the coupled system, it was assumed that a steady-state was reached when the average solid temperature varied less than 0.1 K from the previous calculation. However, without deviations in the average temperature, still changes in the profile can occur. Since the simulated problem is symmetrical in the radial direction and the radial distances are quite small it is unlikely that this is a problem in the current results. When simulating more complex geometries preferably a different metric is used, such as the root-mean-squared difference of the temperature profile compared to the previous iteration.

Table 4.8: Parameters time-dependent simulations of a graphite tube with a fixed coolant temperature on the inside.

parameter	value	units	meaning
D_i	1.588	cm	inner tube diameter
D_o	6.0	cm	outer tube diameter
H	320	cm	tube height
T_{in}	250	K	coolant temperature
T_0	0	K	initial graphite temperature
γ	0.03	W/cm^2/K	heat transfer coefficient
ρ	1.8	g/cm^3	graphite density
c_p	0.72	J/g/K	graphite heat capacity
λ	1.33	W/cm/K	graphite thermal conductivity
α	1.03	J/g/K	graphite thermal diffusivity

4.4. Time dependence

In this section the time-dependent model is investigated. For the time integration the Crank-Nicolson method as described in section 3.3 is used. This is a second-order accurate finite difference approximation for time integration. When $\alpha\frac{\Delta t}{\Delta x^2} > \frac{1}{2}$ the CN method tends to oscillate. The occurrence and effect of these oscillations are investigated by solving the temperature on a similar graphite tube as in the previous section but with a larger outer radius of 4cm. Again, the inside of the tube is a coolant channel boundary and the outside of the tube is insulated. In this simulation the coolant channel inside the tube is kept at a fixed temperature of 100 K. Before t=0 the temperature of the graphite tube is T_0 = 0 K. At t=0, the graphite is heated by the heat transfer with the coolant channel according to the Nusselt relation. The parameters used in this simulation are shown in Table 4.8. The problem is simulated during a total time of 10 seconds with different step sizes. As benchmark for the temperature over the tube, a run with time steps of 0.002 seconds is taken. A cross-section of the tube mesh is shown in Figure 4.10. The radial element length of the mesh is 0.5cm. The expected tipping point for oscillation is then calculated to be $\Delta t > \frac{1}{2}0.5^2 = 0.125$. The average temperature and the temperature of two different elements in the mesh are compared to the results for the benchmark run to get the error for each step size. A top view of the tube mesh and the location of the two elements is shown in Figure 4.10.

Figure 4.11 shows the average temperature over the tube and the temperatures in element 1 and element 2. Element 1 is closer to the coolant channel edge, thus heats up faster than element 2 which is nearer to the outer edge of the tube.
From Figure 4.11 it can be seen that the temperature of element 1 varies the most over the span of the simulation. Therefore the temperature of this element was used to plot the temperature for larger time steps. The difference in temperature between the benchmark with $\Delta t = 0.002$ is plotted in Figure 4.12 for simulations with $\Delta t = 1, 0.5, 0.1$ and 0.05.

As expected the solution for the temperature tends to oscillate for larger time steps. The oscillation damps out after a large number of time steps. Table 4.9 shows the number of time steps required for the oscillations to be damped to less than 0.01 K from the benchmark temperature. The relation of the number of time steps needed for the oscillation to damp appears to be linear. However, since the simulated time increases with the time step size as well, the relation appears quadratic in total time.

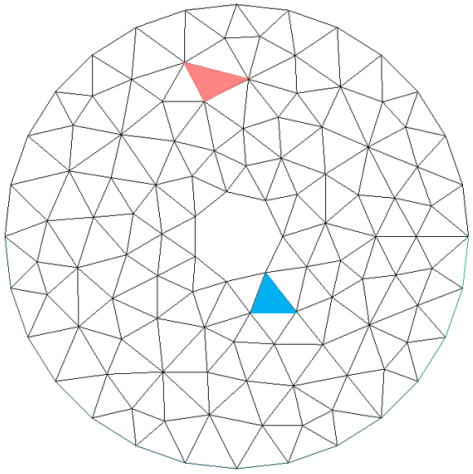


Figure 4.10: Radial view of tube mesh used for time-dependent calculations. The blue element depicts element 1 and the red element is element 2 [22].

Table 4.9: Time and number of time steps needed the temperature difference with the benchmark temperature is reduced to 0.01 K for different time steps sizes and θ = 0.5.

Δt	2	1	0.5	0.2	0.1	0.05
Damping time	120	30	7.5	1.4	0.3	0.05
# of time steps	60	30	15	7	3	1

Discussion

The CN method is prone to spurious oscillations in the solution. These oscillations can be seen in Figure 4.12. For time steps of 1 second or smaller these oscillations are between 0.7 and 0.4 K. Since the temperature scale of the simulated problem is between 0 and 3 K the relative error of during the first 10 seconds of the simulation decreases from 7% to 1% . Depending on the required accuracy in the temperature profile, the size of the time steps needs to be chosen. Naturally, transients over a larger mesh will allow for larger time steps sizes. For the calculations on the U-Battery® the mesh sizes will be of the order of the simulation for the graphite tube in this section. For transients calculations taking less than a few minutes, time steps smaller than 0.1 seconds can be chosen. However, for longer transients larger time steps are recommended to avoid extremely long simulation times.

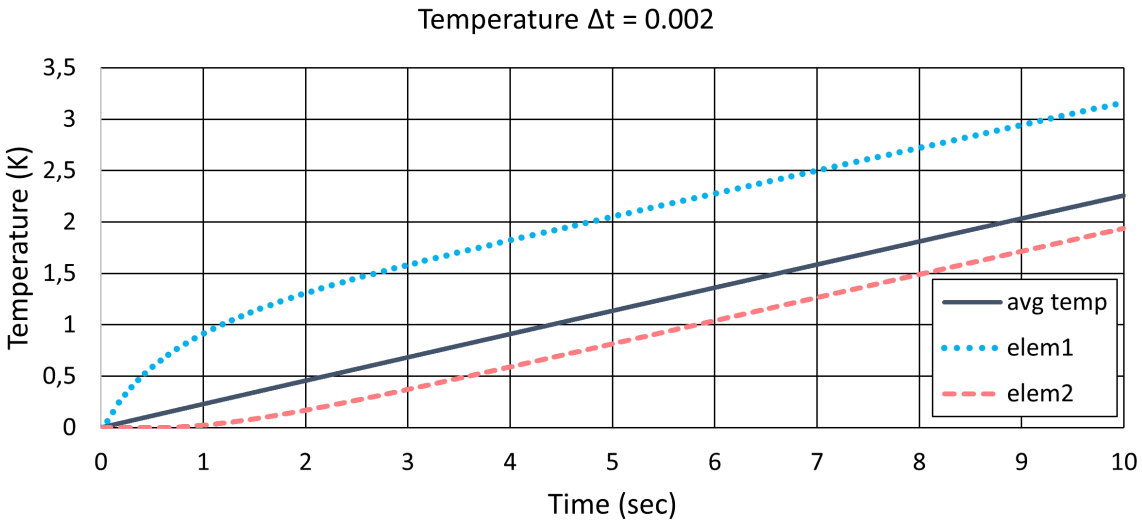


Figure 4.11: Benchmark temperature for a tube heating up from the inside over 10 seconds. Average temperature and the temperatures of two elements in the mesh are shown for 5000 time steps of size 0.002 seconds.

