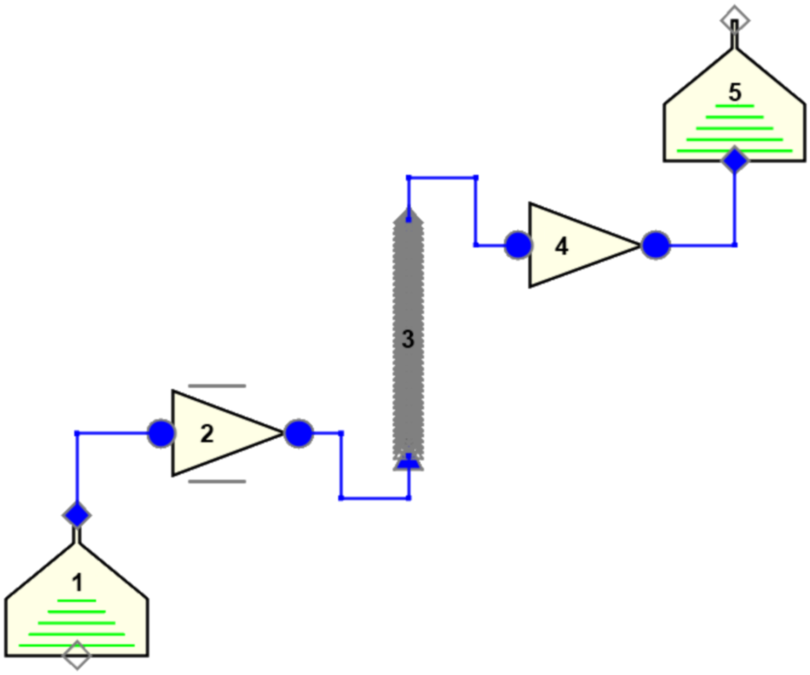
CHALMERS



Modeling of PWR LOCA experiments in RELAP5 based on PREMIUM benchmark

Master's Thesis in Nuclear Engineering

RICKARD MAGNUSSON

Division of Nuclear Engineering
Department of Applied Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2013

CTH-NT-283 ISSN 1653-4662

Thesis for the Master of Science Degree

Modeling of PWR LOCA Experiments in RELAP5 based on PREMIUM Benchmark

Supervisor: Assist. Prof. Tomasz Kozlowski
University of Illinois Urbana-Champaign
Urbana-Champaign, United States of America

Examiners: Prof. Anders Nordlund and Prof. Imre Pázsit
Chalmers University of Technology
Gothenburg, Sweden

RICKARD MAGNUSSON
Division of Nuclear Engineering
Department of Applied Physics
Chalmers University of Technology
Gothenburg, Sweden, 2013

Modeling of PWR LOCA Experiments in RELAP5 based on PREMIUM Benchmark
RICKARD MAGNUSSON

CTH-NT-283
ISSN 1653-4662
Division of Nuclear Engineering
Department of Applied Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden, 2013

The research was performed at
Department of Nuclear, Plasma, and Radiological Engineering
Collage of Engineering
University of Illinois at Urbana-Champaign
216 Talbot Laboratory
104 South Wright Street
Urbana, IL 61801
United States of America

Cover:
RELAP5 model of the FEBA test section

Chalmers Reproservice
Gothenburg, Sweden 2013

Department of Nuclear, Plasma, and Radiological Engineering
Collage of Engineering
University of Illinois at Urbana-Champaign
216 Talbot Laboratory
104 South Wright Street
Urbana, IL 61801
United States of America

Acknowledgements

First of all I would like to thank Prof. Imre Pázsit for making it possible for me to pursue the master's thesis in the United States as he introduced me to Assist. Prof. Tomasz Kozlowski. Imre was also the person who introduced me to nuclear engineering since he gave me the very first opportunity of working in such a field, for my bachelor's thesis.

I would like to express my sincere gratitude to my supervisor, Assist. Prof. Tomasz Kozlowski for letting me come to University of Illinois at Urbana-Champaign, and assigning me such a nice and inspiring project. Without him this project could never have been made.

I would also like to thank the Department of Nuclear, Plasma, and Radiological Engineering at the University of Illinois at Urbana-Champaign for the warm welcome and pleasant time. The project benefited a lot from Rijan Shrestha who helped me with both RELAP5 and DAKOTA. I want to thank my new-found friends (in no specific order): Kuan-Che Lan, Rabie Abu Saleem, Rijan Shrestha, Rui Lin Tan and Xu Wu. These guys made the time much more enjoyable, I hope that we will meet again!

I would like to thank Assist. Prof. Paolo Vinai for his contributions for both the presentation and the final report, this report would not have been the same without him.

Finally, I want to express my gratitude to my uncle Per Nilsson and his family who made the stay in U.S. possible; and to my own family who have supported me throughout all these years.

Abstract

Safety in nuclear power plants is very important since release of radioactive materials to the environment could be extremely dangerous. Thus reliable safety analyses are essential in order to design and operate this kind of systems.

In the past, nuclear power plants were designed according to a conservative approach so that possible uncertainties in the reactor modeling could be taken in account. However, this turned out to give non-realistic analyses and limited operational conditions. In order to obtain more realistic predictions and to improve safely the performance of the reactors, the scientific community addressed huge efforts in the development of best-estimate methodologies and in the investigation/quantification of the uncertainties that can affect reactor calculations.

In this framework the international project PREMIUM is focused on the study of the uncertainties in thermal-hydraulic models that are applied to simulate the final phase of a Loss-of-Coolant Accident (LOCA), namely the core reflooding. Such a work is based on the core reflooding experiments that were carried out in the German test facility FEBA.

In the current thesis, the FEBA experiments are simulated with the thermal-hydraulic program RELAP5. The sensitivity of these calculations is investigated with respect to a group of relevant modeling parameters. To do so, the software DAKOTA was employed.

The RELAP5 model developed for this work provided satisfactory predictions of the behavior of the cladding temperature during the core reflooding phase of a Loss-Of-Coolant Accident. In addition, the sensitivity analysis showed which input parameters can have a relevant impact on the calculated results (e.g., mass flow).

Keywords: Nuclear Power Plants, Safety Analysis, RELAP5, Thermal-hydraulics, Core Reflooding, Sensitivity Analysis, PREMIUM Benchmark.

List of Abbreviations

ANS	American Nuclear Society
BEMUSE	Best-Estimate Methods - Uncertainty and Sensitivity Evaluation
BEPU	Best-Estimate Plus Uncertainty
BWR	Boiling Water Reactor
DAKOTA	Design Analysis Kit for Optimization and Terascale Applications
DBA	Design Basis Accident
ECCS	Emergency Core Cooling System
FEBA	Flooding Experiments with Blocked Arrays
LHS	Latin Hypercube Sampling
LOCA	Loss Of Coolant Accident
LWR	Light Water Reactor
NEA	Nuclear Energy Agency
NRC	Nuclear Regulatory Commission
OECD	Organisation for Economic Co-operation and Development
PREMIUM	Post-BEMUSE Reflood Model Input Uncertainty Methods
PWR	Pressurized Water Reactor
RELAP	Reactor Excursion and Leak Analysis Program
SCRAM	Safety Control Rod Axe Man
SNAP	Symbolic Nuclear Analysis Package
TRACE	TRAC/RELAP Advanced Computational Engine

Meaning of words

DBA	Postulated accident scenario which a nuclear facility must be able to withstand without any release of radioactivity to the environment.
Lower Plenum	The volume within the reactor vessel, below the nuclear core.
Reflooding	Final phase of a LOCA when the emergency core cooling system refills the core with water. The main objective is to keep the cladding below its melting point.
SCRAM	System for rapid insertion of control rods that absorb neutrons and shut down the reactor power.
Temperature quenching	Physical phenomenon such that the temperature of the cladding can be significantly reduced because the water injected by the emergency core cooling system can finally wet the hot and dry fuel rods.
Upper Plenum	The volume within the reactor vessel, above the nuclear core.
Void Fraction	In a vapor/liquid mixture, the volumetric fraction occupied by the gas phase.

List of Figures

List of Tables

Contents

1 INTRODUCTION

This chapter presents a general background on nuclear energy; basics of nuclear safety analysis; an introduction to the loss-of-coolant accident; the purpose and the structure of this thesis.

1.1 Nuclear energy and description of LWRs

Energy is a key issue in our society, in fact the continuously increasing demand combined with the need of preserving our environment requires the development of an energy strategy based on several options [1]. Nuclear power can then play an important role since, for instance, it allows reduction in CO_2 emissions and supports a large-scale production.

Today there are around 430 operable reactors in the world and the production is estimated about 370 GW_e, which roughly corresponds to 13.5% of the total produced electricity [2]. Most of these reactors are Light Water Reactor (LWR) type, i.e. they make use of light water.

Water act as both moderator and as coolant in a LWR. The moderator is used to slow down fast neutrons generated from fission. This is done since the probability to cause fission for ^{235}U is higher when neutrons are in thermal equilibrium with the interacting medium. When ^{235}U is fissioned, two fission products, two-three fast neutrons (prompt neutrons) and energy are released. Most of the fission products are highly unstable, these products will decay to reach a more stable state. Some of them undergo beta decay and, consequently, they emit so-called 'delayed neutrons' (in comparison with the prompt neutrons released immediately in the fission process). Such delayed neutrons are typically generated in the time-scale between milliseconds to minutes after the fissions occur.

The power of a nuclear reactor is proportional to the number of fission, and, thus, it is also proportional to neutron flux (i.e., the number of neutrons per unit area per second). The control of the neutron flux is therefore crucial to avoid power excursions that could damage the reactor core. In the light of this, delayed neutrons play an important role since they make the power response of the reactor relatively slow.

The two most common commercial LWRs are the Boiling Water Reactor (BWR) and the Pressurized Water Reactor (PWR) as they were developed and designed in the western countries.

1.1.1 Boiling water reactor

A BWR [3] consists of one circuit in which the heating process and the electricity producing system is in the same loop, as seen in Figure (1.1). In a BWR water is boiled in the reactor core at about 70 bar. Then the steam from the exit of the core goes through steam separators and dryers so that dry steam is achieved for the turbine systems. Due to lack of space in the upper parts of the core, control rods are entered from the bottom. The fuel for a BWR consists of approximately 800 assemblies with roughly 100 fuel rods in each assembly.

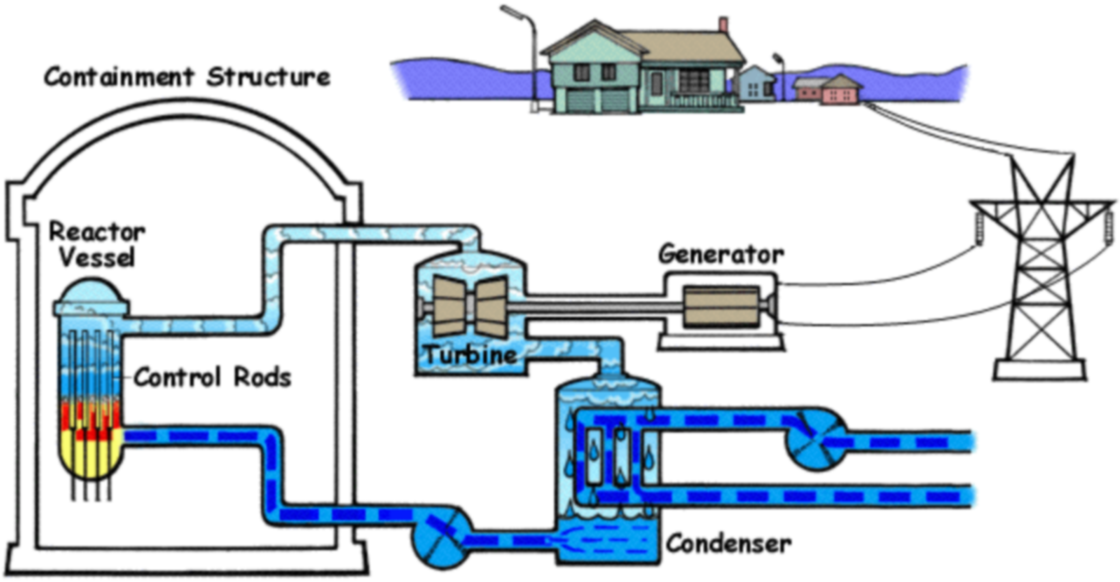


Figure 1.1: Schematics of a BWR [4].

1.1.2 Pressurized water reactor

A PWR [3] is currently the most common reactor type in the world and the schematics can be seen in Figure (1.2). A PWR consists of two circuits. In the primary circuit, water flows through the core at a pressure of about 155 bar without any boiling. The high pressure is maintained by a pressurizer. The heated water of the primary system is used to produce steam for the turbines in a secondary circuit that is at a much lower pressure (around 68 bar). The heat transfer from the primary to the secondary system takes place in heat exchangers, the so-called steam generators. In the upper parts of steam generator on the secondary circuit the steam is also dried before entering the turbines. The core for a PWR is usually smaller and the control rods are entered from the top to the bottom. The fuel for a PWR consists of approximately 150-250 assemblies with 200-300 rods each.

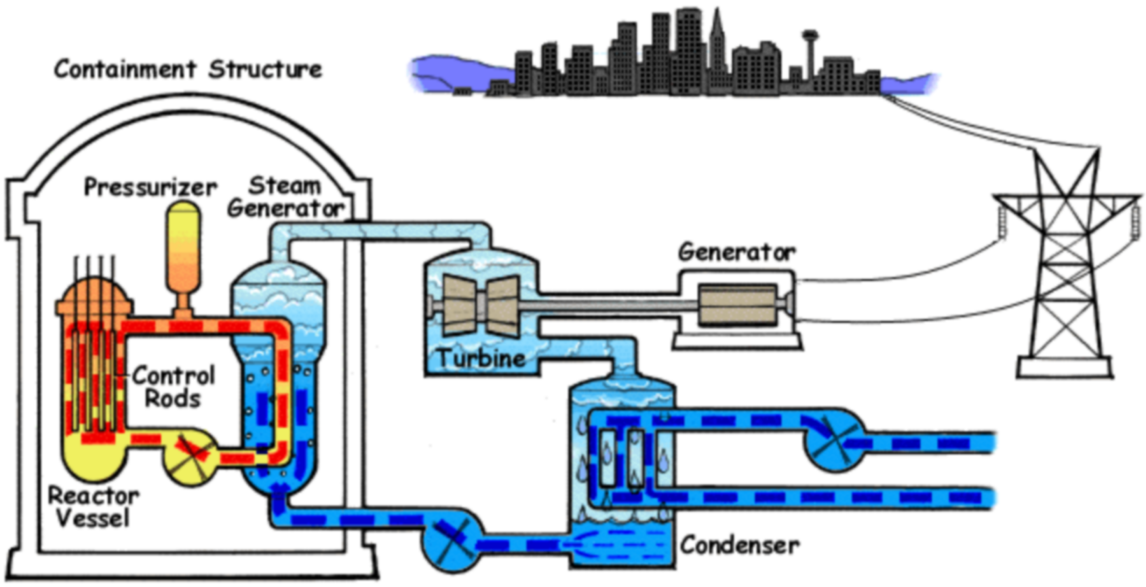


Figure 1.2: Schematics of a PWR [5].

1.2 Nuclear safety analysis

Nuclear safety is a crucial aspect because nuclear power plants contain a large dangerous radioactive inventory. Therefore it is necessary to minimize possible failures, and, in case of accident, to reduce and mitigate the consequences. In this context, reliable calculations able to predict the behavior of the reactor under transient and accidental conditions are essential so that the plant can be demonstrated to meet the safety criteria.

In order to study the safety of a nuclear power plant, to understand if a nuclear power plant can withstand the adverse conditions, or not, a number of accidents are selected and analyzed. These accidents are defined as design basis accidents (DBAs) because they are taken in account during the design phase of the plant. The most important of these DBAs is the Loss-Of-Coolant Accident (LOCA), i.e. the core cannot be cooled properly because a leakage of coolant and an emergency core cooling system is need to reintegrate proper cooling conditions in the core. This is further explained in Section (1.4).