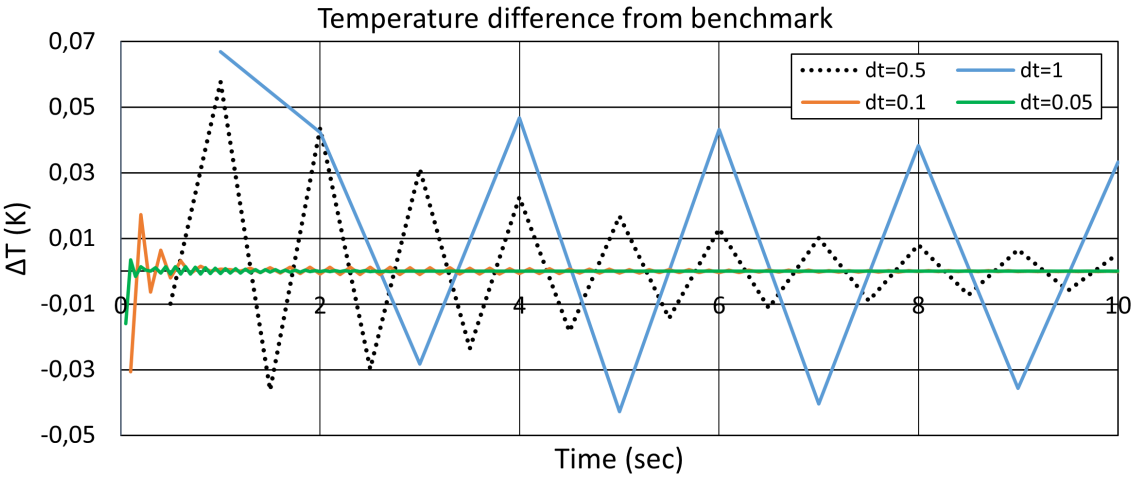


Figure 4.12: Difference between benchmark calculations and the temperature of element 1 with time steps of 1, 0.5, 0.1 and 0.05 seconds.

5

Fuel block simulation

In this chapter the results of the thermal-hydraulic code are shown for simulations on four stacked fuel blocks of the U-Battery®. First, the mesh, parameters and power profile used in the simulation are given. After this, the results for a steady-state calculation of the fuel block temperature are shown followed by transient simulations for the hypothetical blockage of various coolant channels.

5.1. Mesh and parameters

The four stacked fuel blocks are simulated as an isolated system with coolant flowing through the channels and insulated at the outer edges of the fuel block configuration, simulating something similar to a central fuel block. The mesh for the fuel blocks is combined in a single mesh spanning the height of the reactor core and is created in GMSH using the Python api [22]. The refinement of the mesh and the number of elements used to mesh circular surfaces can be adjusted. To keep the computation times for the simulations limited, it was chosen to work with 40 elements in the axial direction and each channel is surrounded by 8 radial elements. A cross-section of the mesh is shown in Figure 5.1. Figure 5.2 gives a zoom of this mesh. The parameters for the model and mesh are given in Table 5.1.

For the power density profile inside the core it is assumed that power production only takes place in the fuel compacts and not in the graphite. The fuel rods are simulated as cylinders spanning the total height of the core. The total power density per fuel rod is calculated from the total power of the U-Battery® of 20MWth. For a total of 37x216=7792 fuel rods, this gives an average power per fuel rod of 2.5kW. The average power density in the fuel is calculated by dividing by the fuel rod volume. Taking into account the meshed volume of the fuel rods, this leads to an average power density of 7.1 W/cm^3. The power density varies over the height of the core. Since the temperature of the fuel will be higher at the bottom of the core, the power production is higher near the top of the core. Similar to the power profile as used in Ding et al. (2013) [10], a cosine-like power profile is used with a peak of 8.0 W/cm^3 at two-thirds of the core height as shown in Figure 5.3. The average power density of the function over total rod is kept at 7.1 W/cm^3.

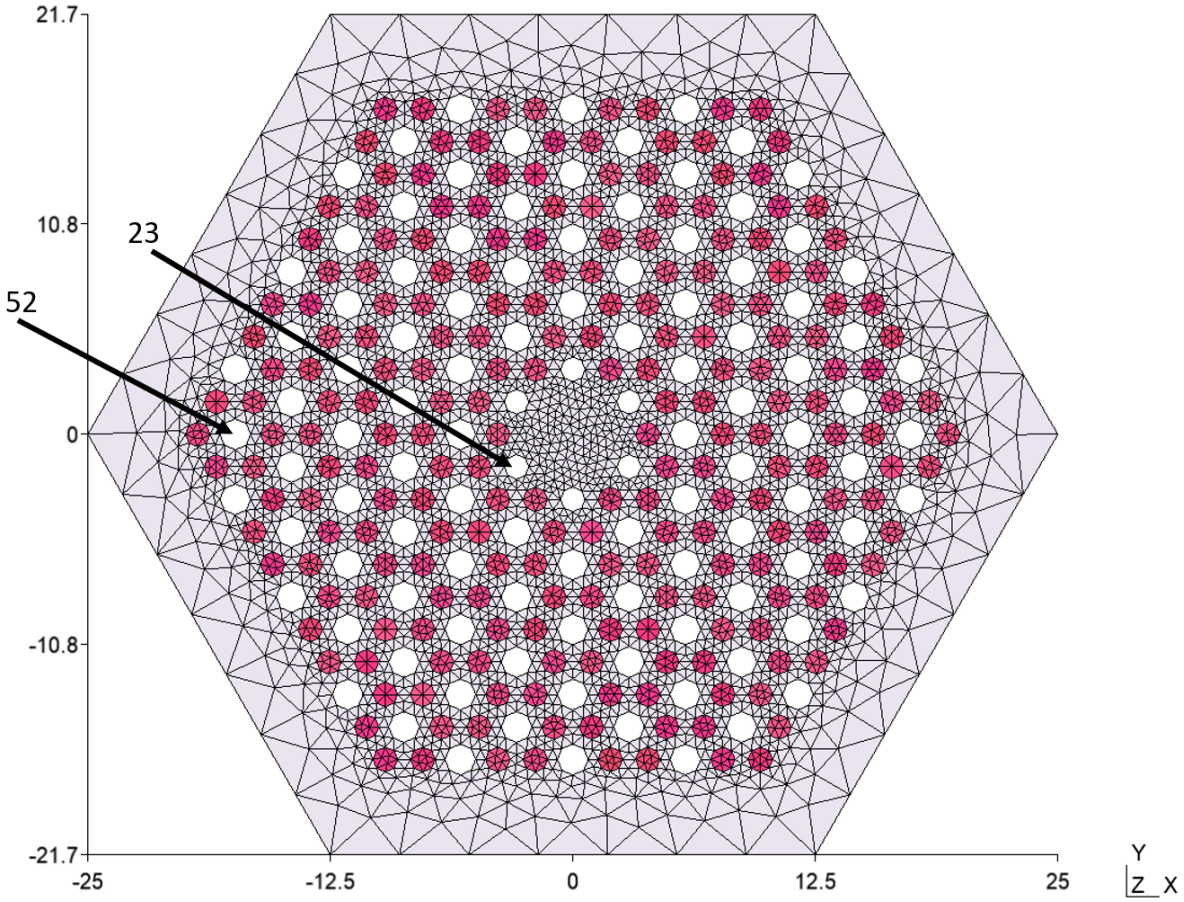


Figure 5.1: Cross-section of the mesh used for simulating a fuel block made in GMSH [22]. The red surfaces are fuel compacts and the grey surface is the graphite. Coolant channels are empty. The arrows point to cool channel number 23 and 52. The six coolant channels in the middle of the fuel block have the same diameter as the fuel channels in accordance with the design of the U-Battery® [3].

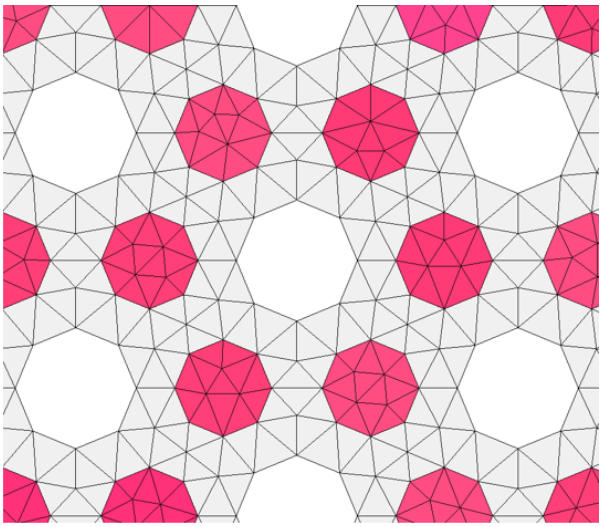


Figure 5.2: Zoom of the cross-section of the mesh [22].

Table 5.1: Model and mesh parameters and their value and meaning for simulation of the U-Battery® fuel block configuration.

parameter	value	units	meaning
T_{in}	520	K	inlet coolant temperature
γ	0.025	W/cm^2/K	heat transfer coefficient
ϕ_m	1.91	g/s	coolant mass flux per channel
ρ_g	1.8	g/cm^3	graphite density
ρ_f	2.7	g/cm^3	fuel density
$c_{p,g}$	0.72	J/g/K	graphite heat capacity
$c_{p,f}$	0.25	J/g/K	fuel heat capacity
λ_g	1.33	W/cm/K	graphite conductivity
λ_f	0.43	W/cm/K	fuel conductivity
α_g	1.03	cm^2/s	graphite diffusivity
α_f	0.64	cm^2/s	fuel diffusivity
R_{fb}	25	cm	radius of fuel block
D_c	1.588	cm	coolant channel diameter
D_f	1.27	cm	fuel channel diameter
H	320	cm	height
N_C	8	-	elements per channel
N_H	40	-	elements in z-direction
refinement	0.5	-	general mesh refinement
N	406796	-	total number of elements

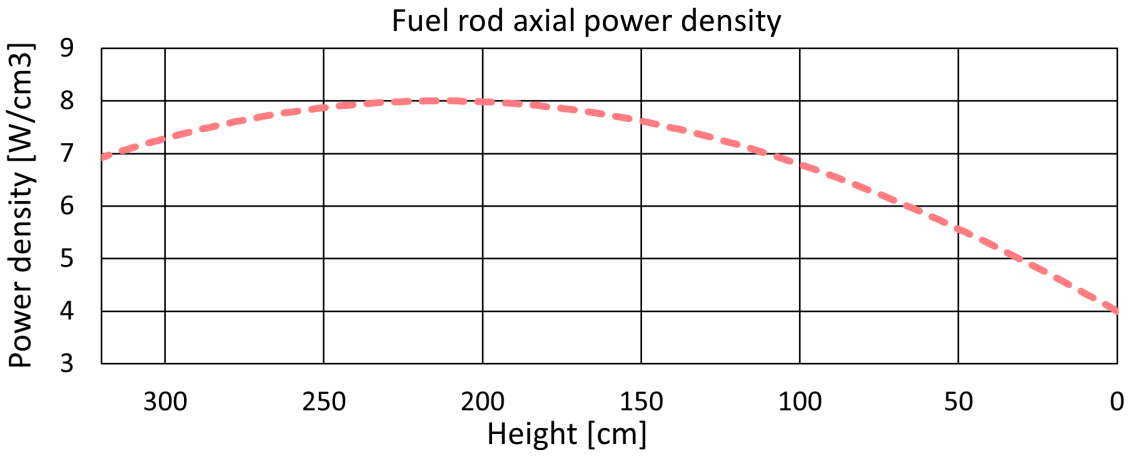


Figure 5.3: Axial power density profile for a fuel component. The function is given by: $P(z) = 8.0\cos\left(\frac{\pi(z-2H/3)}{2H}\right)$, with H=320cm.

5.2. Steady-state simulation

For the first simulation, the iterative model is used to calculate the steady-state temperature of the stacked fuel blocks with the power density profile in the fuel components. The results for the steady-state temperature profile of the solids and the coolant are shown in Figure 5.4 and Figure 5.5. The color scales of the figures vary since the variation in the radial direction is around 20 K, compared to a variation of 440 K between the top and bottom of the core. The average outlet temperature for the coolant is 1046.1 K. For the warmest and coldest channels this is 1049.4 and 1013.1 K respectively. The warmest channels are the six coolant channels on the outer edges of the hexagon. The six smaller coolant channels in the middle of the fuel blocks are the coldest. These channels are marked with an arrow in Figure 5.1. Similarly, the six fuel rods on the outer edges have the highest temperatures and the fuel rods in the middle have lower temperatures. The maximum temperature is reached in the fuel at the bottom of the core and is 1120.6 K.