In the past nuclear power plants were designed and constructed based on a conservative approach. Accordingly, pessimistic assumptions were used to compensate the possible uncertainties due to the limited knowledge and the simplified simulation models. However, this kind of strategy could give unrealistic (even un-physical) results which turned out in plant performances much below the real potential. In the last decades, huge efforts have been spent to develop and apply best-estimate methodologies. This approach is characterized by a more realistic way to model all the physical processes of a nuclear power plant. Therefore, this kind of calculations can predict the behavior of the system much closer, and support more sound safety analyses. However, if no conservatism is applied, it becomes crucial to rigorously evaluate the impact of the possible uncertainties on the best-estimate simulations (Best-Estimate Plus Uncertainties, BEPU).

1.3 Research projects in Nuclear Safety Analysis

Recent international projects as the BEMUSE (Best-Estimate Methods Uncertainty and Sensitivity Evaluation) benchmark showed that methodologies and understanding of uncertainty sources in thermal-hydraulic calculations for nuclear safety have reached a good level of maturity [6].

Nevertheless, questions are still open and one of the issues that must be addressed is how uncertainty in input and models parameters can be quantified. This aspect is important because it directly affects the uncertainty in the final results that are used for safety evaluations. From this need, a new international effort has been started, i.e. the PREMIUM (post-BEMUSE reflood model input uncertainty methods) benchmark [6, 7].

The PREMIUM benchmark aims at evaluating models uncertainties for the simulation of LOCA scenarios, and, in particular, for the reflooding phase during a LOCA. The current thesis is a contribution to the PREMIUM benchmark whose detailed description can be found in Section (2.1).

1.4 Large loss of coolant accident in a PWR

As mentioned in the previous sections, a significant part of the research in nuclear safety has been related to the study of Loss-Of-Coolant Accident (LOCA).

A LOCA takes place when a failure in the primary system leads to a loss of primary coolant, and the proper cooling of the reactor is disrupted. The scenario ranges from small leakages to massive fracture of the reactor vessel [8]. One of the most severe conditions that the reactor could experience is when a large break occurs in one of the main pipes of the primary loop.

In the case of a LOCA, the fuel cladding temperature is the key parameter because it is directly connected to the possible degradation of the fuel rods: if very high temperature are reached in the core because of the loss of primary coolant, then fuel rods can melt down and radioactive fission products can be released.

Although the design and the manufacturing of the reactor is such that the probability of a LOCA is very low, it is important to show that the plant safely can deal with the scenario. This means that the reactor core can be shut down and the emergency core cooling system is able to effectively decrease the core temperature before any core melting happens.

For the analysis of these conditions, thermal-hydraulic codes together with thermal-hydraulic experiments are used.

Since the experiments in the PREMIUM benchmark are carried out for PWR conditions, the focus of the discussion in this thesis will be on PWR LOCA.

In normal operations, the pressure in the primary system of a PWR is approximately 155 bar, and the water is in sub-cooled conditions (boiling is not allowed). When there is a rupture of one of the main pipes connected to the reactor vessel, the primary circuit and the core experience a loss of coolant as well as a rapid and strong depressurization [8]. The latter causes water to reach the saturation conditions, and boiling can start.

This lead to a quick voiding of the core. On one hand, the power is reduced to the level of the "decay heat" because of the very low density of the moderator left in the core and also due to the scram of the reactor. On the other hand, a little amount of coolant is in the core, and the heat transfer is very low. The result is such that the cladding temperature dramatically increases.

The plant need to be designed so that the peak cladding temperature can be limited below about 1200°C to prevent extensive metal-water reactions. In fact this kind of chemical reactions are dangerous because then the cladding could get brittle and easy to fail.

To stop the increase of the cladding temperature, an Emergency Core Cooling System (ECCS) that is implemented in the plant is activated. Such a system will deliver water to the core despite the break, and it should be able to cool the hot fuel elements and to decrease their temperature. This phase of the scenario is called 'core reflooding'.

The reflooding looks different depending on which type of LWR is considered. A BWR emergency core cooling system adds water from top to bottom in the core, while the PWR does the opposite. Since the experiment was based on data from a PWR the reflood will happen from the bottom and upwards.

As water is added from the bottom it cannot cool the cladding since a film of vapor is covering the fuel elements. But after a while, by continuing to add water, the fuel rods will be quenched and this quenched front will continue to move upwards as more and more water will be added. This effect of the quenched front lagging behind the water front is known as *hysteresis* which is Greek for "lagging behind" [9].

The void fraction, α_g, in the top part of the core is initially close to 1.0 (i.e. vapor is found to be almost dry) [9]. While the water front advance upwards along the channel and quench the cladding temperature, the void fraction decreases. The cladding temperature reaches its peak value in the upper region of the core because of the injection of the emergency coolant from the bottom. The completion of the reflood phase is achieved when the core is refilled and the fuel is fully quenched. The quenching phenomenon along with the behavior of the cladding temperature is discussed in details throughout the next chapters of this thesis (see Subsections (2.1.3) and (4.1.1)).

In the current work, the analysis is ended as the reactor is successfully flooded. However, in a real case, sufficient cooling must be ensured over a long term perspective in order to properly handle the heat due to the decay of the fission products.

1.5 Purpose and structure of the thesis

The current master of science thesis aims at evaluating a RELAP5 model for the simulation of reflooding experiments available in the PREMIUM benchmark. In particular, such a work includes a sensitivity study that quantifies the impact of some key input parameters on the prediction of the cladding temperature during the experiments.

The document is structured as follows:

- in Chapter 2 relevant information to the PREMIUM project is described plus the tools used to perform the analysis for this project;
- in Chapter 3 the methodology is described ;
- in Chapter 4 the results are presented and discussed;
- in Chapter 5 the work is summarized and conclusions are drawn.

2 THEORY

This chapter provides background information for the current work.

First, the PREMIUM project is discussed in Section (2.1). The data from this benchmark will be used to validate the model developed for the RELAP5 code, for simulating PWR reflooding experiments.

The approach followed to assess the variability of the simulations with respect to the variability of the input is introduced in Section (2.2).

The code RELAP5, the tool SNAP for processing the RELAP5 input and output files, and the software DAKOTA for sensitivity analysis are described in Subsections (2.3.1) to (2.3.3), respectively.

2.1 The PREMIUM benchmark

This section gives a description of what PREMIUM benchmark is, the objectives of the benchmark and also how the benchmark is structured. Thereafter the test facility and the results from one reflooding experiment are presented.

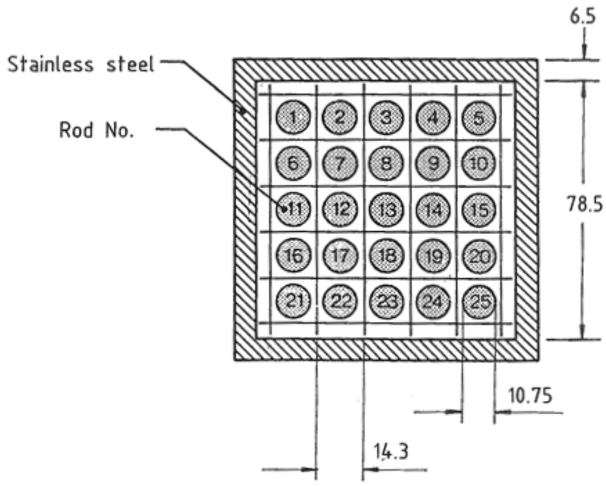
2.1.1 Description of the benchmark

PREMIUM is an international project assembled by OECD/NEA where the scope of the benchmark is to analyze the uncertainties in input parameters and physical models for thermal-hydraulic simulations [10]. The preferred way to model these uncertainties is by using experimental data and then comparing this with the prediction of the thermal-hydraulic codes. To do so, for the PREMIUM benchmark, reflooding experiments performed in the German facility FEBA were considered (see Section (2.1.2)).

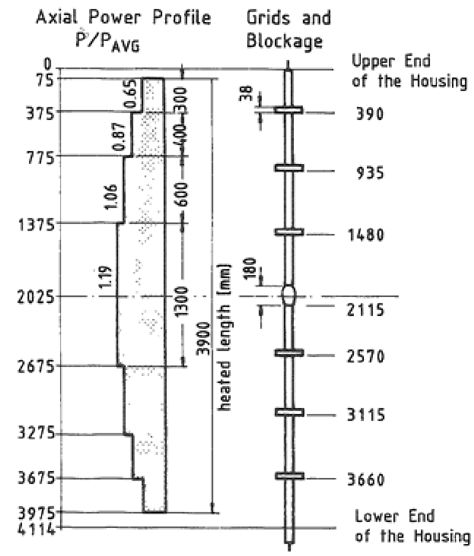
The first step of the project was to determine the influence of different input and model parameters on the prediction of the cladding temperature. In view of this, the participants in the benchmark could choose a thermal-hydraulic system code (e.g., RELAP5 or TRACE), and they were asked to assemble a list of input and modeling parameters whose influence on the results is significant according to specific criteria (further details are discussed in Section (3.5)). Then, the uncertainties of the most influencial parameters were quantified, and their impact were studied with respect to the simulations.

2.1.2 FEBA Facility

The experimental database employed in the benchmark is based on the experiments carried out in the FEBA facility that was operated at KfK (Karlsruhe, Germany). This test facility is designed to perform reflooding experiments with the possibility to maintain constant flooding rates as well as back pressure. The test section used consists of a full length 5 x 5 rod sub-bundle of a typical PWR as shown in Figure (2.1a). The fuel rods, however, are electrically heated with an approximated cosine axial power profile which can be seen in Figure (2.1b).



(a) FEBA model - cross section view



(b) Axial power profile over a heater rod

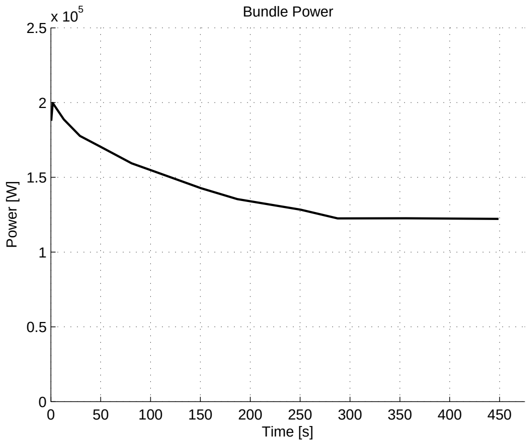
Figure 2.1: Schematics over the FEBA testing model [7].

The Figure (2.1a) shows the cross sectional view of the bundle. The quadratic shaped housing consists of stainless steel, the outer side length is 91.5 mm and the thickness of the housing is 6.5 mm. This leaves a quadratic area of 78.5x78.5 mm^2 for the 5 x 5 rod bundle, where each heated rod got a diameter of 10.75 mm and a heated length of 3900 mm. This also corresponds to a hydraulic diameter, h_D, of the bundle array equal to 13.44 mm. The thick wall was chosen to prevent premature quenching of the wall relative to the bundle quench front progression. The other Figure (2.1b) shows the axial power profile which is a symmetric approximation of the cosine profile consisting of seven steps as well as the axial dimensions of the rods. The seven step approximated power profile consists of a power ratio (P/P_{avg}) from 0.65 to 1.19 and its largest value occurs in the central region. This Figure also shows the typical PWR height of 4114 mm [7].

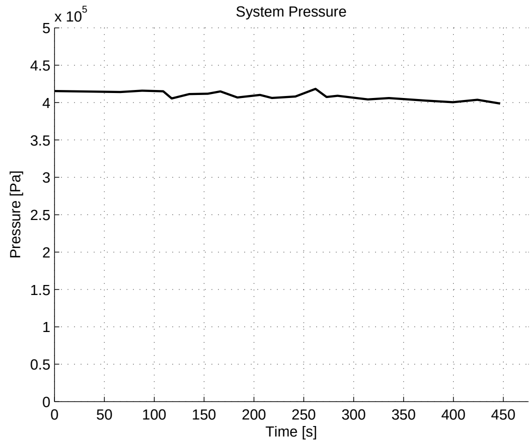
2.1.3 Description of the FEBA core reflooding experiments

In order to establish the proper initial conditions for the reflooding experiment, a preparation had to be done. Accordingly, the test section was filled up with steam, and, in stagnant conditions, the fuel rod simulators were heated up. This also caused the passive heating of the housing because of the radiation heat from the hot rods. In this process, water was circulated so that temperature could be maintained at the desired value as well as the inlet plenum could be kept cooled. In addition, the bundle power was set in such a way that decay heat for a real nuclear bundle could be reproduced according to 120% ANS standard.

Power and boundary conditions for the experiment are displayed in Figure (2.2) and Figure (2.3). The figures have been recreated with MATLAB with information from the test section in the PREMIUM report.

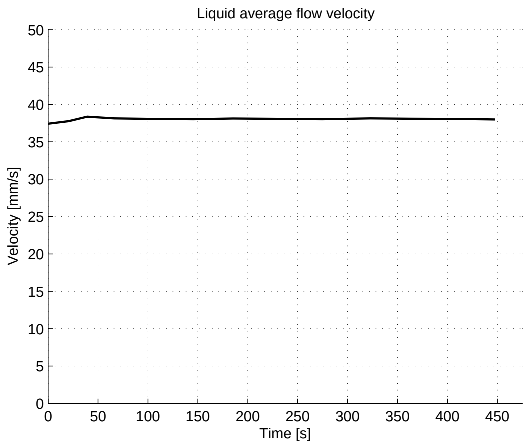


(a) Bundle power

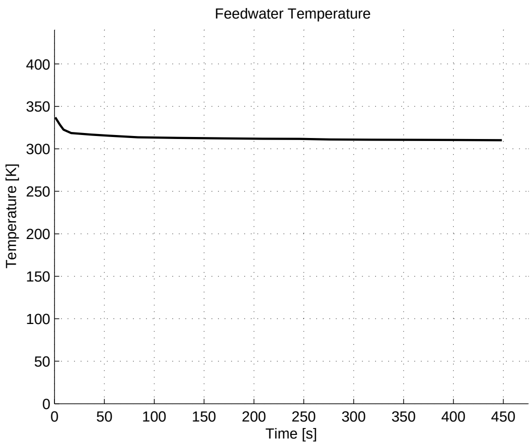


(b) System pressure

Figure 2.2: Power and system pressure.



(a) Flow velocity



(b) Feedwater temperature

Figure 2.3: Inlet velocity and Feedwater temperature.

Figure (2.2a) shows the bundle power. In the beginning of the test the value is around 200 kW, and, in the end of the test it is around 120 kW. The pressure boundary condition can be seen in Figure (2.2b) and it varies between 4.2 bar to approximately 4.0 bar. Other boundary conditions can be seen in Figure (2.3). The third is in Figure (2.3a) and shows the liquid flow rate boundary condition. This value both starts and ends at 38 mm/s but has a maximum at around 38.5 mm/s. The last boundary condition can be seen in Figure (2.3b) which is the feedwater temperature. This parameter starts at around 340 K and decreases to approximately 310 K.

The test facility was equipped with thermocouples for measuring the cladding temperature at several axial locations along the bundle. The measurements for one experiment are shown in Figure (2.4).