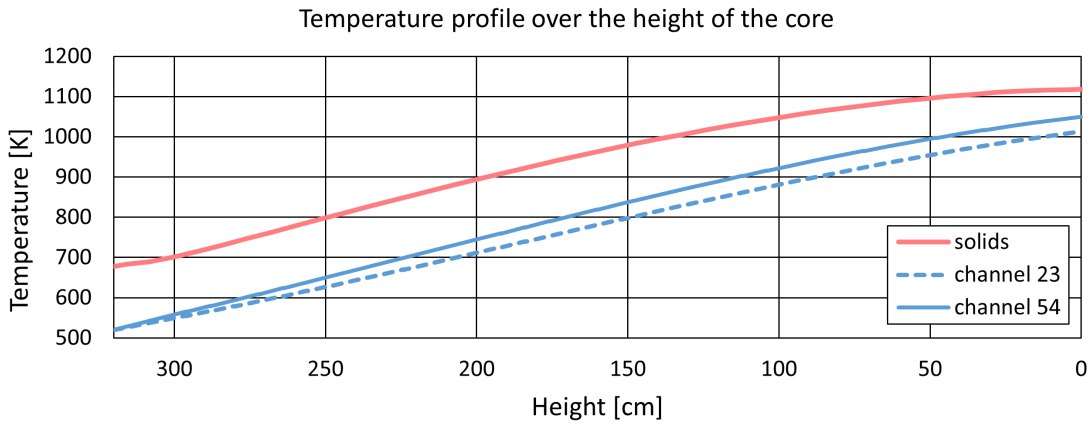
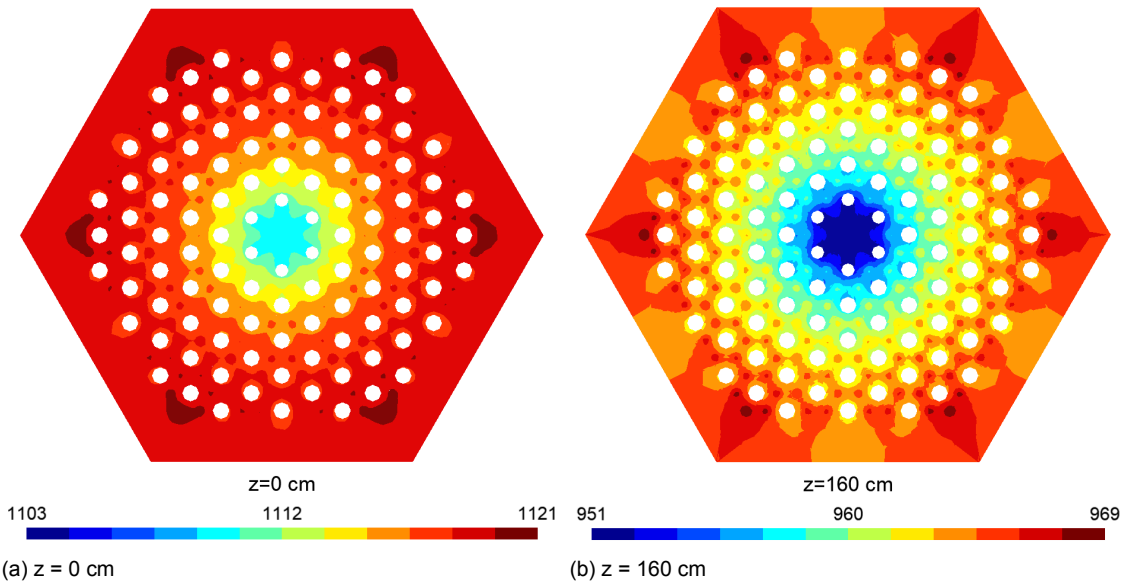


Figure 5.4: Temperature profiles over the height of the core for the average solid temperature, the coolant channel with the highest outlet temperature (54) and the channel with the lowest outlet temperature (23).



(a) z = 0 cm (b) z = 160 cm

Figure 5.5: Cut of the solid temperature profile for the planes z=0, the bottom of the core and z=160, the middle of the core. The color scales are in Kelvin and vary for the two temperature plots.

Discussion

The steady-state temperatures of the solid and coolant give an impression of the temperature differences within the U-Battery®. The temperature variations over the radial directions are quite small. The temperature difference between fuel compacts and the surrounding graphite is a few degrees. This small difference can be explained from the values for the thermal conductivity compared to the distance between the channels and the power input: the distances between the channels are approximately 1 cm, the conductivity λ_g is 1.7 W/cm/K. Since the heat production is in the order of 7W/cm^3, the expected temperature differences are around $\frac{7}{1 \cdot 1.7} = 4$ K. The temperature variation over the height of the fuel blocks is much larger than the radial temperature difference. The average increase of the coolant temperature over the core is 526 K. This can be assumed to be an overestimate: The current steady-state simulation uses insulating boundary conditions at the boundary of the fuel block so all produced power is transferred to the coolant. Additionally, eight radial elements were used to mesh the channels. In the previous chapter it was shown that an increase in the circle refinement decreases the calculated outlet temperature.

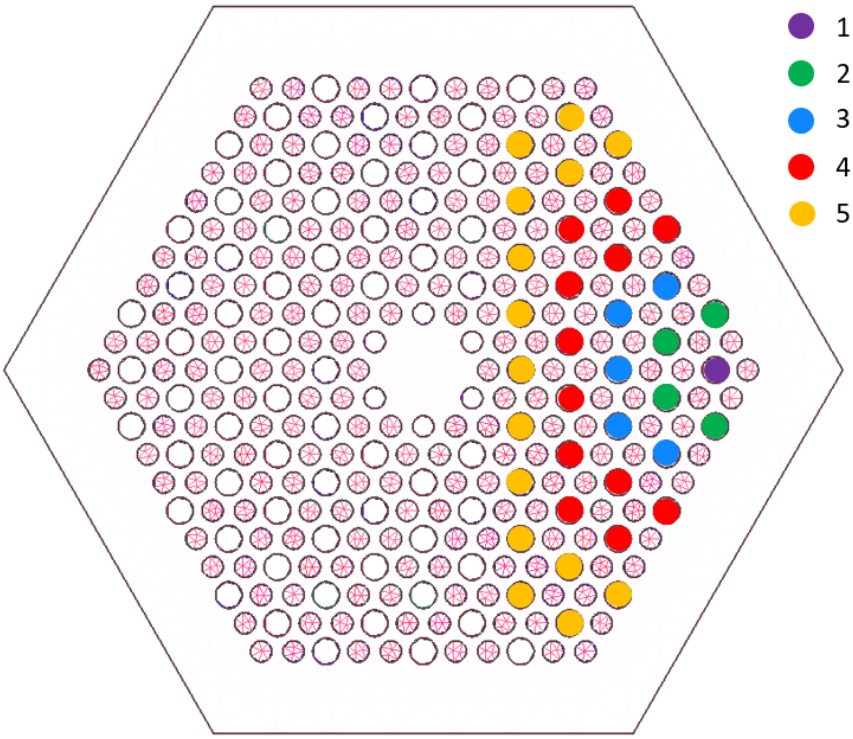


Figure 5.6: Cross-section of the fuel block mesh showing the blocked channels for each simulation. Channels blocked in the previous simulation remain blocked in the next.

5.3. Transient calculation for blocked coolant channels

The time-dependent thermal-hydraulics model of the U-Battery® is used to study the temperature changes in the hypothetical case of a blockage of coolant channels. The blockage of a channel is reproduced in the simulation by changing the Neumann boundary condition on the coolant channel surface to the insulating boundary condition. This is done for five different configurations with 1, 5, 10, 22 and 37 blocked channels labeled simulation 1 to 5. The channels blocked in these simulations are marked in Figure 5.6. The channels blocked in simulation 1 remain blocked in simulation 2 and so on. The steady-state temperature profile from the previous section is used as input for the temperature at $t < 0$. The temperatures are calculated during a transient of 360 time steps of 10 seconds, spanning a total time of one hour. The variation in maximum and average temperature compared to the temperatures at $t < 0$ is plotted over time for the five configurations of blocked channels in Figure 5.7 and 5.8. The final maximum and average temperature increases the more channels are blocked. The solid temperatures at $t = 3600$ seconds are plotted in Figure 5.9 for the bottom plane of the core. These figures show that there is a build up of heat at the right-side of the fuel blocks, around the blocked channels. TRISO fuel particles can withstand temperatures up to 1870 K without damage. During the transient the maximum temperature of the fuel does not exceed this value, even for a blockage of 35% of the coolant channels in the fuel block.

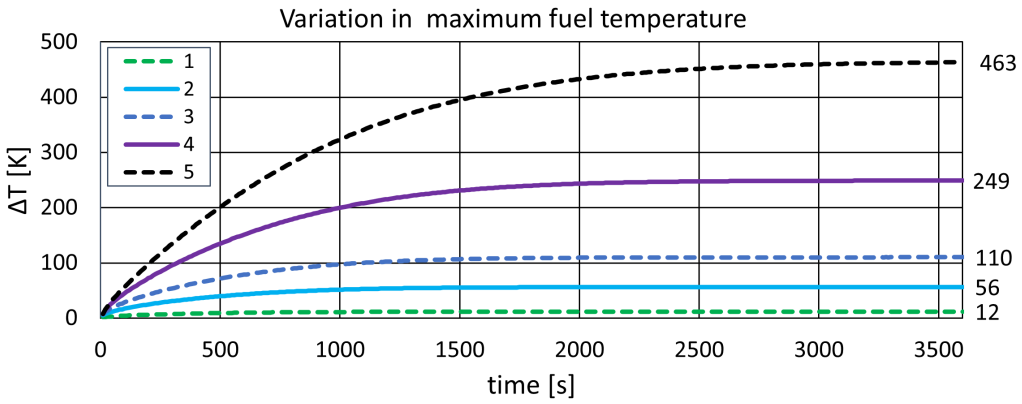


Figure 5.7: Difference in the maximum fuel temperature compared to the temperature at t<0 for simulations with 1, 5, 10, 22 and 37 blocked channels. The values on the right-side of the graph indicate the difference at t=3600.

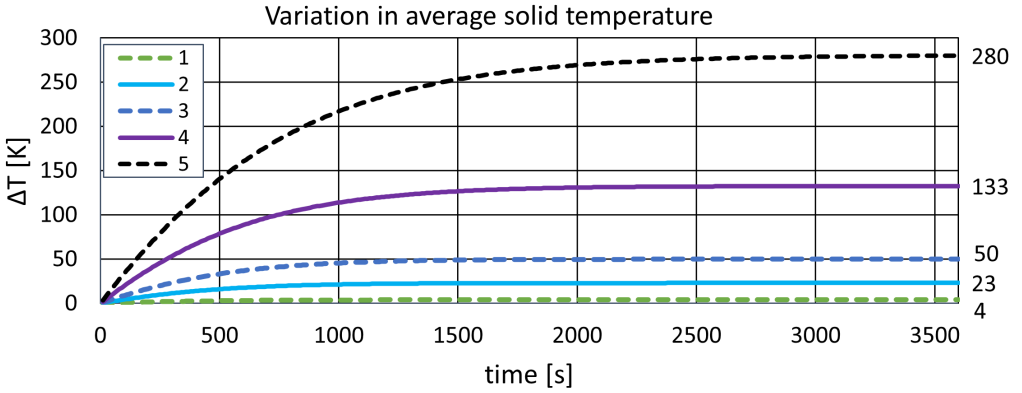


Figure 5.8: Difference in the average solid temperature compared to the temperature at t<0 for simulations with 1, 5, 10, 22 and 37 blocked channels. The values on the right-side of the graph indicate the difference at t=3600.

In the previous chapter it was shown that the Crank-Nicolson method shows oscillations for time steps that are large compared to the diffusivity and element scale. The radial element lengths of the fuel blocks are around 0.7cm. With time steps of 10 second and a diffusivity of 1 cm^2/s oscillations in the CN method are expected to occur in the simulations. Figure 5.10 shows a zoom of the temperature profile for the first simulation where only a single channel is blocked. The first three time steps are shown for 10, 20 and 30 seconds after the channel is blocked. The temperature around the channel decreases from the first to the second time step and increases again in the third time step.

Discussion

During the transient, the temperature of the solid increases during the first 2000 seconds. This is in accordance with the expected diffusion time calculated from the thermal diffusivity $\alpha \approx 1$ cm^2/s and the width of the fuel block of 50 cm. The time it takes for heat to diffuse from the right-side of the fuel block to the left side is approximately $\Delta x^2\ 2\alpha = 1250$ seconds. As expected the heat builds up around the blocked channels on the right. The radial differences in the solid temperatures vary from 20 K for the case of a singe blocked channel up to 201 K for 37 blocked channels and are much larger compared to the steady-state variations. The maximum fuel temperature for the blocked channels was 1578 K which is still within the safety limits for preventing damage to the fuel particles. Additionally, the transient leaves for sufficient time to intervene for reactor operators by for instance inserting control rods.

The time steps taken in the simulation are large compared to the element and diffusivity of the materials. Figure 5.10 shows that for the first time steps the temperature profile oscillates slightly. This behaviour can be explained from the large time steps compared to the diffusivity and element length.Since the temperature profiles vary only a few Kelvin, they do not affect the conclusions on the safety of the reactor.

In general the simulations differ at some points from the actual situation for blocked coolant channels: First, the blocked channels are modelled as insulating boundaries. This choice is explained from the fact that the helium left in the blocked channels will eventually heat up to the temperature of the surrounding graphite. Additionally, the fuel block in the simulation is isolated from the other fuel blocks and has insulating boundaries. In reality, the heat will also transfer to the surrounding fuel blocks or to the reflector, thus decreasing the temperature of the fuel block. Finally, an increase in temperature in general leads to a decrease of the power production in the fuel. Thus it can be expected that in reality the power profile will not remain constant, but will actually decrease over time. This will reduce the calculated temperature difference as well.