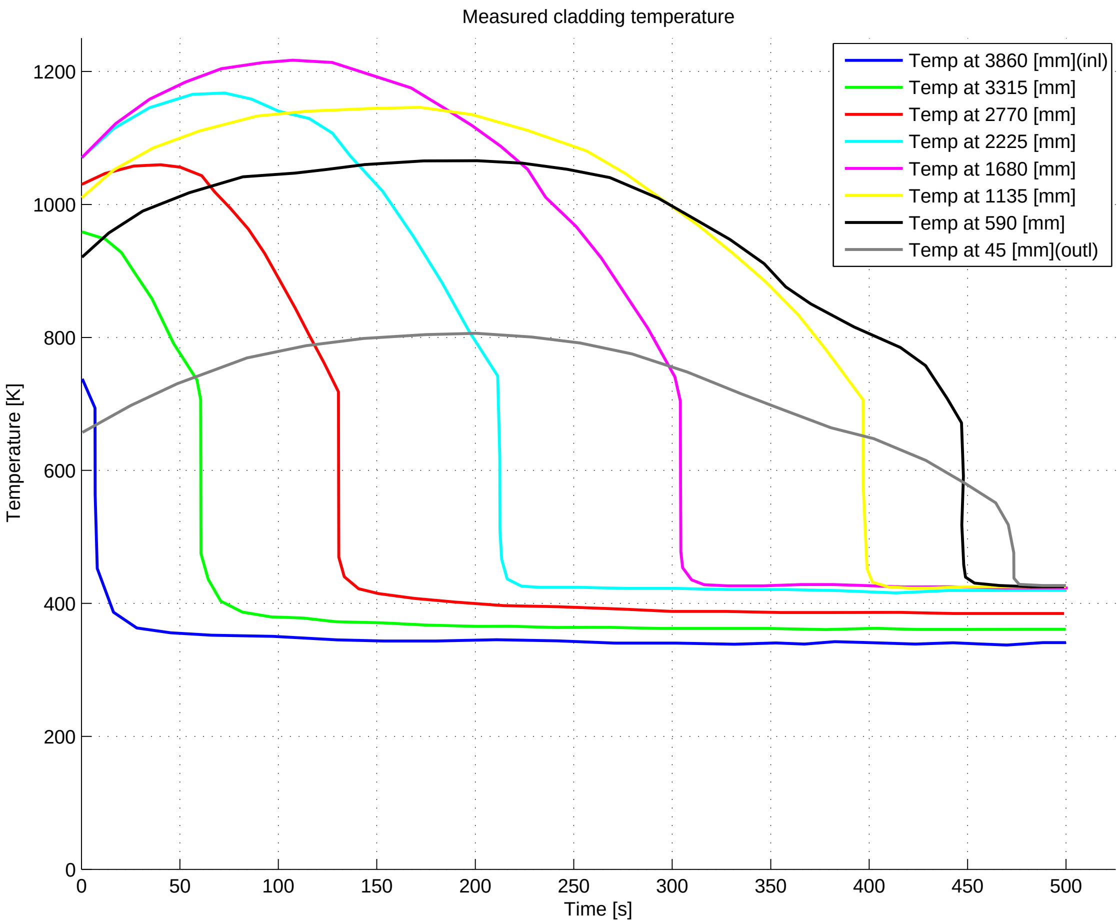


Figure 2.4: Measured cladding temperature.

It can be seen how the temperature varies with respect time, at different elevations in the test section (see also Figure (2.1b)). The lower part of the heated rod is quenched first since the water flow is delivered from the bottom. For instance, the temperature in the lowest position starts to decrease just after few seconds from the beginning of the transient (see the blue line in Figure (2.4)). The maximum temperature is reached at about middle-elevation because of the cosine shape of the axial power and because of the time necessary for the quenching front to travel from the bottom (see the magenta line in Figure (2.4)). Although the increase of temperature for the upper positions is lower than the one for the middle positions (again, due to the cosine power profile), it lasts longer since it takes time for the water to fill up the channel and quench the top as well (compare, for instance, the black and magenta lines in Figure (2.4)).

2.2 Sensitivity study

Sensitivity studies for code simulations aims at identifying the most important input and modeling parameters that affect the results. To do so, a systematic variation of those parameters is performed and the impact of such a variation on the calculations is assessed.

A method that is applied for the purpose of sensitivity analysis is based on a statistical approach and it is displayed in Figure (2.5). Given one or more input or modeling parameters, one can select a range of values over which each parameter can vary with respect to a probability distribution function. According to this information, a sample of possible values for the parameters is estimated by applying a Monte Carlo (MC) or a Latin Hypercube Sampling (LHS) method. For each set of values of the parameters from the sample, a calculation is then performed. Therefore, a set of results is calculated for the values of the sample generated for the input or modeling parameters. The variability of the result with respect to the variability of the input or modeling parameters can be investigated by making use of correlation coefficients and/or regression models. This analysis gives a measure of the relative importance of each parameter on the calculation as well as the kind of relationship between outputs and inputs.

There are a variety of correlation coefficients, as the Partial Rank Correlation Coefficients (PRCC). The results from the PRCC ranges between -1 and 1. The coefficient for a certain input or modeling parameter is close to 1 or -1 whenever a strong linear correlation exists between that parameter and the output of the calculation. A negative value means that the correlation is inversely proportional, whereas a positive value indicates a direct proportional relationship between input and output.

2.3 Software used in this thesis

This section provides information about the software applied to perform the work, as RELAP5, SNAP and DAKOTA. MATLAB was also used for some post-processing of the data. Because MATLAB is a relatively standard program employed in engineering analysis, the reader may be directly referred to the information available on the official website: http://www.mathworks.se/products/matlab/.

RELAP5 is the thermal-hydraulic program used in this thesis to perform the reflooding simulations. SNAP was used as a tool to create the model of the FEBA facility in RELAP5. Thereafter a sensitivity study was needed and for this purpose DAKOTA was employed since its capability to manage multiple runs from any codes, in an automatic manner.

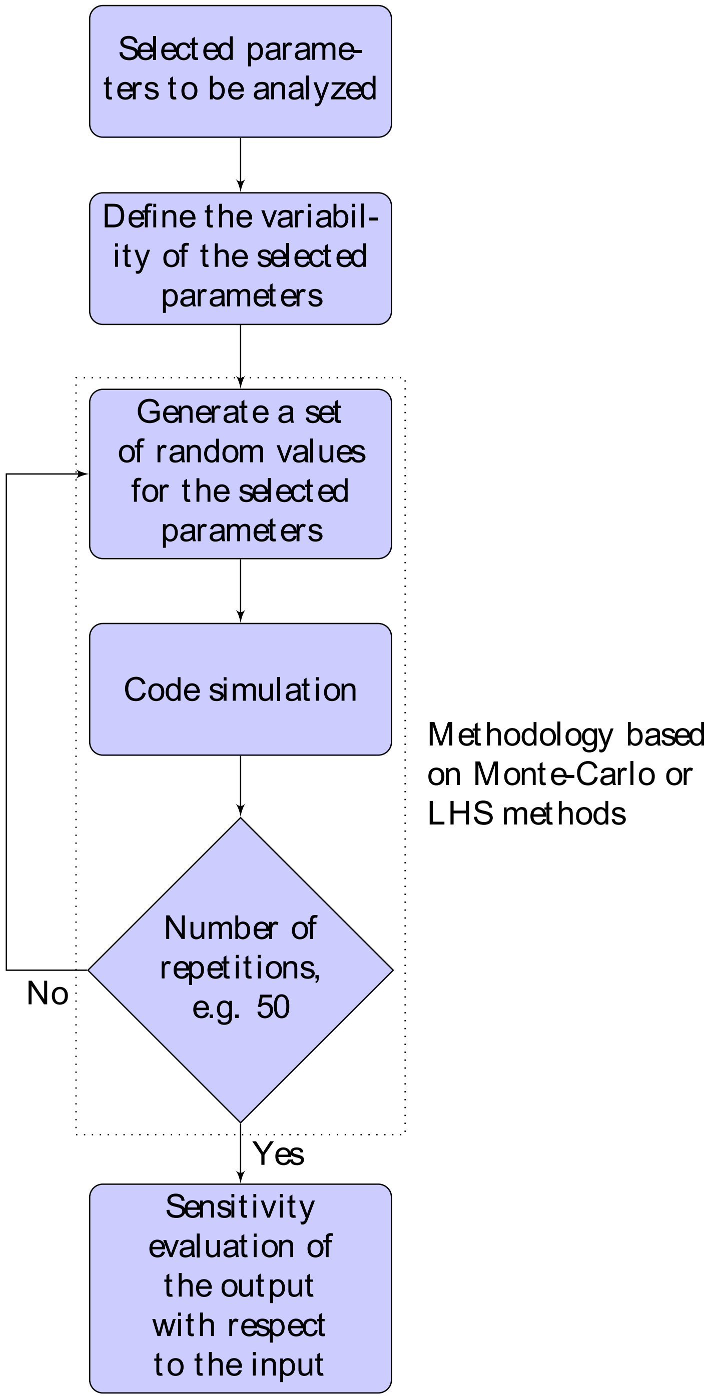


Figure 2.5: Flow chart of a sensitivity analysis.

2.3.1 RELAP5

The BE thermal-hydraulic system code used in this project is RELAP5. The program was originally developed by Idaho National Laboratory, and supported by the United States Nuclear Regulatory Commission (U.S. NRC) [11]. The program is suitable for the analysis of transients and postulated accidents in LWRs, as loss of coolant, anticipated transients without scrams, turbine trip, loss of feedwater and other scenarios. The code is based on a nonequilibrium and nonhomogeneous model for two-phase flow allowing to have unequal temperatures and velocities of the vapor and liquid phase. This system of equations is solved with a semi-implicit finite difference numerical scheme.

2.3.1.1 Hydrodynamic component and heat structures

A number of different hydrodynamic components are available in RELAP5, so several parts of a nuclear power plant can be modeled properly. In the framework of this thesis, only a few components were needed:

- Time-dependent volume, TMDPVOL: it is used to provide sets of time-dependent quantities, as temperature and pressure. Therefore this component is particularly suitable for specifying boundary conditions that can vary with respect to time.
- Time-dependent junction, TMDPJUN: it can be used to define mass flow or velocity of the liquid and/or the vapor as functions of time, from a component to another one. This type pf component is usually necessary to describe how the mass flow (or the velocity) changes at the boundaries of the system under transient conditions.
- Pipe, PIPE: it allows to simulate the flow through a hydraulic channel, and it consists of a sequence of control volumes. The adjacent control volumes are connected with internal junctions. Therefore, if the number of control volumes (cells) is NV (with NV greater than zero), then the number of internal junctions is NV-1. To define a pipe it is necessary to specify the geometrical data (cell length, cross-section area, volume), orientation angles, wall roughness and loss coefficients, as well as the initial conditions of the flow within the pipe.
- Single junction, SNGLJUN: it is similar to the time-dependent junction, but mass flow and velocity are constant in time.

In addition to the hydraulic components there are heat structures employed to model the heat flux [11]. Heat structures are used to calculate the heat transferred across heated solid regions, eventually to the fluid flowing through hydrodynamic components. It can be connected to a single or a pair of hydrodynamic volumes. It can also be used to simulate heat sources as fuel pins or electrically heated plates. To calculate the temperature distribution and heat transfer RELAP5 uses a 1-dimensional rectangular, cylindrical or spherical heat conduction model. To fully define the heat structure, one also have to specify the thermo-physical properties such as thermal conductivity and heat capacity.

2.3.1.2 Input Data Requirements

The input to RELAP5 consists of input records or cards which have a length of 80 characters maximum. These cards are grouped into:

- Title cards
- Comment cards
- Data cards
- Continuation Cards
- Terminator cards

Each hydrodynamic component is identified with a unique ID number CCC (with possible values between 1 and 999) that corresponds to the first 3 digits of the relative data cards [12]. As regards the case of a heat structure, the ID number looks like 1CCC. The interested reader is referred to the input manual for further details.

2.3.2 SNAP

SNAP (Symbolic Nuclear Analysis Package) has been developed for NRC codes and currently includes support for CONTAIN, COBRA, FRAPCON-3, MELCOR, PARCS, RADTRAD, RELAP5 and TRACE [13]. It is designed to simplify the process of performing analysis and one can build input models, manage code runs and process output files. It can be run on a local machine or on a calculation server. The applications included in SNAP are Model Editor, Job Status and Configuration Tool.

2.3.2.1 Model Editor

Model Editor is the primary user interface to SNAP and is used for the development and modification of the input models for the supported analysis codes. Model Editor can handle the design of job streams and also the result animation which is done in AptPlot.

Figure (2.6) shows how Model Editor 2.2.2 looks like. The toolbar to the left gives access to, for example, basic file and model operations, memory usage, etc.; on the right, the command for copy, cut, selection tool, and paste are available. The navigator presents a hierarchical representation of components. In this specific case the navigator field also indicates that the model is a RELAP5-type one. Model view shows the design model. The property region is where one can view and edit the properties of the model. The Message window displays warnings, errors, alerts and notice messages.

2.3.2.2 Configuration Tool

Configuration Tool links the code and the plotting tool to SNAP. It is also used for startup, shutdown and configuration of the local calculation server. Figure (2.7) is a print screen of SNAP Configuration Tool. On the left a list of the linked applications appears. On the right information of the marked applications (in the current case, RELAP5) is reported.

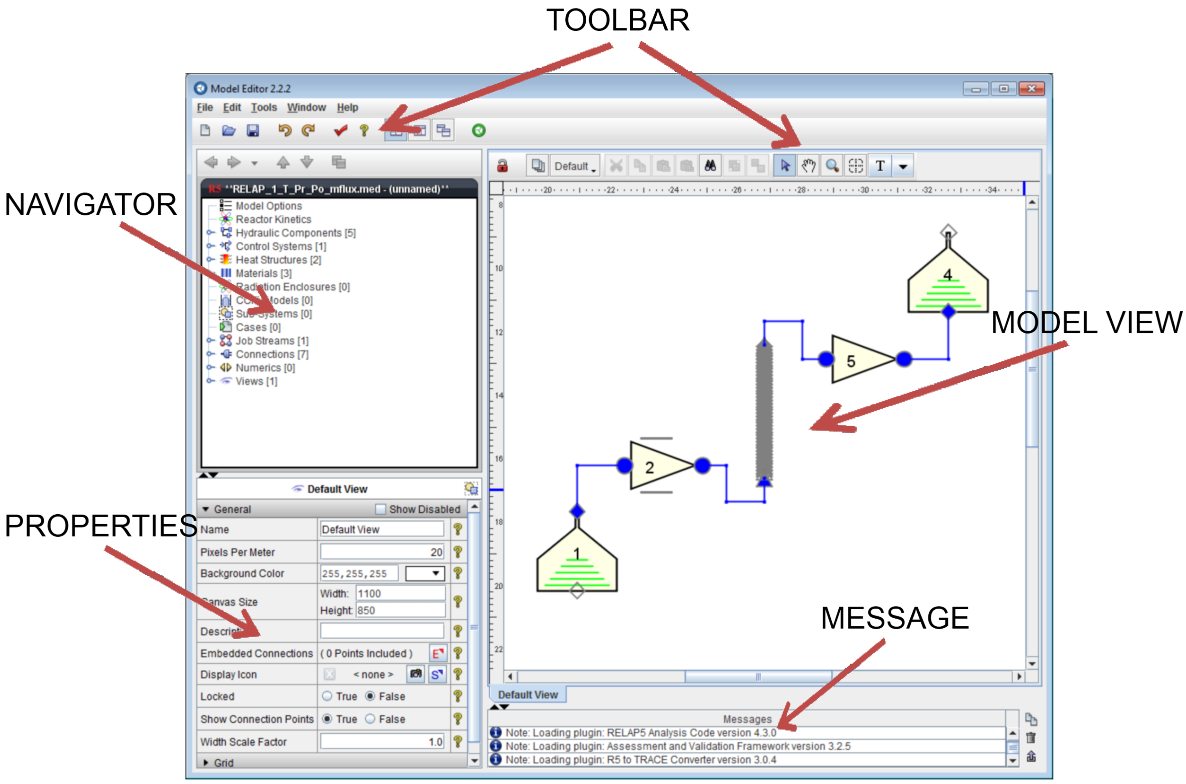


Figure 2.6: Schematic image over Model Editor.

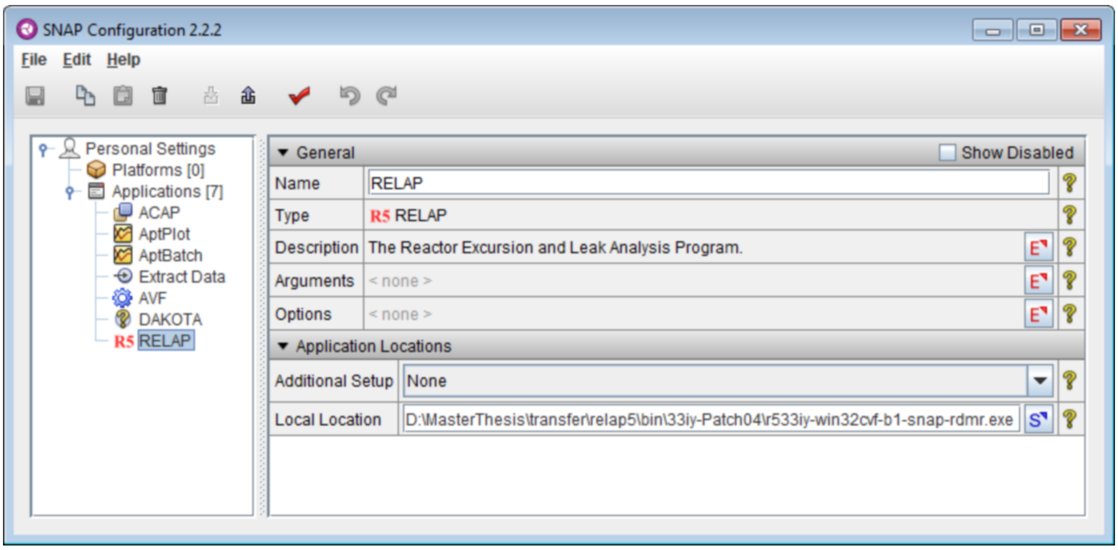


Figure 2.7: Schematic image over Configuration Tool.