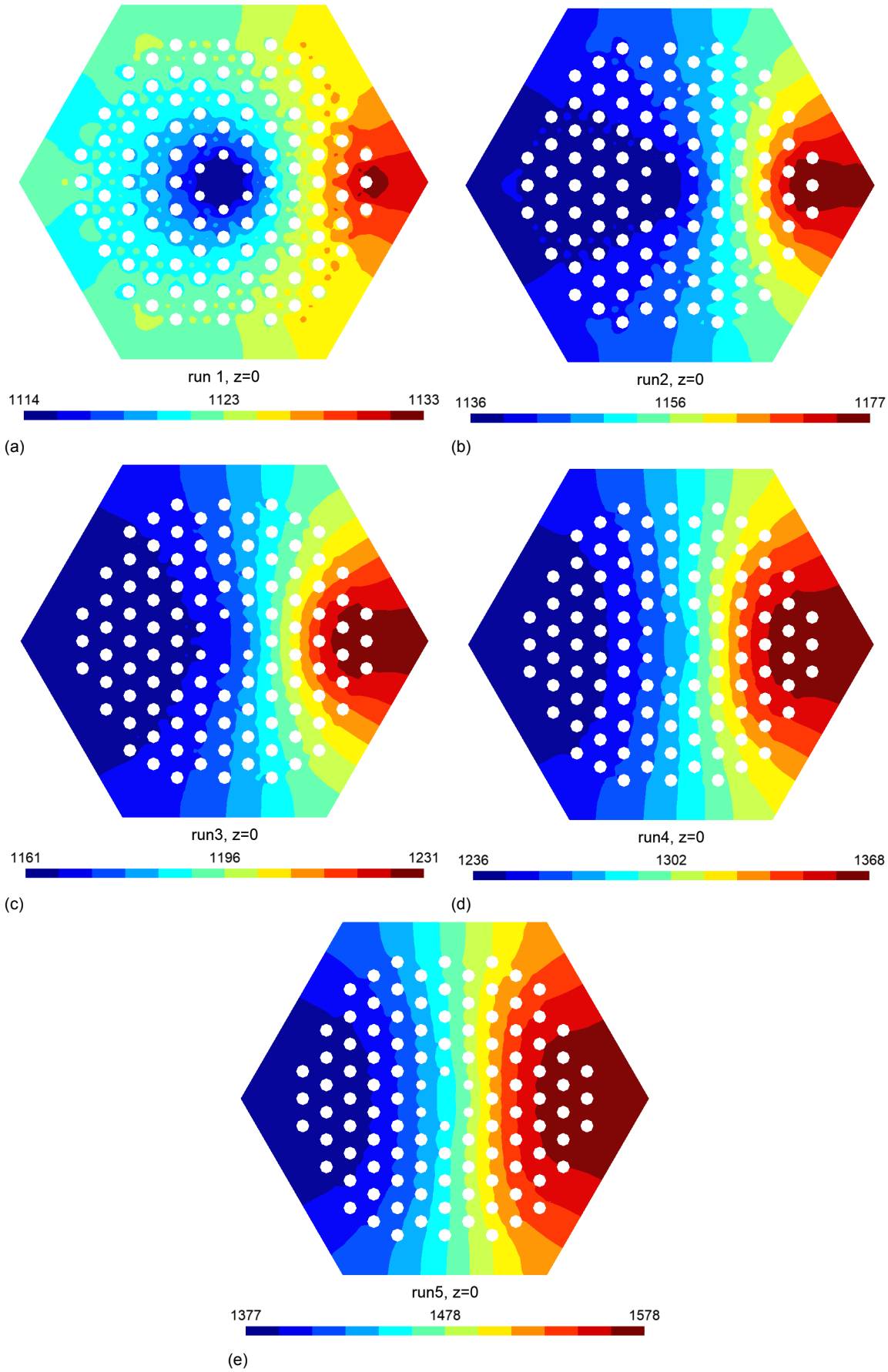


Figure 5.9: Temperature profiles of the solid at t=3600s at the bottom of the core for the 5 different blocked channel configurations. The color scales are the temperature in Kelvin. All figures have different color scales due to the large differences between the simulations [22].

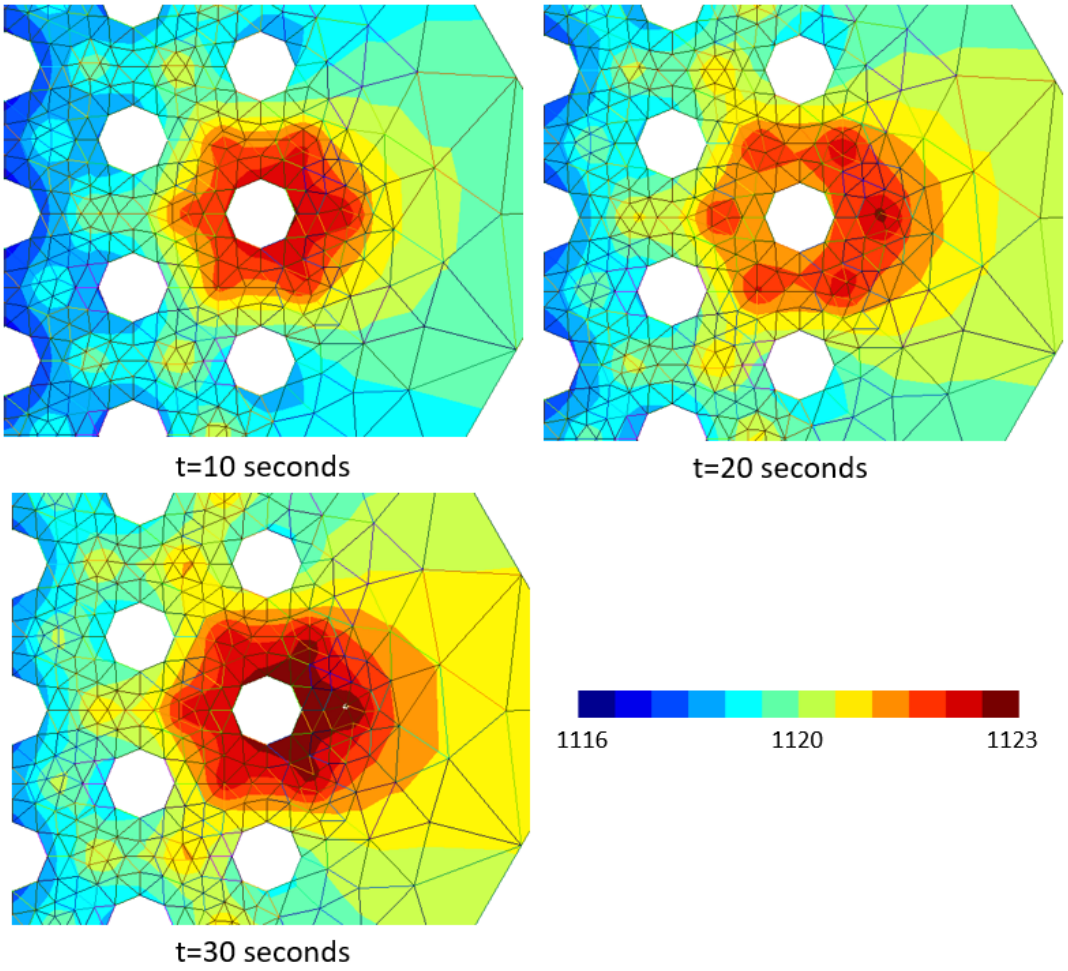


Figure 5.10: Temperature of the solid in the plane where z=0 for blockage of a single channel at the first, second and third time step of transient calculation. The color scale is the temperature in Kelvin [22].

6

Conclusions and recommendations

In this chapter conclusions are drawn up on the work done in this research and recommendations for improvements and future research are made.

6.1. Steady-state models

Two separate codes have been created for thermal-hydraulic calculations of the U-Battery® core. One for the coolant channel temperature based on the discontinuous Galerkin method and one for the solid temperature based on the symmetric interior penalty Galerkin method. The results for the one-dimensional coolant channel model follow the analytical solution and show a theoretical convergence to second-order accuracy using linear basis functions as well as conservation of energy over the solution.
The second code for the three-dimensional solid structures uses linear prismatic elements for meshing the geometries. The implementation of this element type in the Fortran90 code has been verified by simulating a neutron transport problem on a box. The results of the SIPG method follow the analytical solution and the root-mean-square error converges with an order of 1.98, close to the theoretical convergence of 2 for linear basis functions.

6.2. Coupled time-dependent model

The two separate steady-state codes have been combined using an iterative algorithm. The results of this combined code have first been analyzed for steady-state simulations on a graphite tube with a single coolant channel on the inside. For calculations with a constant graphite temperatures the coolant temperature follows the analytical solution. Different parameters have been analyzed for a constant power density in the graphite. The PETSc KSP solver tolerance decreases the error in the energy conservation calculation, improvements stagnate for tolerances smaller than 10^{-8}. Increasing the number of elements over the height leads to a small increase in the temperature that converges the more elements are used. Increasing the number of prisms used for meshing the coolant channel leads to a decrease in the maximum and average graphite temperature as well as the outlet temperature of the coolant. This is explained by differences in the surface area and volume of the channel when varying the number of elements.
For time-dependent calculations the Crank-Nicolson finite difference method for time integration has been applied to the coupled system. Since the CN method relies on the helium temperature at the next time step to calculate the solid temperature, the extrapolated solid temperature has been used to calculate the helium temperature at the next time step. The CN method is prone to spurious oscillations in the solution depending on the size of the mesh and

the diffusivity of the material. In the simulation of the single coolant channel the amplitudes of the oscillations are smaller than 0.1 K for time steps below 1 second.