2.3.2.3 Job Status

When a job is submitted in Model Editor, then the progression of the work can be monitored with Job Status.

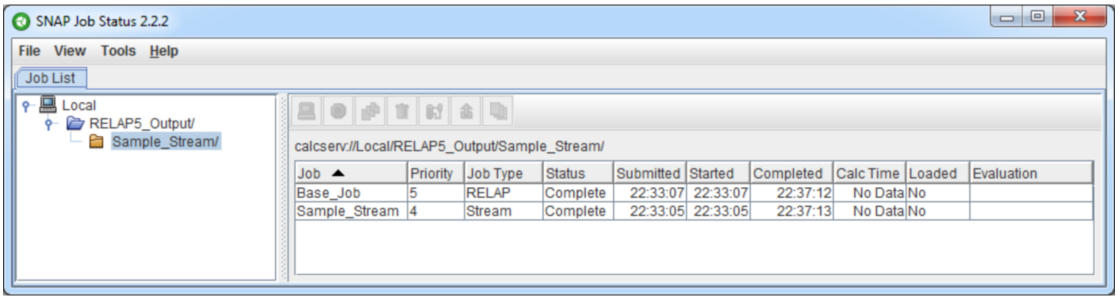


Figure 2.8: Schematic image over SNAP's application, Job Status.

Figure (2.8) shows a print screen on how Job Status looks like when a SNAP run is finished. The left-hand side of the figure lists jobs. The right-hand side in the figure gives information about the type of job performed, when it was started and ended, the total time that was required.

2.3.3 DAKOTA

The DAKOTA project was initiated in 1994, and it has been developed by Sandia National Laboratories in Albuquerque, New Mexico (USA) [14]. In the very early stage of the program, DAKOTA was thought to be used only as optimization tool for structural analysis and design problems. However, further capabilities has been added along the years. Therefore, the application area expanded significantly, and, nowadays, DAKOTA can be applied to computational fluid dynamics, electrical circuits, heat transfer, nonlinear dynamics, shock physics and many other science and engineering models. As many other simulator tools DAKOTA helps to develop a better understanding and also help the predictive capability for complex behaviors in corresponding physical systems.

2.3.3.1 Capabilities of DAKOTA

One both useful and economical feature DAKOTA has is the ability to exploit the computational model in parallel computing which could be applied either on a multiprocessor desktop or a high-performance computer cluster. The principal classes of DAKOTA algorithms [15] are:

- Calibration: How the experimental data best can be modeled.
- Optimization: To ensure the best performing design.
- Sensitivities: To find the most crucial parameter.
- Uncertainties: Determine how safe, reliable or robust the system is.

2.3.3.2 Coupling DAKOTA with a simulator

DAKOTA offers many different iterative methods such as uncertainty quantification through a coupling between DAKOTA and the simulator. If a different iterative method is wanted, only a few changes are needed in the input file [14].

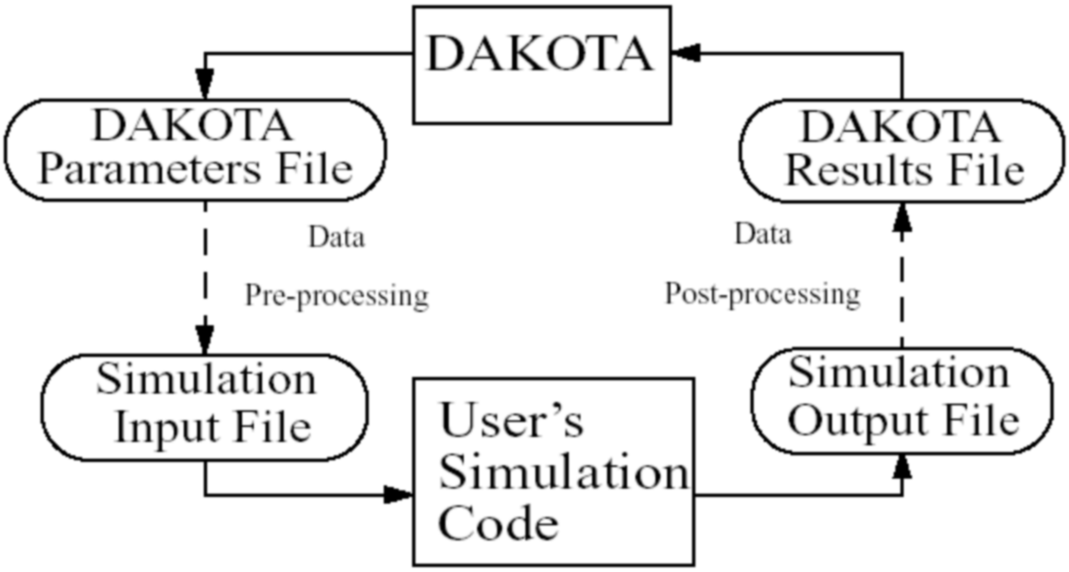


Figure 2.9: Schematic diagram of the coupling between DAKOTA and a generic 'black-box' code for multiple-run problems [14].

Figure (2.9) shows a typical coupling between DAKOTA and the simulation code. Since DAKOTA has very little or no awareness at all of the internal details of the computational model this coupling is often referred to as a "black box". DAKOTA and the simulation code exchange data by reading and writing short data files. The type of analysis that should be done is defined in the text input file to DAKOTA and the options available are described earlier in this subsection, see (2.3.3.1). The solid lines in the figure represent the input/output file operations built-in to DAKOTA or the simulation code. On the other hand, the dotted arrows indicate the transfer or conversion of data that must be managed by routines implemented by the user since these processes depend on the peculiarities of the simulator chosen by the user. A parameter file containing the current variables is written by default as DAKOTA runs. Thereafter DAKOTA starts the simulation code and, when the simulation finishes, it reads the response data from an output file. This procedure is repeated until all the runs required by the iterative study are completed.

3 METHODOLOGY

In the beginning of this work some time was spent to read through manuals to understand the newly installed programs and basically just to get them working. When the programs needed for the thesis were installed, a working RELAP5 model was handed out in Model Editor by Assist. Prof. Tomasz Kozlowski.

In this chapter the RELAP5 model of the FEBA experimental test section is first discussed. Then, the approach followed to study the influence of the input parameters is explained. Thereafter the preparation work for the sensitivity analysis is described, so that one can evaluate the influence of some key parameters on the RELAP5 simulation of the FEBA experiment.

3.1 RELAP5 input model

The RELAP5 model of the FEBA test section is shown in Figure (3.1) and it consists of the following RELAP5 components: time-dependent volume 1, the time dependent junction 2, the pipe 3, a single junction 4, and a volume 5. A heat structure is also included.

The conditions at the inlet are specified with the time-dependent volume 1 and the time-dependent junction 2. The first component gives the changes of pressure and temperature over time, whereas the second one provides the time variation of the mass flow rate. These boundary conditions were defined consistently with Figure (2.3b) and (2.3a).

Pipe 3 together with the related heat structure models the actual PWR fuel rod bundle. The pipe consists of 41 control volumes (the first volume is placed at the bottom). One single heat structure is used to model all the heated rods in such a manner that the correct heat flux is assured. The heat structure has 41 nodes (each node is associated to one of the control volumes of the pipe), and 15 radial points (which takes the actual structure of the rod into account with the gap and the cladding). The data used to define the pipe resemble the same flow area and hydraulic diameter of the real test section.

As regards the outlet conditions, the single junction 4 and the time-dependent volume 5 were used. In particular, the latter serves to define the outlet pressure according to Figure (2.2b).

To get the full model working, in addition to the hydraulic components mentioned above, information on materials, time and time-steps as well as other general data need to be specified. The Model Option part enables SI-units and defines the end-time, minimum and maximum time-steps and other edit and plot frequency options. The time evolution of the power was defined with a general table and corresponds to the one shown in Figure (2.2a). Thermal conductivity and heat

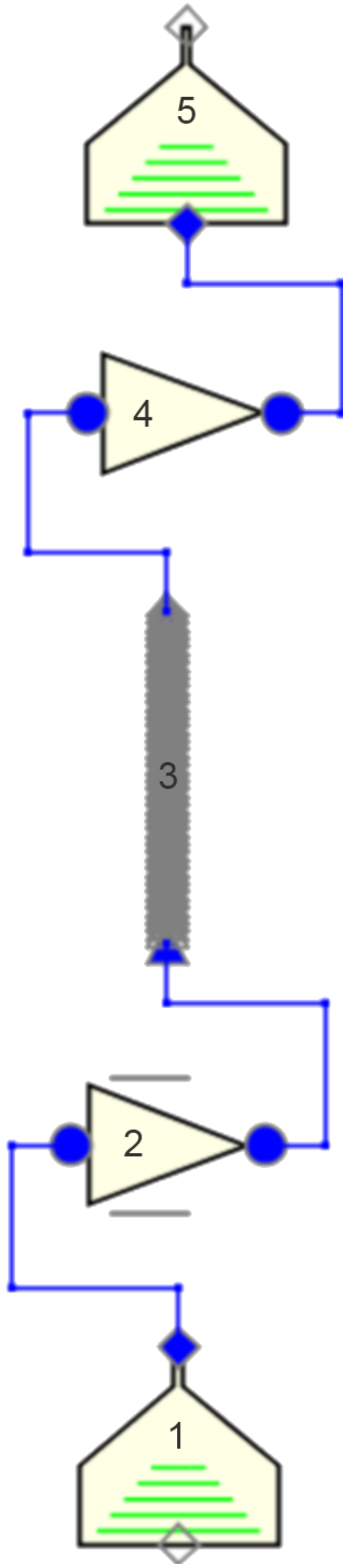


Figure 3.1: RELAP5 model of the FEBA test section.

capacity for the different materials were given at different temperatures by compiling specific RELAP5 tables for the material properties. The interested reader is referred to RELAP5's input manual, [12].

3.2 Inlet mass flow

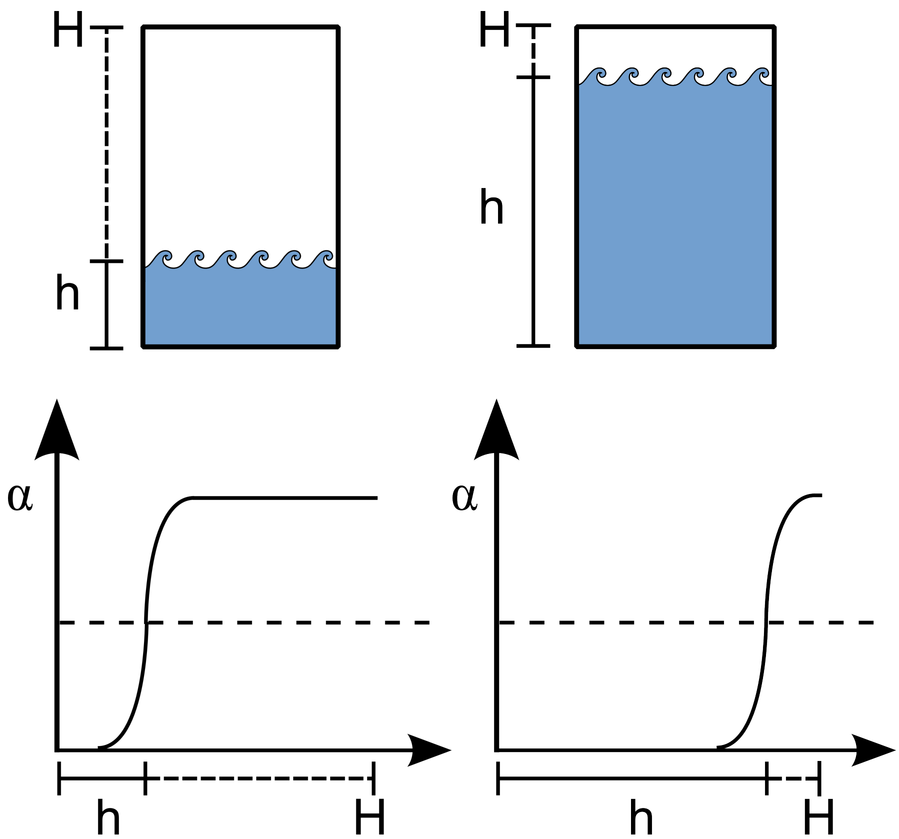
One of the boundary conditions in the PREMIUM report was the liquid flow velocity. However, it is in general more convenient in RELAP5 to use the mass flow since RELAP5 then considers density and area variations. This was mostly done for the sake of learning even though it was not completely necessary.

The idea is to test a mass flow and to derive the related velocity with the following relationship:

$$\dot{m} = \rho v_{flow} A \tag{3.1}$$

where ρ is the density, v_{flow} is the flow velocity and A is the area.

The procedure to determine the flow velocity is illustrated in Figure (3.2). A constant value for the fraction of volume occupied by the gas phase, i.e. void fraction, was used to measure the velocity by knowing when and where it was measured.



(a) at t_1 **(b)** at t_2

Figure 3.2: Simple concept of flooding water.

So, the velocity is calculated when the void fraction, according to the Figure (3.2), is approximately 0.5 at two different heights as:

$$v_{flow} = \frac{\Delta height}{\Delta time} = \frac{h_{x+1} - h_x}{t_{x+1} - t_x} \tag{3.2}$$

This recalculation was performed on the reflood model but with some modifications. For instance the influence of the heated rods cannot be disregarded and therefore the power was put to zero and the heat structure was removed. The results was analyzed with a MATLAB script that simply searched for a specific value of the void fraction for two different elevations preferable around bottom- and top elevation and divided by the time of occurrence. This process was repeated by changing the mass flow in RELAP5 until the desired velocity was achieved.

3.3 Mass error check

In the current work, the mass error related to the RELAP5 simulations was checked carefully. The mass error is due to the fact that the thermal-hydraulic equations implemented in the code are solved numerically, i.e. with some degree of approximation. RELAP5 is built-up in such a manner that if the mass error at a certain time step is beyond a normal limit, then the calculation at that time step is repeated, with a reduced time interval [11].

Although RELAP5 already performs a check on the mass error by default [11], the results calculated for the reflood experiment were examined further in MATLAB. The mass balance of the system can be written as:

$$\underbrace{m_{start}}_{\text{initial mass}} + \underbrace{m_{in}}_{\text{mass flowing in}} - \underbrace{m_{out}}_{\text{mass flowing out}} = \underbrace{m_{current}}_{\text{mass at current time}} + \underbrace{m_{error}}_{\text{cumulative mass error}} \tag{3.3}$$

The initial mass in equation (3.3) is negligible, so it is disregarded. The formula employed for the 'manual' check is:

$$\cancel{m_{start}} + m_{in} - m_{current} - m_{out} = m_{error} \tag{3.4}$$

where m_{in} and m_{out} are taken as the mass change over the inlet and outlet. The m_i is obtained at each cell as:

$$\int_{t_0}^{t_{end}} \dot{m}_{in} dt - \sum_{cells} m_i - \int_{t_0}^{t_{end}} \dot{m}_{out} dt = m_{error} \tag{3.5}$$

where $\dot{m}_{in}$ is the mass flow rate at the inlet, $\dot{m}_{out}$ is the mass flow rate at the outlet and m_i is the mass at cell i. This finally gives:

$$m_{error} = \int_{t_0}^{t_{end}} \dot{m}_{in} dt - \sum_{cells} V_i \left(\alpha_i \rho_{g,i} + (1 - \alpha_i)\rho_{f,i}\right) - \int_{t_0}^{t_{end}} \dot{m}_{out} dt \tag{3.6}$$

and V_i is the volume, α_i is the void fraction, $\rho_{g,i}$ is the density of the vapor and $\rho_{f,i}$ is the density of the liquid.

The results of this analysis are presented in Section (4.1.2).

3.4 Coupling with DAKOTA

As mentioned in Subsection (2.3.3), DAKOTA is an interface that allows to handle different types of problems. For instance, it can be used to run multiple RELAP5 calculations for sensitivity study. In view of this, one can prepare:

- a simulator script
- a RELAP5 input template file (i.e., base/nominal case)
- an option file

The simulator script drives DAKOTA with respect to the actions that need to be taken. The template file, ros.template, contains information about which input parameters of the RELAP5 input file must be modified. The selected input parameters are changed by performing a latin hypercube sampling based on variation ranges and probability distributions specified by the user. For each set of changed values a new RELAP5 input file is created. This procedure is repeated as many times as defined in the option file.

Once the new RELAP5 input file is ready, the instructions in the simulator script are such that DAKOTA will launch the related RELAP5 calculation. When the simulation is finished the output file from RELAP5 is generated. The RELAP5 output files are then post-processed with SNAP so that the information of interest can be extracted and be readable to MATLAB. Further details are explained in Section (3.6).

DAKOTA can also be applied to carry out the sensitivity analysis: a mathematical tool is available for the estimation of the partial rank correlation coefficients (PRCC). As explained in Section (2.2), this type of coefficients are a measure of the possible correlation existing between two parameters: the larger the absolute value is, the stronger the correlation is. The coefficient can vary between -1 and +1, and it is considered significant when its absolute value exceeds 0.5.

3.5 Selection of the influential input parameters

The PREMIUM project determined the influential input parameters in the following way. First of all, participants in the benchmark discussed and compiled a list of influential input parameters based on their expert judgments. To make the selection process more objective and reliable, the input parameters also had to fulfill several criteria with respect to the main reflood responses. From the initial list they kept only those input parameters whose variation caused:

- the absolute value of the cladding temperature to vary more than 50 K
- the rewetting time variation t_{rew} to change at least 10 %
- the elevation variation of the quenching front propagation versus time to be at least 10%.

After the input parameters were tested against the first 3 conditions, additional criteria were applied to ensure the realism of the uncertainty of the parameters:

- the variation of an input parameter should not cause sudden deviation or oscillation in the time behavior of the cladding temperature (i.e., no induced physical or numerical instabilities)
- the range of variation of the input parameters should not be unphysical (e.g., density should not exceed physical limits).

3.6 Output processing

After the RELAP5 calculations are performed, a MATLAB program was used to process the RELAP5 outputs, see Appendix A. MATLAB was employed because it has good capabilities of handling data and creating graphical representation of the results. For instance it could create a video over the cladding temperature distribution throughout the whole model in time. Once the output file was converted into MATLAB all the figures were created.

The data used to create the figures in this project are general quantities, which are printed in the output file as default. RELAP5 prints the more significant quantities in to the output file, whereas other secondary quantities are only given upon user's request. In this manner it is possible to limit the output file to a reasonable size. The interested reader can read about this in the input requirement manual [12]. Table (1) presents quantities that were relevant in this simulations.

Table 1: Example of quantities

Variable	Description
HTTEMP	Mesh point temperature
HTHTC	Heat transfer coefficient
MFLOWJ	Combined liquid and vapor flow rate
P	Volume pressure
RHOF	Liquid density
RHOG	Vapor density
VOIDG	Void fraction

In the current case, one of the most interesting quantity is the temperature httemp calculated in the radial outermost position of the heat structure that corresponds to the cladding temperature of the fuel rods.

4 RESULTS AND DISCUSSION

In the current chapter the results obtained with the methodology described previously are presented. First, the RELAP5 model developed for the FEBA test section is assessed against the measurements from the reflooding experiment. Thereafter the related sensitivity study is discussed.

4.1 Simulation of the FEBA experiment

As explained in Section (1.4), one of the most important parameters that is considered in a LOCA scenario is the cladding temperature. The maximum value reached by such a parameter during a LOCA should be below a certain safety limit in order to avoid fuel failures. It is then very important that nuclear system codes are able to predict such a variable in an accurate manner.

In this section the cladding temperature calculated with the model reported in Section (3.1), for the reflooding experiment carried out in the FEBA facility (Subsection (2.1.2) and (2.1.3)) is shown together with the experimental measurements.

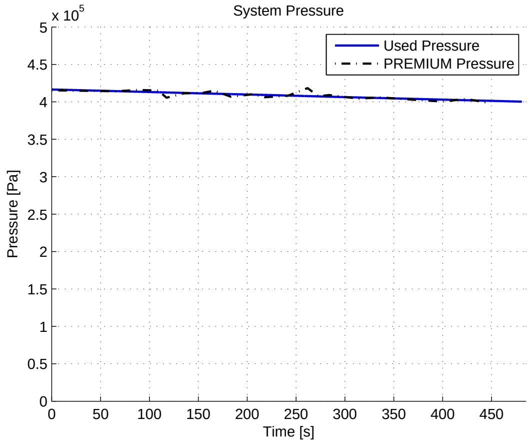
In addition, results for the mass error check are also included in the discussion.

4.1.1 Nominal calculation

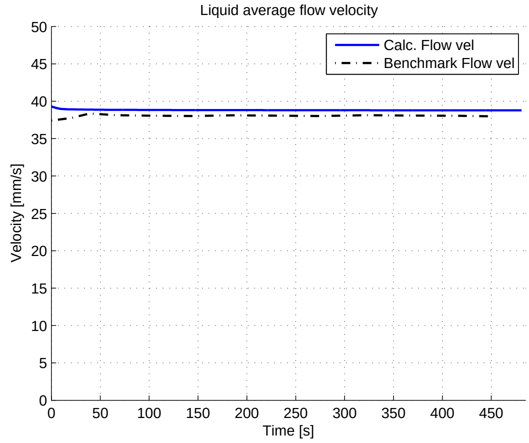
In order to predict the FEBA experiments correctly, the bundle power and the boundary conditions discussed in Subsection (2.1.3) have to be provided to RELAP5. As seen in Figure (4.1)a-b, the system pressure and the inlet velocity were reproduced in RELAP5 close to the experimental values. In the case of the system pressure, the experimental values were approximated with a linear fitting, so small pressure oscillations were disregarded. Following the procedure described in Section (3.2), the inlet mass flow (that was used instead of the inlet velocity) was set equal to 0.15 kg/s. Some deviation between the RELAP5 and the experimental values can be seen, and they may impact the accuracy of the calculations.

Figure (4.1c) shows the cladding temperature calculated with RELAP5 with respect to the elevations along the test section. It can be seen that the qualitative behavior is reproduced fairly good. As expected from the physics of the experiment (see Section (2.1.3)), the temperature is quenched with time at higher and higher elevations as the water front is propagating upward. The maximum temperature is reached around the middle elevation, where the power peak occurs according to the cosine axial profile.

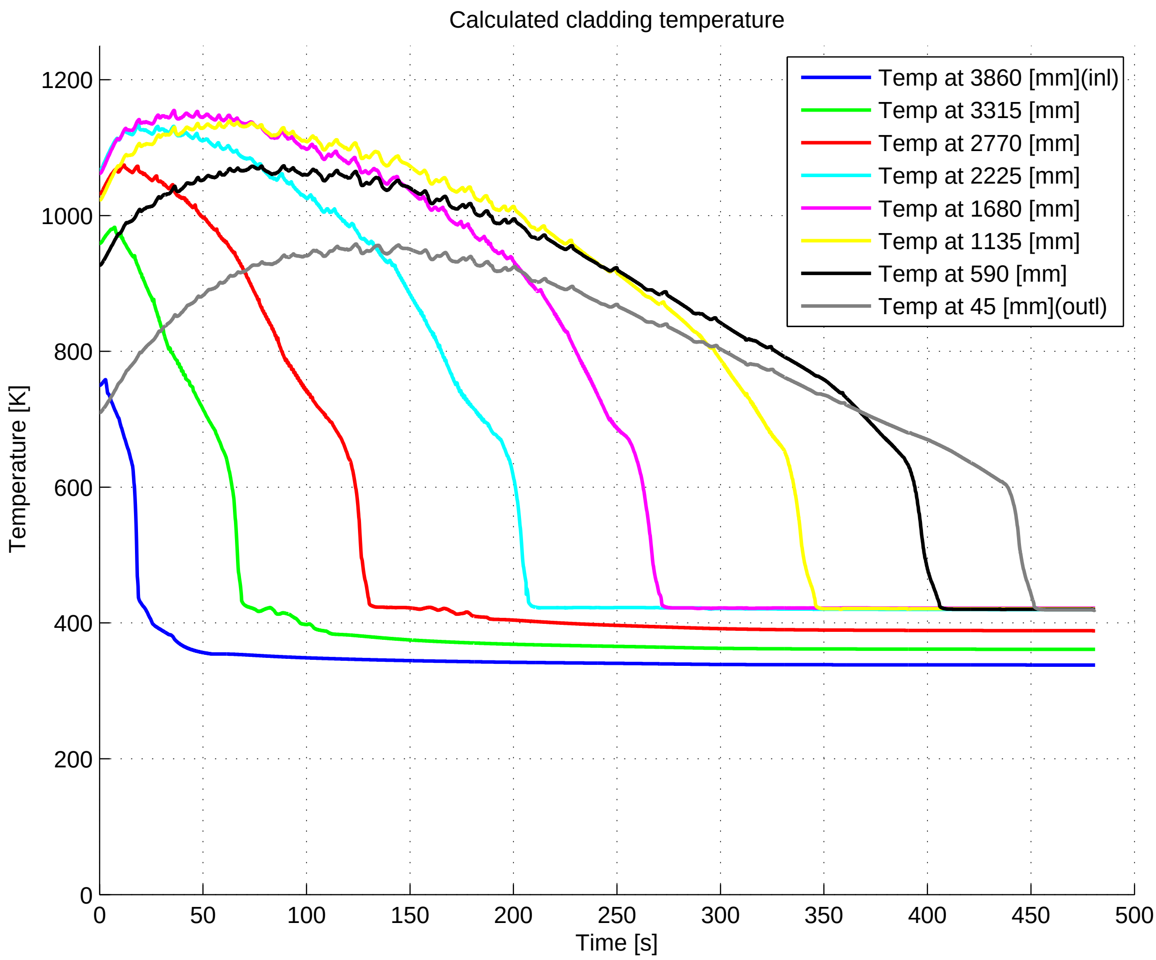
If the calculated temperature is compared with the measurements, one can see how the performance of the model changes with the elevation. For the lower part of the bundle, it can be seen that a good agreement is obtained (see Figure (4.2)a-b). The decreases of temperature at 2770 and 2225 mm are somewhat predicted earlier than the experiment, and the maximum values are estimated relatively well



(a) System pressure



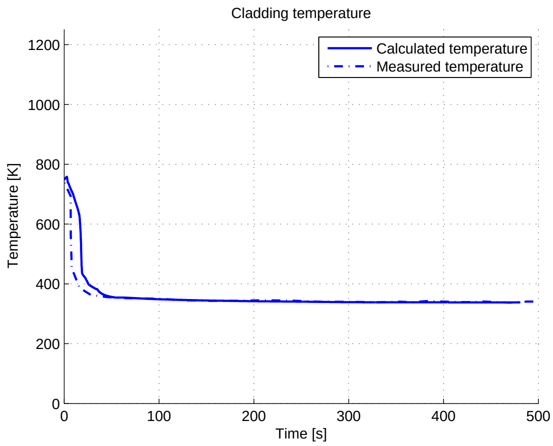
(b) Flow rate



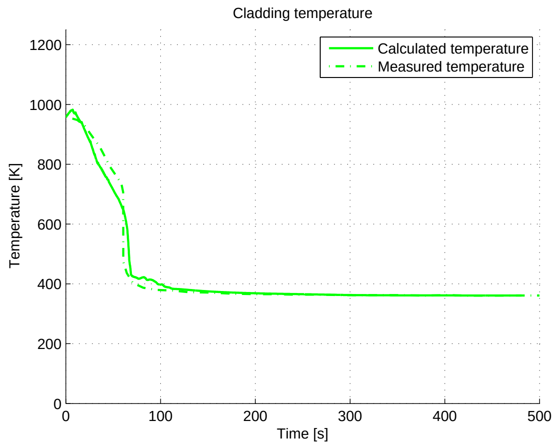
(c) Calculated cladding temperature

Figure 4.1: Pressure and flow rate boundary condition and objective function.

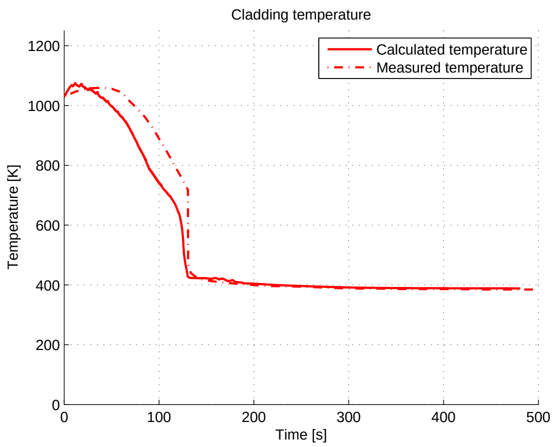
(see Figure (4.2)c-d). For elevations around the middle height, the quenching time and the maximum temperature are underestimated (Figure (4.3)a-b). In the upper part of the core, the quenching time is also underestimated, but the peak in the temperature goes from a good estimation up to a quite significant over-estimation (Figure (4.3)c-d).



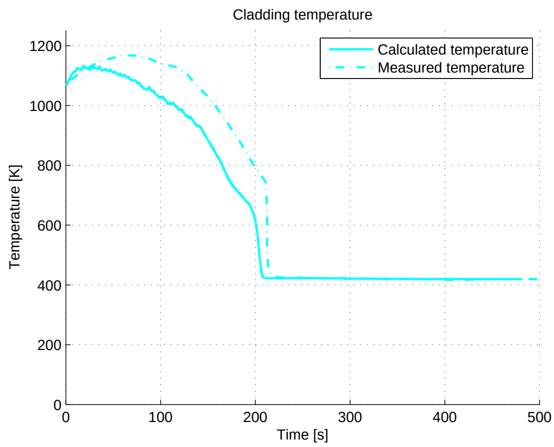
(a) Temperature distribution at 3860 mm (inlet)



(b) Temperature distribution at 3315 mm



(c) Temperature distribution at 2770 mm

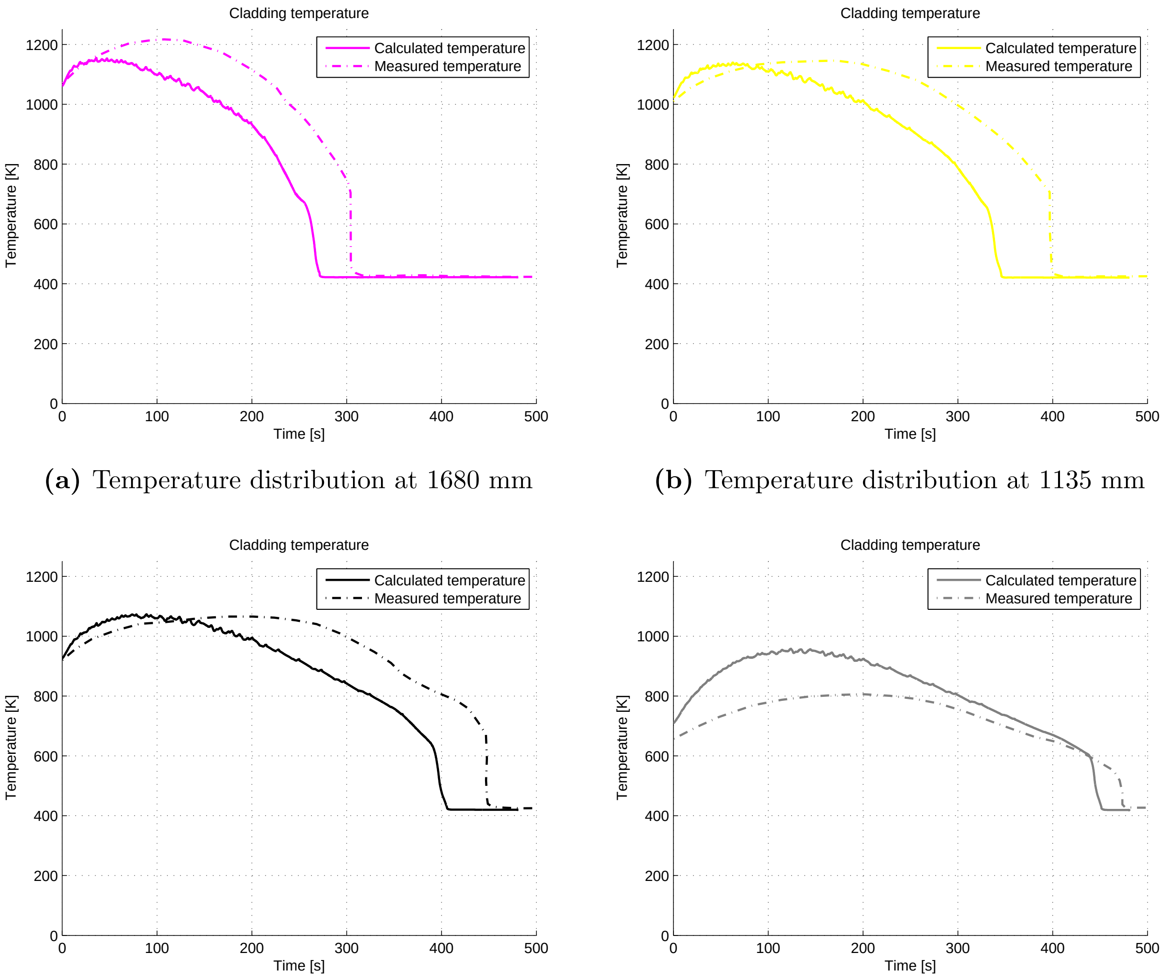


(d) Temperature distribution at 2225 mm

Figure 4.2: Calculated and measured temperatures at different elevations, I.