6.3. Fuel block simulation

A configuration of four stacked, insulated fuel blocks of the U-Battery® has been simulated using combined thermal-hydraulics code. With the assumed parameters the average temperature increase of the coolant over the reactor is 526 K. For the solid structures the maximum temperature of 1121 K is reached at the bottom of the reactor in the six outer fuel channels. A time-dependent transient has been simulated for the hypothetical blockage of 1, 5, 10, 22 and 37 coolant channels for 360 time steps of 10 seconds. The maximum increase in fuel temperature compared to the steady-state simulation was 463 K for 37 blocked channels. The maximum temperature is within safety limits for all simulations. Oscillations of the Crank-Nicolson method in the temperature are seen in the first time steps of the simulation. The variations in temperature are a few Kelvin.

6.4. Recommendations

In this section first recommendations are made regarding possible improvements to the current code followed by recommendations for future research.

Temperature dependence of material parameters

In the current code the material parameters are taken to be constant with respect to the temperature. As was shown in section 2.2.1 this is not true for the thermal conductivity; for graphite λ at 500K is almost twice as large as λ at 1250 K. Other parameters that are assumed to be constant in this research such as the material densities and viscosity are also dependent on temperature. These parameters affect the heat diffusion through the solid as well as the heat transfer between the coolant and graphite. The temperature dependence of the parameters could be added to the current code using experimental relations for the parameters with respect to temperature. The material parameters can then be calculated at each new time step using the temperature profile of the previous time step. A disadvantage of this is however that the SIPG matrix L and mass matrix M need to be calculated anew at each time step. This will increase the computation time of the code.

Coolant model

In the coolant model the convection of helium is taken to be forced and the natural convection is neglected. For computing general temperature profiles in the U-Battery® this is an acceptable approximation due to the high pressure. However, when simulating loss of forced cooling incidents natural convection takes over from the forced convection and should ideally be taken into account. Natural convection is caused by variations in density due to temperature differences. Implementing natural convection should go hand-in-hand with adding temperature dependent material parameters.

Time-dependent coupled model

For the simulation of the fuel block large time steps were taken compared to the size of the mesh and the diffusivity. The Crank-Nicolson method shows small oscillations in the temperature profile around the coolant channels. Two possible methods to decrease these oscillations are to build up the time steps size at the beginning of the simulation or by taking a larger value for θ in the Crank-Nicolson scheme.

Elements and meshing

The solid materials are currently modeled using linear prismatic elements. It was shown that using these elements leads to a difference in the surface area and volumes of the channels. This results in inaccuracies in the temperature calculations. A solution to this problem would be to use curvilinear elements. This way the curved geometries of the coolant and fuel channels can be meshed precisely. Implementing the curvilinear prismatic elements requires the addition of second-order basis functions for the prisms and at least third-order Gauss quadrature for integration. The basis functions and integration points can be found with to the method used in section 2.2.3 where the outer product of the triangular and line elements is used to compute the basis functions. Implementing higher-order prisms will increase the accuracy of the solution, but will also increase the complexity of the computation.

Future research

A first recommendation for future research is to do simulations with different (time-dependent) power profiles as input. The goal of the thermal-hydraulics code is to be coupled to neutronics calculations. In doing so the temperature output of the code in this research is used as input for neutronics calculations resulting in a more accurate power profile of the U-Battery®.
Additionally, for simulating the blockage of coolant channels in the fuel block the fuel rods were simulated as a single material with one parameter for the diffusivity. The fuel rods consist of smaller fuel compacts with graphite and TRISO particles. These particles have a complex structure of uranium-oxide covered with different carbide layers. Each layer of the TRISO particle has a different diffusivity. It would be interesting to investigate whether taking into account these differences in material parameters changes the maximum temperature in the fuel and if accurate averaged parameters for the fuel can be found for simulating the fuel rods.
However not shown in the results of this thesis, the current code can be used to simulate time-dependent power profiles. This opens up possibilities to investigate the behaviour of the temperature profile during transients where the power over the reactor changes such as a start-up or shut down of the reactor.
Finally, the thermal-hydraulics code is able to work with asymmetrical three-dimensional simulations. Since transients in previous research were done in one or two dimensions only, the current code opens up new possibilities for asymmetrical transient simulations. Blocking for instance more configurations of coolant channels over multiple fuel blocks.

References

[1] Marco E. Ricotti and Roman V. Fomin. *Nuclear Reactor Technology Development and Utilization*. Woodhead Publishing, 2020, pp. 187–211. ISBN: 978-0-12-818483-7. DOI: https://doi.org/10.1016/B978-0-12-818483-7.00005-6. URL: https://www.sciencedirect.com/science/article/pii/B9780128184837000056.

[2] *Benefits and Challenges of Small Modular Fast Reactors*. TECDOC Series 1972. Vienna: INTERNATIONAL ATOMIC ENERGY AGENCY, 2021. ISBN: 978-92-0-124021-7. URL: https://www.iaea.org/publications/14928/benefits-and-challenges-of-small-modular-fast-reactors.

[3] Ming Ding et al. *Design of a U-Battery*. TU Delft and Manchester University, 2011.

[4] Ming Ding and Jan Leen Kloosterman. "Thorium utilization in a small long-life HTR. Part I: Th/U MOX fuel blocks". In: *Nuclear Engineering and Design* 267 (2014), pp. 238–244. DOI: 10.1016/j.nucengdes.2013.08.074.

[5] Ming Ding and Jan Leen Kloosterman. "Thorium utilization in a small long-life HTR. Part II: Seed-and-blanket fuel blocks". In: *Nuclear Engineering and Design* 267 (2014), pp. 245–252. DOI: 10.1016/j.nucengdes.2013.08.076.

[6] Jacques Verrue, Ming Ding, and Jan Leen Kloosterman. "Thorium utilisation in a small long-life HTR. Part III: Composite-rod fuel blocks". In: *Nuclear Engineering and Design* 267 (2014), pp. 253–262. DOI: 10.1016/j.nucengdes.2013.08.075.

[7] Gilbert Melese and Robert Katz. *Thermal and Flow Design of Helium-Cooled Reactors*. American Nuclear Society, 1984.

[8] J. Kópházi, D. Lathouwers, and J.L. Kloosterman. "Development of a Three-Dimensional Time-Dependent Calculation Scheme for Molten Salt Reactors and Validation of the Measurement Data of the Molten Salt Reactor Experiment". In: *Nuclear science and engineering* 163 (2009), pp. 118–131. DOI: 10.13182/NSE163-118.

[9] M Ding, JL Kloosterman, and FJ Wols. "Thermal-hydraulic evaluations of the u-battery for loss of forced-cooling conditions". English. In: *6th International topical meeting on High Temperature Reactor Technology (HTR 2012), Nuclear energy for the future*. Ed. by s.n. N12-016 Paper: HTR2012-6-019; HTR 2012, 6th International topical meeting on High Temperature Reactor Technology, Tokyo, Japan ; Conference date: 28-10-2012 Through 01-11-2012. s.n., 2012, pp. 1–11.