Looking at Figures (4.2) to (4.3) one can also see another deviation. In fact the calculated cladding temperature in RELAP5 exhibits a oscillatory behavior especially at higher temperatures. This is likely due to sudden density variation from a time step to another, and there is no evidence of oscillations in the measured data.

By comparing the quenching time with the FEBA measurements it appears that RELAP5 predict quenching a bit too early, especially for the higher elevations. The biggest deviation in quenching time appears at 1135 mm from the outlet where



(a) Temperature distribution at 1680 mm

(b) Temperature distribution at 1135 mm

(c) Temperature distribution at 590 mm

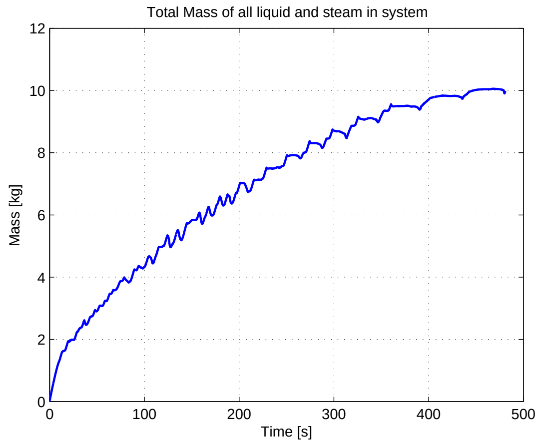
(d) Temperature distribution at 45 mm (outlet)

Figure 4.3: Calculated and measured temperatures at different elevations, II.

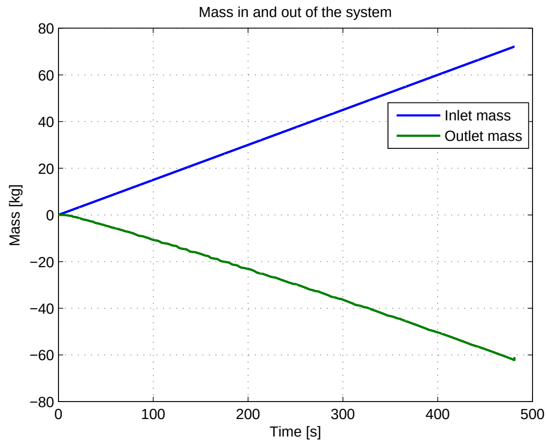
the difference in prediction is around 50 seconds. The time at which the RELAP5 model have been fully quenched occurs somewhere around 455 seconds which can be compared with 480 seconds for the measurements.

4.1.2 Mass Error

As discussed in Subsection (3.3), the mass error related to the calculation was also studied. Figure (4.4) displays the total mass, the incoming and the outgoing masses. One can clearly see in Figure (4.4a) that the total mass increases continuously until the whole system is filled by the fluid. In fact, as shown in Figure (4.4b), the amount of liquid added because of the reflooding phase is larger than the one lost through the break. This difference between incoming and outgoing mass is much larger than the total mass in the system, so, as desired, the impact of the mass error stays very low. It must also be pointed out that the fluctuations in the total mass are due to the fluctuations of the outlet mass. Although it is not visible in Figure (4.4b) because of the scale used, the outlet mass is indeed affected by numerical and physical fluctuations.



(a) Total mass in system



(b) Incoming and outgoing mass

Figure 4.4: Total mass in system and incoming plus outgoing mass in time.

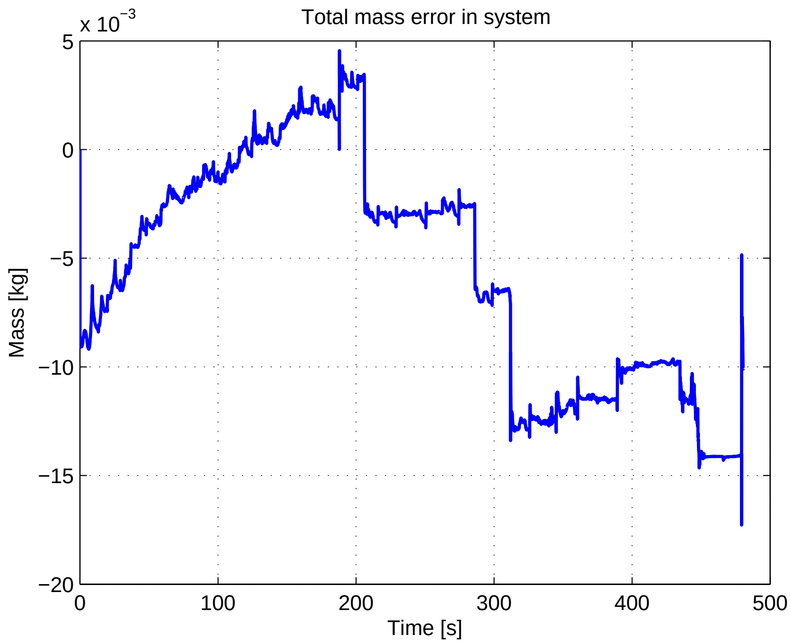


Figure 4.5: Actual mass error in time.

Figure (4.5) shows the corresponding mass error with respect to time for a RELAP5 simulation performed in this work. The mass error ranges from −20 to +5 grams.

4.2 Sensitivity study

After the model was assessed against the measured data given in the PREMIUM benchmark, the second part of the work was focused on the analysis of the influential parameters via a sensitivity study. This study aims to evaluate and measure the impact that certain input parameters have on the computed cladding temperature and this is carried out through DAKOTA.

4.2.1 Selected input parameters

According to the specifications provided within the PREMIUM benchmark, 73 input parameters were considered as potential influential parameters. Only few of them were then identified as significant with respect to criteria in the reflooding experiments. However, not all these input parameters were analyzed in the current thesis, and the sensitivity study was focused on a sub-set of those parameters, which are reported in Table (2).

The parameters having multiple number of positions could be taken at different values of:

- temperatures;
- times;
- spatial distribution.

For instance, the temperature at the inlet of the test bundle was changed at 7 different times along the total transient time and each of these values were modified in order to study the response to the variations.

Table 2: Input parameters in this project.

Input parameters	Total positions	Dependence
Flow area	1	
Heat capacity of heater	2	temperature
Heat capacity of insulation	2	temperature
Hydraulic diameter	1	
Inlet liquid temperature	7	time
Mass flow	1	
Pressure	9	time
Power	7	time
Spacer form loss coefficient	5	elevation
Thermal conductivity heater	2	temperature
Thermal conductivity insulation	5	temperature
Wall roughness	1	

4.2.2 Variation of the selected input parameters

The input parameters reported in Table (2) were varied uniformly between −2% to +2% of their nominal values, according to the procedure described in Sections (2.2) and (3.4). The selected parameters were changed simultaneously so a new set of values of the selected input parameters could be obtained. The new set was then employed to perform a new RELAP5 calculation. This procedure was repeated 50 times and the job was performed in an automatic manner by making use of DAKOTA (see Subsection (2.3.3) and Section (3.4)). The results from all these runs were then processed in MATLAB, see Appendix B.

The several calculations based on the variation of the input parameters were then investigated in terms of the variability of the simulated peak cladding temperature along with its elevation and time of occurrence. The change in maximum cladding temperature is summarized in Table (3).

Table 3: Ranges of maximum cladding temperature, time and elevation.

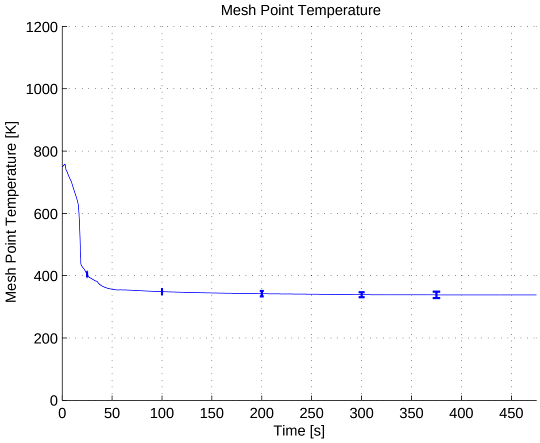
Type	Min - Max	Unit
Temperature	1156.9 - 1170.5	[K]
Time	47.202 - 61.203	[s]
Elevation	2639 - 2739	[mm]

The variation of elevation at which the maximum cladding temperature occurs could be affected by the choice of the nodalization scheme. In the specific case, the spatial discretization of the bundle consists of 42 axial control volumes. Therefore, the length of each axial node is equal to 100 mm that is the same as the variation range for the elevation shown in Table (3). A finer nodalization would definitely lead to a higher resolution and a more accurate evaluation of such a range, but with a heavier computational cost.

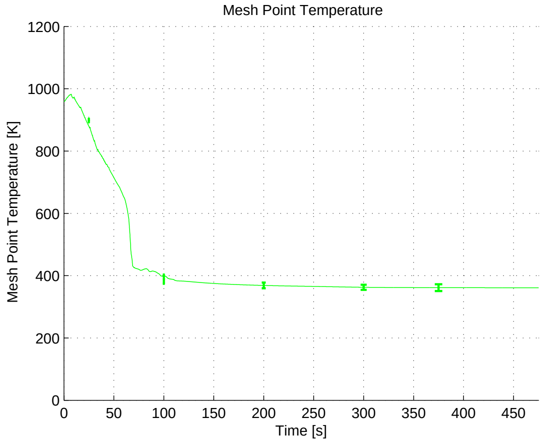
The variation of the cladding temperature at different elevations can be seen in Figures (4.6) to (4.7). As described above, such a variation was achieved from 50 calculations based on the variation of the selected input parameters.

The figures show that the temperatures calculated with RELAP5 tend to change less in the first part of the transient, whereas they are more sensitive in the phase of the transient when they are decreasing, i.e. when the water can start to cool the heated rods properly. For instance, the range of variation of the temperature at 2225 mm is particularly significant at 200 seconds (see Figure (4.6d)).

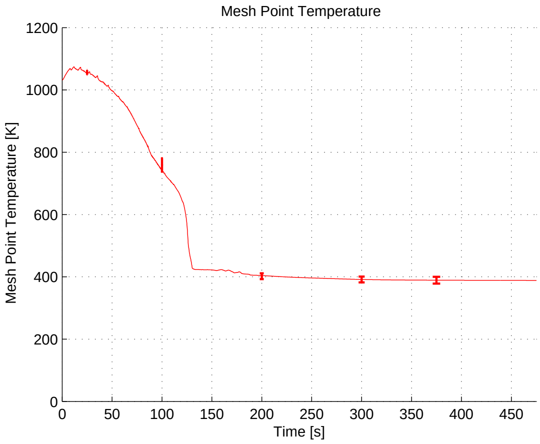
Although the changes of the selected input parameters may be relevant, one must take in account that the effect of one parameter could be compensated by the effect of another parameter, giving a small variation of the cladding temperature.



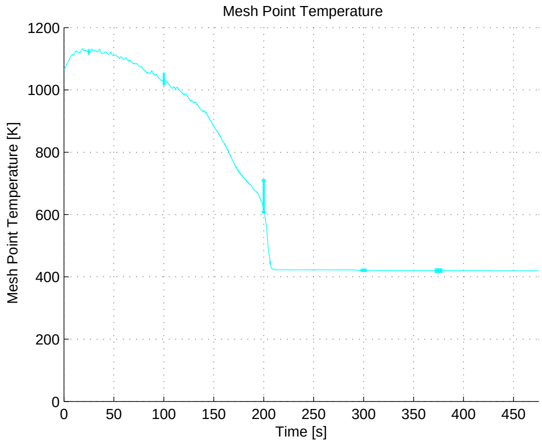
(a) Temperature variation at 3860 mm



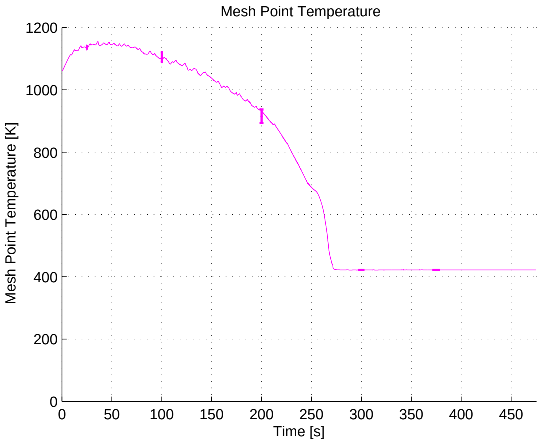
(b) Temperature variation at 3315 mm



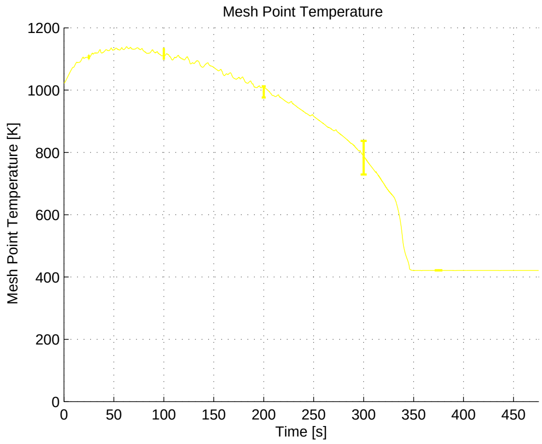
(c) Temperature variation at 2770 mm



(d) Temperature variation at 2225 mm

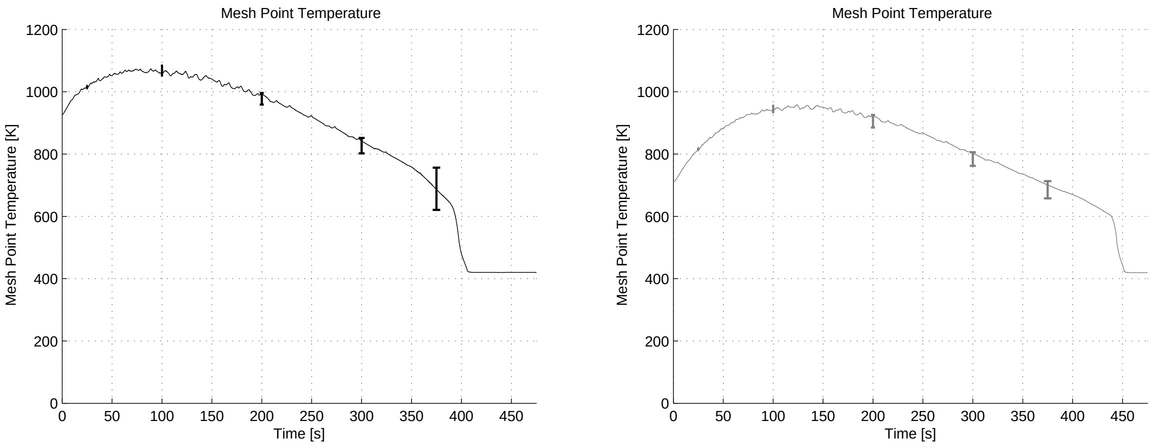


(e) Temperature variation at 1680 mm



(f) Temperature variation at 1135 mm

Figure 4.6: Calculated temperatures due to variation of input parameters, I.



(a) Temperature variation at 590 mm **(b)** Temperature variation at 45 mm (outlet)

Figure 4.7: Calculated temperatures due to variation of input parameters, II.