[10] Ming Ding and Jan Leen Kloosterman. "Thermal-hydraulic design and transient evaluation of a small long-life HTR". In: *Nuclear Engineering and Design* 255 (2013), pp. 347–358. DOI: 10.1016/j.nucengdes.2012.11.009.

[11] Brian Boer et al. "Coupled neutronics/thermal hydraulics calculations for High Temperature Reactors with the DALTON-THERMIX code system". In: *International Conference on the Physics of Reactors 2008, PHYSOR 08* 3 (Jan. 2008).

[12] H.E.A van den Akker and R.F Mudde. *Transport phenomena: the art of balancing*. Delft Academic Press, 2014.

[13] V. Gnielinski. “Neue Gleichungen für den Wärme- und den Stoffübergang in turbulent durchströmten Rohren und Kanälen”. In: *Forsch Ing-Wes* 41 (1975), pp. 8–16. DOI: 10.1007/BF02559682.

[14] H. Petersen. *The properties of helium: Density, specific heats, viscosity, and thermal conductivity at pressures from 1 to 100 bar and from room temperature to about 1800 K*. Denmark. Forskningscenter Risoe. Risoe-R 224. Risø National Laboratory, 1970.

[15] Jos van Kan, Guus Segal, and Fred Vermolen. *Numerical Methods in Scientific Computing*. VSSD, 2008.

[16] Béatrice Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations*. Society for Industrial and Applied Mathematics, 2008.

[17] Donald McEligot et al. “Thermal Properties of G-348 Graphite”. In: (May 2016). DOI: 10.2172/1330693. URL: https://www.osti.gov/biblio/1330693.

[18] P. Slingerland and C. Vuik. *Spectral two-level deflation for DG: a preconditioner for CG that does not need symmetry*. Delft University of Technology, 2011.

[19] Grzegorz Zboiński. “Application of the three-dimensional triangular-prism hpq adaptive finite element to plate and shell analysis”. In: *Computers & Structures* 67 (Nov. 1997), pp. 497–514. DOI: 10.1016/S0045-7949(96)00415-4.

[20] Koen Hillewaert. “Development of Discontinuous Galerkin Method For High-Resolution, Large Scale CFD And Accoustics in Industrial Geometries”. PhD thesis. 2013.

[21] Satish Balay et al. *PETSc Users Manual*. Tech. rep. ANL-95/11 - Revision 3.7. Argonne National Laboratory, 2016.

[22] Geuzaine, Christophe and Remacle, Jean-Francois. *Gmsh*. Version 4.8.3. Dec. 1, 2020. URL: http://http://gmsh.info/.

[23] Gordon D Smith, Gordon D Smith, and Gordon Dennis Smith Smith. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985.

[24] James J. Duderstadt and Louis J. Hamilton. *Nuclear reactor analysis*. Wiley and Sons, 1976.

A

Appendix

A.1. General weak form for the time-dependent diffusion equation

Starting from the time-dependent diffusion equation with Neumann boundary condition $g_N = \gamma\,(T(r_s) - T_{He}(z))$:

$$\begin{aligned} -\lambda\nabla\cdot(\nabla T) &= p(r,t)\ , \\ -\lambda\,(\nabla T\cdot\hat{n}) &= g_N,\ \text{on}\ \Gamma_N\ . \end{aligned} \tag{A.1}$$

Substituting T for u, multiplying by v and integrating over an element e_k gives:

$$\rho c_p\int_{e_k}\frac{du}{dt}v - \lambda\int_{e_k}(\nabla\cdot(v\nabla u) - \nabla u\cdot\nabla v) = \int_{e_k} pv\ . \tag{A.2}$$

Applying Gauss's theorem gives the contribution to the weak formulation:

$$\rho c_p\int_{e_k}\frac{du}{dt}v + \lambda\int_{e_k}\nabla u\cdot\nabla v - \int_{de_k} v\nabla u = \int_{e_k} pv\ . \tag{A.3}$$

Substituting u and v for the space-discretized temperature and the basis functions in space and implementing the Neumann boundary condition gives the local matrix on the element e_k:

$$\rho c_p\frac{dT_i}{dt}\int_{e_k} h_i h_j + \lambda\int_{e_k}\nabla h_i\cdot\nabla h_j + \gamma\int_{de_k\in\Gamma_N} h_i h_j T_i = \int_{e_k} p h_j + \gamma\int_{de_k\in\Gamma_N} T_{He} h_j\ . \tag{A.4}$$

Substituting M,L and s_i gives the time-dependent system in equation 3.4.

$$\begin{aligned} M &= \int_{e_k} h_i h_j\ , \\ L &= \lambda\int_{e_k}\nabla h_i\cdot\nabla h_j\ , +\gamma\int_{de_k\in\Gamma_N} h_i h_j \\ s_i &= \int_{e_k} p(r,t)h_i + \gamma\int_{de_k\in\Gamma_N} T_{He}(z,t)h_i\ . \end{aligned} \tag{A.5}$$

A.2. Additional results for coupling verification

Table A.1: Average and maximum solid temperatures and outlet temperature for steady-state simulations on a graphite tube with a single coolant channel. Temperatures are in Kelvin.

				Initial core temp	0					1000				
Mesh	N	N_H	**refinement**	**circle refinement**	**Iter**	T_{max}	T_{avg}	T_{outlet}	**run time**	**Iter**	T_{max}	T_{avg}	T_{outlet}	**time [s]**
1	2140	20	1	7	14	1526.989	1063.679		4.479	13	1527.41	1063.78	1306.53	3.89
2	5600	20	0.5	10	14	1515.89	1056.475	1296.914	15.14	13	1516.308	1056.57	1297.296	12.27
3	30320	20	0.2	20	14	1500.612	1046.54	1284.046	172	13	1501.027	1046.64	1284.277	147.6
4	4280	40	1	7	14	1532.017	1061.514	1306.336	10.26	13	1532.369	1061.592	1306.517	8.721
5	11200	40	0.5	10	14	1520.895	1054.33	1296.93	34.14	13	1521.24	1054.41	1297.123	29.4
6	60640	40	0.2	20	14	1505.58	1044.42	1284.07	355.8	13	1505.92	1044.5	1284.26	316.3
7	8560	80	1	7	14	1533.487	1060.95	1306.342	22.75	13	1533.82	1061.023	1306.51	19.53
8	22400	80	0.5	10	14	1522.36	1053.77	1296.94	73.83	13	1522.69	1053.845	1297.119	55.77
9	121280	80	0.2	20	14	1507.036	1043.87	1284.08	719.2	13	1507.368	1043.945	1284.25	635.3
10	17120	160	1	7						13	1534.21	1060.87	1306.512	41.87
11	44800	160	0.5	10						13	1523.08	1053.701	1297.118	131.8
12	398400	160	0.2	40						13	1502.823	1040.6	1280.118	2424
13	34240	320	1	7						13	1534.317	1060.842	1306.512	91.41
14	89600	320	0.5	10						13	1523.19	1053.67	1297.118	289.6
15	99600	40	0.2	40						13	1500.99	1041.3	1280.12	546.1
16	129680	40	0.2	60						13	1500.11	1040.717	1279.38	751.2