4.2.3 Ranking of the influential input parameters

In order to determine the influence of each of the selected input parameters on the computed cladding temperature, an analysis of the 50 RELAP5 calculations discussed in the previous sections was performed with the DAKOTA tool, and it was based on the partial rank correlation coefficients.

The impact of one parameter can depend on the system conditions. For instance, variations of inlet temperature can have little contribution when the power level is low or a change of the space form loss coefficient in the middle regions of the core impacts more due to the power profile.

The results are presented in Figure (4.8) and Table (4).

Figure (4.8) shows the absolute value of the correlation coefficients and they are arranged from the highest to the lowest, i.e. from the parameter with the strongest correlation to the parameter with the weakest correlation (for the meaning of the PRCC, see Section (2.2)). The five most infuential input parameters are (i.e., with the highest absolute partial rank correlation coefficients):

i) Mass flow;

ii) Power;

iii) Inlet liquid temperature;

iv) Heat capacity of insulation;

v) Heat capacity of heater.

In table (4) the partial rank correlation coefficients for the input parameters with respect to different 'positions' are reported.

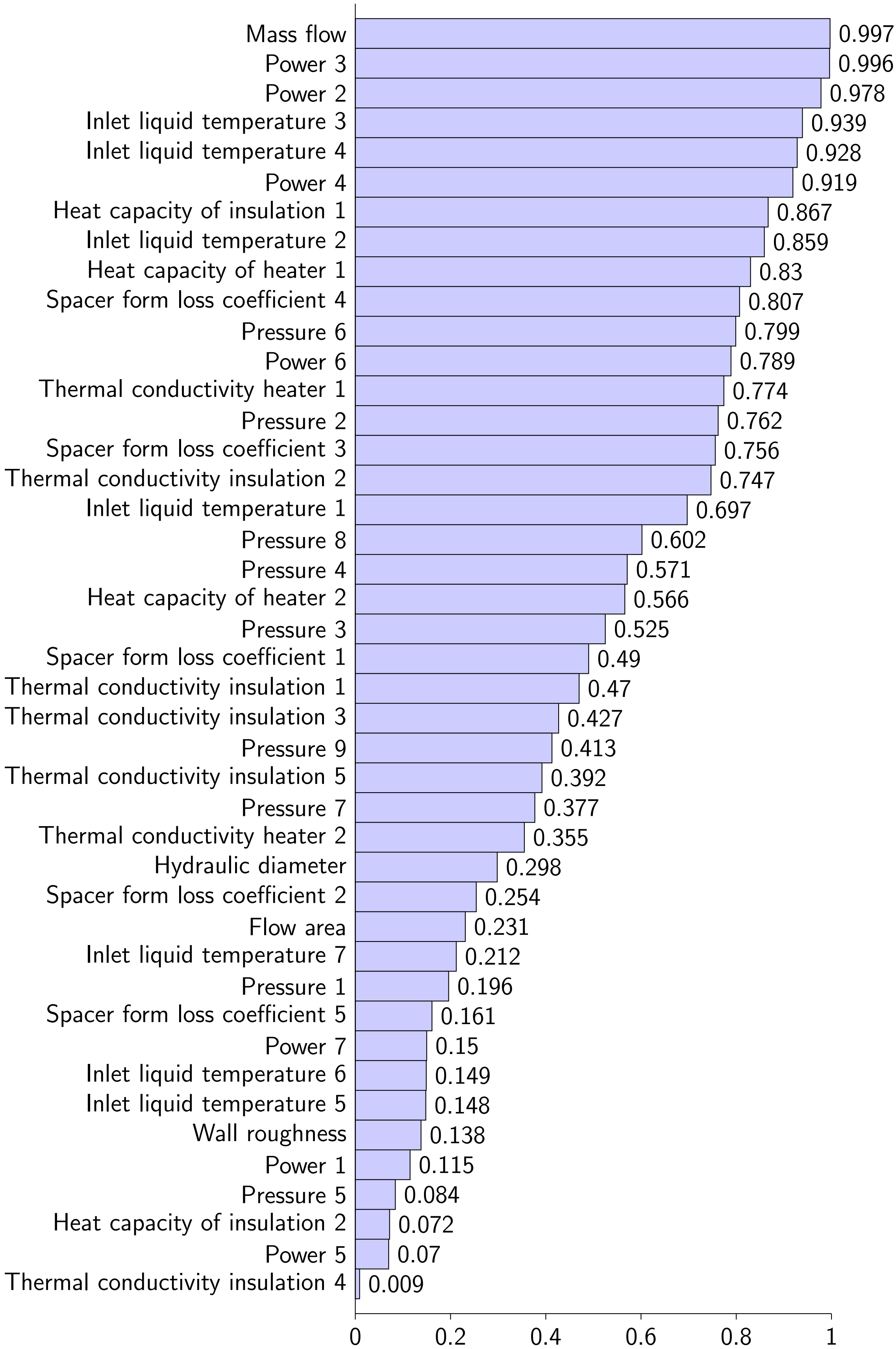


Figure 4.8: Absolute values of the correlation coefficients.

Table 4: Partial Rank Correlation Matrix between input and output.

Input Parameter (tot. Nr.)	(Nr.): PRCC-Value	
Flow area (1)	(1): 0.231	
Heat capacity of heater (2)	(1): -0.830 @195°C	(2): 0.566 @9350°C
Heat capacity of insulation (2)	(1): -0.867 @200°C	(2): -0.072 @9250°C
Hydraulic diameter (1)	(1): -0.298	
Inlet liquid temperature (7)	(1): 0.697 @0.0 s	(2): -0.859 @9.0 s
	(3): -0.939 @16.8 s	(4): -0.928 @54.9 s
	(5): -0.148 @83.7 s	(6): -0.149 @208.7 s
	(7): -0.212 @310.6 s	
Mass flow (1)	(1): -0.997	
Pressure (9)	(1): -0.196 @0.0 s	(2): -0.762 @75.0 s
	(3): 0.525 @150.0 s	(4): -0.571 @225.0 s
	(5): -0.084 @300.0 s	(6): 0.799 @375.0 s
	(7): 0.377 @450.0 s	(8): 0.602 @525.0 s
	(9): 0.413 @600.0 s	
Power (7)	(1): -0.115 @0.4 s	(2): 0.978 @1.9 s
	(3): 0.996 @29.3 s	(4): 0.919 @81.0 s
	(5): -0.070 @187.0 s	(6): 0.789 @250.4 s
	(7): -0.150 @287.7 s	
Spacer form loss coefficient (5)	(1): 0.490 @339 mm	(2): -0.254 @1339 mm
	(3): 0.756 @1939 mm	(4): 0.807 @2939 mm
	(5): -0.161 @3539 mm	
Thermal conductivity heater (2)	(1): 0.774 @195°C	(2): -0.355 @9350°C
Thermal conductivity insulation (5)	(1): 0.47 @200°C	(2): 0.747 @500°C
	(3): 0.427 @900°C	(4): 0.009 @1400°C
	(5): -0.392 @1700°C	
Wall roughness (1)	(1): 0.138	

The impact of the heat capacity of the heater is larger at the low temperature: the PRCC at 195°C is equal to -0.83 whereas the PRCC at 9350°C is 0.566. When RELAP5 estimates the heat capacity, it uses an input table where values are known at discrete temperatures, and interpolates to get the value at the proper temperature needed in the calculation. In this case the actual value happens to be much closer to the low temperature: therefore, changes in the input heat capacity at the lower temperature cause more significant variations. It can also be seen that at low temperature the relationship between cladding temperature and heat capacity is inversely proportional (PRCC<0), and at high temperature is directly proportional (PRCC>0)

Similar results were obtained for the heat capacity of the insulation, even though the relationship is always inversely proportional at low and high temperature.

The effect of the inlet liquid temperature was investigated at different times along the transient. The strongest impact occurs at time 16.8 s and 54.9 s, and the the relationship between this parameter and the cladding temperature is inversely proportional (except for the time at which the transient start).

The role of the system pressure was also analyzed with respect to different times. The largest contribution to the change of the cladding temperature comes from the variation of the system pressure occurring at t=375 s (positive). In addition, these results point out that the changes in system pressure are quite significant between t=75 s and t=225 s, even though the trend is not clear (PRCC is negative at 75 seconds, positive at 150 seconds, and, again, negative at 225 seconds).

For the power correlation coefficients one can see that is is a crucial parameter for the cladding temperature especially when the power is relatively high (where the power is high, the PRCC is also positive). An exception can be observed at 0.4 second, where the power dips just before it peaks (see the behavior in Figure (2.2a)).

The spacer form loss coefficient is analyzed at different heights. The PRCC have three positive values and two negative. The strongest impact occur at height equal to 1939 mm and 2930 mm and these are directly proportional.

The thermal conductivity was varied and analyzed at different locations. For the heater it was in two different locations. At 195°C the PRCC is 0.774 and at 9350°C it is -0.355. The behavior is similar to what was explained for the heat capacity of the heater. However, now the proportionality is direct at lower temperatures, whereas it is inverse at higher temperature. Five different variation points were used for the insulation and the variation was performed at lower temperatures (at 200°C, 500°C, 900°C, 1400°C and 1700°C). The correlation is direct proportional at lower temperatures and peaks at 500°C. At 1700°C the relationship between the cladding temperature is inversely proportional.

According to the variation assigned (see Section 2.2), input parameters as the flow area, the hydraulic diameter, and the wall roughness seem to have a relatively low importance with a PRCC that is around 0.2.

5 SUMMARY AND CONCLUSION

This chapter provides a summary of the work with the main results. Finally, recommendations for future work are given.

5.1 Summary of the work and main conclusions

Computer simulations provide an important support to nuclear safety. The assessment of nuclear codes against experimental data and the study of variability of the results with respect to the variability of the input parameters are essential in order to apply these codes in a reliable manner for safety purposes.

A large effort has been addressed to develop and validate computational models for the simulation of a Loss-Of-Coolant Accident (LOCA) which is one of the most serious scenarios that a nuclear power plant must withstand.

In this context, research programs as the PREMIUM benchmark, which this thesis is related to, allow to reach a better understanding of the capabilities of nuclear systems codes.

The PREMIUM benchmark is based on the analysis of a reflooding experiment carried out in the FEBA facility. Such an experiment reproduces the last phase of a LOCA in which the core is re-flooded and then finally the nuclear fuel can be cooled before a too high cladding temperature is reached. The aim of the benchmark is to investigate and model uncertainties that can impact the prediction of the behavior of the cladding temperature during such an experiment.

In the present work a RELAP5 model for the simulation of the reflooding experiment was evaluated against experimental measurements available in the PREMIUM benchmark. Besides, a sensitivity study was performed in order to analyze the influence of some selected input parameters on the calculation of the cladding temperature.

It was shown that the RELAP5 model can reproduce the evolution of the temperature during the transient in a satisfactory way (see, for instance, Figure (4.1c)). The mass error related to the calculation was also kept low as displayed in Figure (4.5). However, discrepancies for the cladding temperature in terms of quantitative values and quenching time were found, in particular in the upper part of the test section as shown in Figures (4.2) to (4.3).

The impact of some selected input parameters, reported in Table (2), was assessed with respect to the computed cladding temperature. For each input parameter partial rank correlation coefficients were estimated so that the input parameters could be ranked by importance in descending order (Figure (4.8)). Looking at the largest coefficients then the five most influential input parameters are:

i) Mass flow;

ii) Power;

iii) Inlet liquid temperature;

iv) Heat capacity of insulation;

v) Heat capacity of heater.

This sensitivity study was achieved by making use of DAKOTA. Such a software was very suitable for the purpose. In fact it allows to generate random samples for the input parameters of interest; to modify the RELAP5 model according to the variation of the input parameters; to manage the multiple RELAP5 runs based on the variation of the input parameters; and to compute the partial rank correlation coefficients for selected input parameters with respect to output variables.

Conclusively, the contribution of PREMIUM and similar projects is a step forward in the right direction for the development of thermal-hydraulic programs using BEPU approaches for the safety assessment purposes.

5.2 Future work

From the work presented in this thesis, some recommendations for future work can be given.

The RELAP5 model applied to simulate the reflooding experiment shows some discrepancies with the experimental propagation of the quenching water front throughout the test bundle. An interesting study may be to refine the nodalization of the model, and to study the effect of it on the results.

Partial rank correlation coefficients are based on the assumption of a linear model. Therefore, additional sensitivity measures could also be employed so that a deeper investigation of the relationship between variability of the output with respect to the input can be achieved.

The number of calculations for the sensitivity study was chosen equal to 50. One could evaluate whether the partial rank correlation coefficients would be sensitive to a different number of calculations or not.

After the identification of the most influential input parameters, an open question is how the uncertainties related to those parameters could be modeled. To do so, analysis of experimental data and/or testing different probability distribution functions could be beneficial.

Finally, comparison of the RELAP5 results with those calculated from other codes can really be helpful to reach a better knowledge about the modeling of this kind of phenomena.

References

[1] World Energy Needs and Nuclear Power (September 2013).
URL http://www.world-nuclear.org/info/Current-and-Future-Generation/World-Energy-Needs-and-Nuclear-Power/#.UnFemPlLNio

[2] Nuclear Power in the World Today (April 2012).
URL http://www.world-nuclear.org/info/Current-and-Future-Generation/Nuclear-Power-in-the-World-Today/#.UmQHcPnIZip

[3] Nuclear Power Reactors (July 2013).
URL http://www.world-nuclear.org/info/Nuclear-Fuel-Cycle/Power-Reactors/Nuclear-Power-Reactors/#.UlFhvVDIZio

[4] The Boiling Water Reactor (BWR) (29 March 2012).
URL http://www.nrc.gov/reading-rm/basic-ref/students/animated-bwr.html

[5] The Pressurized Water Reactor (PWR) (29 March 2012).
URL http://www.nrc.gov/reading-rm/basic-ref/students/animated-pwr.html

[6] F. Reventós, E. de Alfonso, R. Mendizábal Sanz, PREMIUM – Benchmark on the quantification of the uncertainty of the physical models in the system thermal-hydraulic codes, Phase I: Methodology and data review (2012).

[7] A. Kovtonyuk, A. Petruzzi, F. D'Auria, OECD/NEA PREMIUM Benchmark, Phase II: Identification of influential parameters (November 2012).

[8] P. Vinai, TIF250 - Nuclear Reactor Safety: Lecture slides (2012).

[9] I. Brittain, Thermopedia - reflood (7 February 2011).
URL http://www.thermopedia.com/content/1083/

[10] T. Skorek, A. de Crécy, PREMIUM – Benchmark on the quantification of the uncertainty of the physical models in the system thermal-hydraulic codes (2011).

[11] Nuclear Safety Analysis Division, RELAP5/MOD3.3 Code Manual, Volume I: Code Structure, System Models, and Solution Methods (December 2001).

[12] Nuclear Safety Analysis Division, RELAP5/MOD3.3 Code Manual, Volume II: Appendix A Input Requirements (October 2010).

[13] Applied Programming Technology, Inc., Symbolic Nuclear Analysis Package (SNAP) - User's Manual (February 2011).

[14] Quantitative Modeling and Analysis Department - Sandia National Laboratories, Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis - Version 5.3 User's Manual (February 2013).

[15] Sandia National Laboratories (24 August 2013).
URL http://dakota.sandia.gov/about.html

Appendix A - Main Matlab code

This is the main Matlab-file which extracts data from RELAP5 runs. It compares the experimental data with the calculated data and creates numerous of plots.

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                                                                         %%%
%%%             Matlab file analysing data from RELAP5-Output               %%%
%%%                                  by                                     %%%
%%%                          Rickard Magnusson                              %%%
%%%                                2013                                     %%%
%%%                                                                         %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
clear all;
clc;
format long

javaaddpath('D:\MasterThesis\MATLAB_Masterthesis\matlab.jar');
javaaddpath('D:\MasterThesis\MATLAB_Masterthesis\relap5.jar');
javaaddpath('D:\MasterThesis\MATLAB_Masterthesis\trace.jar');

import com.cafean.matlab.*;
intf=MatlabIntf;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now the file to open
STR1 = 'D:\MasterThesis\RELAP5_Output\Sample_Stream\Base_Job/Base_Job.rst';
intf.openRelapRstplt(STR1);
intf.setUseSIunits(true); % <-- SI units

% Retrieves a list of all open plot files and their respective file types.
setting=menu('Choose setting:','Plots of calc. ','Movie');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Relap_output = intf.getRelapPlotVars(0); %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                  Listing useful parameters and indata                   %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mass_flow_tmdvol = { 'mflowj-2000000' 'mflowj-5000000'};
Pressure = {'p-1010000' 'p-3270000' 'p-4010000'};
% 'p-3010000' <-- dont use this as inlet pressure
% 'p-3410000' <-- don't use this. Use the pressure at tmdptv
Fluid_temp = { 'tempf-1010000' 'tempf-3270000'};
%'tempf-3010000' -||-
Mass = { 'tmass-0'};
Temp_outer = { 'httemp-1000115' 'httemp-1000215' 'httemp-1000315' ...
     'httemp-1000415' 'httemp-1000515' 'httemp-1000615' 'httemp-1000715' ...
     'httemp-1000815' 'httemp-1000915' 'httemp-1001015' 'httemp-1001115' ...
     'httemp-1001215' 'httemp-1001315' 'httemp-1001415' 'httemp-1001515' ...
     'httemp-1001615' 'httemp-1001715' 'httemp-1001815' 'httemp-1001915' ...
     'httemp-1002015' 'httemp-1002115' 'httemp-1002215' 'httemp-1002315' ...
     'httemp-1002415' 'httemp-1002515' 'httemp-1002615' 'httemp-1002715' ...
     'httemp-1002815' 'httemp-1002915' 'httemp-1003015' 'httemp-1003115' ...
     'httemp-1003215' 'httemp-1003315' 'httemp-1003415' 'httemp-1003515' ...
     'httemp-1003615' 'httemp-1003715' 'httemp-1003815' 'httemp-1003915' ...
     'httemp-1004015' 'httemp-1004115'};
Density_f = {'rhof-3010000' 'rhof-3020000' 'rhof-3030000' 'rhof-3040000'...
     'rhof-3050000' 'rhof-3060000' 'rhof-3070000' 'rhof-3080000' ...
```

```
    'rhof-3090000' 'rhof-3100000' 'rhof-3110000' 'rhof-3120000' ...
    'rhof-3130000' 'rhof-3140000' 'rhof-3150000' 'rhof-3160000' ...
    'rhof-3170000' 'rhof-3180000' 'rhof-3190000' 'rhof-3200000' ...
    'rhof-3210000' 'rhof-3220000' 'rhof-3230000' 'rhof-3240000' ...
    'rhof-3250000' 'rhof-3260000' 'rhof-3270000' 'rhof-3280000' ...
    'rhof-3290000' 'rhof-3300000' 'rhof-3310000' 'rhof-3320000' ...
    'rhof-3330000' 'rhof-3340000' 'rhof-3350000' 'rhof-3360000' ...
    'rhof-3370000' 'rhof-3380000' 'rhof-3390000' 'rhof-3400000' ...
    'rhof-3410000'};
Density_g = {'rhog-3010000' 'rhog-3020000' 'rhog-3030000' 'rhog-3040000'...
    'rhog-3050000' 'rhog-3060000' 'rhog-3070000' 'rhog-3080000' ...
    'rhog-3090000' 'rhog-3100000' 'rhog-3110000' 'rhog-3120000' ...
    'rhog-3130000' 'rhog-3140000' 'rhog-3150000' 'rhog-3160000' ...
    'rhog-3170000' 'rhog-3180000' 'rhog-3190000' 'rhog-3200000' ...
    'rhog-3210000' 'rhog-3220000' 'rhog-3230000' 'rhog-3240000' ...
    'rhog-3250000' 'rhog-3260000' 'rhog-3270000' 'rhog-3280000' ...
    'rhog-3290000' 'rhog-3300000' 'rhog-3310000' 'rhog-3320000' ...
    'rhog-3330000' 'rhog-3340000' 'rhog-3350000' 'rhog-3360000' ...
    'rhog-3370000' 'rhog-3380000' 'rhog-3390000' 'rhog-3400000' ...
    'rhog-3410000'};
Void_g = {'voidg-3010000' 'voidg-3020000' 'voidg-3030000' 'voidg-3040000'...
    'voidg-3050000' 'voidg-3060000' 'voidg-3070000' 'voidg-3080000' ...
    'voidg-3090000' 'voidg-3100000' 'voidg-3110000' 'voidg-3120000' ...
    'voidg-3130000' 'voidg-3140000' 'voidg-3150000' 'voidg-3160000' ...
    'voidg-3170000' 'voidg-3180000' 'voidg-3190000' 'voidg-3200000' ...
    'voidg-3210000' 'voidg-3220000' 'voidg-3230000' 'voidg-3240000' ...
    'voidg-3250000' 'voidg-3260000' 'voidg-3270000' 'voidg-3280000' ...
    'voidg-3290000' 'voidg-3300000' 'voidg-3310000' 'voidg-3320000' ...
    'voidg-3330000' 'voidg-3340000' 'voidg-3350000' 'voidg-3360000' ...
    'voidg-3370000' 'voidg-3380000' 'voidg-3390000' 'voidg-3400000' ...
    'voidg-3410000'};
HTC = { 'hthtc-1000101' 'hthtc-1000201' 'hthtc-1000301' ...
    'hthtc-1000401' 'hthtc-1000501' 'hthtc-1000601' 'hthtc-1000701' ...
    'hthtc-1000801' 'hthtc-1000901' 'hthtc-1001001' 'hthtc-1001101' ...
    'hthtc-1001201' 'hthtc-1001301' 'hthtc-1001401' 'hthtc-1001501' ...
    'hthtc-1001601' 'hthtc-1001701' 'hthtc-1001801' 'hthtc-1001901' ...
    'hthtc-1002001' 'hthtc-1002101' 'hthtc-1002201' 'hthtc-1002301' ...
    'hthtc-1002401' 'hthtc-1002501' 'hthtc-1002601' 'hthtc-1002701' ...
    'hthtc-1002801' 'hthtc-1002901' 'hthtc-1003001' 'hthtc-1003101' ...
    'hthtc-1003201' 'hthtc-1003301' 'hthtc-1003401' 'hthtc-1003501' ...
    'hthtc-1003601' 'hthtc-1003701' 'hthtc-1003801' 'hthtc-1003901' ...
    'hthtc-1004001' 'hthtc-1004101'};
Fluid_vel = {'velf-1010000'};
Length(1) = 0.139;
Length(2:40)=0.1;
Length(41) = 0.075;
Area = 3.89318E-3;
Volume = Area * Length;
Time = length(intf.getRelapData(0,Mass));
Mass_flow_pipe ={'mflowj-3010000' 'mflowj-3020000' 'mflowj-3030000' ...
    'mflowj-3040000' 'mflowj-3050000' 'mflowj-3060000' 'mflowj-3070000' ...
    'mflowj-3080000' 'mflowj-3090000' 'mflowj-3100000' 'mflowj-3110000' ...
    'mflowj-3120000' 'mflowj-3130000' 'mflowj-3140000' 'mflowj-3150000' ...
    'mflowj-3160000' 'mflowj-3170000' 'mflowj-3180000' 'mflowj-3190000' ...
    'mflowj-3200000' 'mflowj-3210000' 'mflowj-3220000' 'mflowj-3230000' ...
    'mflowj-3240000' 'mflowj-3250000' 'mflowj-3260000' 'mflowj-3270000' ...
    'mflowj-3280000' 'mflowj-3290000' 'mflowj-3300000' 'mflowj-3310000' ...
```

```
    'mflowj-3330000' 'mflowj-3340000' 'mflowj-3350000' 'mflowj-3360000' ...
    'mflowj-3370000' 'mflowj-3380000' 'mflowj-3390000' 'mflowj-3400000'};

F_m=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Floodvelocity.txt');
Bench_Floodvel =[F_m(:,1) F_m(:,2)*0.01];
B_P_kw=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Power.txt');
Bench_Power =[B_P_kw(:,1) B_P_kw(:,2)*1000];
P_o_bar=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Pressure.txt');
Bench_Press_outlet =[P_o_bar(:,1) P_o_bar(:,2)*100000];
B_Tc_i_C=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Tcool.txt');
Bench_Tcool_inlet =[B_Tc_i_C(:,1) B_Tc_i_C(:,2)+273];
Temp45=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp45.txt');
Temp590=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp590.txt');
Temp1135=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp1135.txt');
Temp1680=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp1680.txt');
Temp2225=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp2225.txt');
Temp2770=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp2770.txt');
Temp3315=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp3315.txt');
Temp3860=load('D:\MasterThesis\MATLAB_Masterthesis\Benchmark\Temp3860.txt');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                    Saving parameters into matrixes                  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                             Pressure                                %%%
tic;
N=length(Pressure);
info_pressure=cell(2,N);
j=1;
y_pressure=zeros(Time,N);
for i =1:N
    parameter = Pressure(i);
    a1 = intf.getRelapData(0,parameter);

    y_pressure(:,i) = a1(:,2);
    if j==1
        info_pressure(1,j)=intf.getRelapPlotVarUnits(0,parameter);
        info_pressure(2,j)=intf.getRelapPlotVarType(0,parameter);
    end
    j=j+1;
end
x_data(:,1) = a1(:,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                           Fluid Temp                                %%%
N=length(Fluid_temp);
info_temp_f=cell(2,N);
j=1;
y_temp_f=zeros(Time,N);
for i =1:N
    parameter = Fluid_temp(i);
    a1 = intf.getRelapData(0,parameter);

    y_temp_f(:,i) = a1(:,2);
    if j==1
        info_temp_f(1,j)=intf.getRelapPlotVarUnits(0,parameter);
        info_temp_f(2,j)=intf.getRelapPlotVarType(0,parameter);
```

```
        end
        j=j+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                        Total Mass e-mass                       %%%
N=length(Mass);
info_mass=cell(2,N);
j=1;
y_emass=zeros(Time,N);
for i =1:N
        parameter_mass = Mass(i);
        a1_mf = intf.getRelapData(0,parameter_mass);
        y_emass(:,i) = a1_mf(:,2);

        if j==1
            info_mass(1,j)=intf.getRelapPlotVarUnits(0,parameter_mass);
            info_mass(2,j)=intf.getRelapPlotVarType(0,parameter_mass);
        end
        j=j+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                        Fluid Velocity                          %%%
N=length(Fluid_vel);
info_Fluid_vel=cell(2,N);
j=1;
y_Fluid_vel=zeros(Time,N);
for i =1:N
        parameter_mass = Fluid_vel(i);
       %info(2,i)=intf.getRelapPlotVarType(0,parameter);
       %info(1,i)=intf.getRelapPlotVarUnits(0,parameter);
        a1_mf = intf.getRelapData(0,parameter_mass);
        y_Fluid_vel(:,i) = a1_mf(:,2);

        if j==1
            info_Fluid_vel(1,j)=intf.getRelapPlotVarUnits(0,parameter_mass);
            info_Fluid_vel(2,j)=intf.getRelapPlotVarType(0,parameter_mass);
        end
        j=j+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                            HTTEMP                              %%%
N=length(Temp_outer);
info_temp_c=cell(2,N);
j=1;
y_temp_c=zeros(Time,N);
for i =1:N
        parameter = Temp_outer(i);
        a1 = intf.getRelapData(0,parameter);

        y_temp_c(:,i) = a1(:,2);
        if j==1
            info_temp_c(1,j)=intf.getRelapPlotVarUnits(0,parameter);
            info_temp_c(2,j)=intf.getRelapPlotVarType(0,parameter);
        end
        j=j+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%                              Massflow                              %%%

%%%% Massflow in TMDVOL <-- the interesting one to calc the mass balance
N=length(Mass_flow_tmdvol);
info_massflow_tmdvol=cell(2,N);
j=1;
y_massflow_tmdvol=zeros(Time,N);
for i =1:N
    parameter = Mass_flow_tmdvol(i);
    a1 = intf.getRelapData(0,parameter);

    y_massflow_tmdvol(:,i) = a1(:,2);
    if j==1
        info_massflow_tmdvol(1,j)=intf.getRelapPlotVarUnits(0,parameter);
        info_massflow_tmdvol(2,j)=intf.getRelapPlotVarType(0,parameter);
    end
    j=j+1;
end
int_inl = trapz(x_data(:),y_massflow_tmdvol(:,1));
int_outl = trapz(x_data(:),y_massflow_tmdvol(:,2));

%%%% Massflow in pipe
N=length(Mass_flow_pipe);
info_massflow=cell(2,N);
j=1;
y_massflow=zeros(Time,N);
for i =1:N
    parameter = Mass_flow_pipe(i);
    a1 = intf.getRelapData(0,parameter);

    y_massflow(:,i) = a1(:,2);
    if j==1
        info_massflow(1,j)=intf.getRelapPlotVarUnits(0,parameter);
        info_massflow(2,j)=intf.getRelapPlotVarType(0,parameter);
    end
    j=j+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                        rho_G, rho_F, void_G                        %%%
N=length(Density_f);
info_Density_f=cell(2,N);
info_Density_g=cell(2,N);
info_Void_g=cell(2,N);
j=1;
y_densF=zeros(Time,N);
y_densG=zeros(Time,N);
VoidG=zeros(Time,N);
for i =1:N
    parameter_densF = Density_f(i);
    parameter_densG = Density_g(i);
    parameter_VoidG = Void_g(i);
    a1_densF = intf.getRelapData(0,parameter_densF);
    a1_densG = intf.getRelapData(0,parameter_densG);
    a1_VoidG = intf.getRelapData(0,parameter_VoidG);

    y_densF(:,i) = a1_densF(:,2);
```

```
    y_densG(:,i) = a1_densG(:,2);
    VoidG(:,i) = a1_VoidG(:,2);

    if j==1
        info_Void_g(1,j)=intf.getRelapPlotVarUnits(0,parameter_VoidG);
        info_Void_g(2,j)=intf.getRelapPlotVarType(0,parameter_VoidG);
    end
    j=j+1;

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                          Calculating mass balance                        %%%
[Time,N] = size(VoidG);
mass=zeros(Time,N);
for i=1:Time

    for j=1:N
        mass(i,j) = Volume(j)*(VoidG(i,j)*y_densG(i,j)+(1-VoidG(i,j))*...
            y_densF(i,j));

    end
end
unit_v = ones(41,1);    % might as well use "SUM"
mass_sum = mass*unit_v; % The total mass of the system for all cells.

mass_in = y_massflow_tmdvol(:,1)';
mass_out= y_massflow_tmdvol(:,2)';
mass_in(1,1)=0;                  %mass_in(1) Should be zero
mass_out(1,1)=0;                 %mass_out(1) Should be zero

unit_v = ones(Time,1);
Mass_in  = zeros(1,Time);
Mass_out = zeros(1,Time);
Tot_in   = zeros(1,Time);
Tot_out  = zeros(1,Time);
ERROR = zeros(1,Time);

for i = 2:Time
    deltat = x_data(i) - x_data(i-1);
    Mass_in(1,i) =  1/2 * deltat * (mass_in(i) + mass_in(i-1));
    Mass_out(1,i) = 1/2 * deltat * (mass_out(i) + mass_out(i-1));
    Tot_in(i) = Mass_in*unit_v;
    Tot_out(i)= Mass_out*unit_v;
    ERROR(i) = mass_sum(1)+ Tot_in(i) - Tot_out(i) -mass_sum(i);
end

mass_diff = Tot_out-Tot_in ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                               Void coefficient                           %%%

%Void = {'voidg-3040000' 'voidg-3120000' 'voidg-3300000' 'voidg-3390000'};
% % 2 voidg-values to calc the velocity of the fluid, when power is zero.
% % This script below calculates the velocity at alpha = 0.5 using linear
% % interpolation between 0.4 and 0.6.

% % A massflow in SNAP 0.1480 kg/s ---> flow vel =  0.0380941
```

```matlab
% Void = [y_VoidG(:,2)  y_VoidG(:,40) ] ;
%
% val2 = find(Void(:,1)<0.6 & Void(:,1)>0.4)
% val41= find(Void(:,2)<0.6 & Void(:,2)>0.4)
%
% for i = 1:length(val2)
%     new_2(i)  = Void(val2(i),1); % also find the corresponding time-step
%     time_2(i) = x_data(val2(i));
% end
% p1 = polyfit(time_2,new_2,1);
% t_1 = ( 0.5-p1(2) )/p1(1) ;
%
% for i = 1:length(val41)
%     new_41(i)  = Void(val41(i),2)
%     time_41(i) = x_data(val41(i))
% end
% p2 = polyfit(time_41,new_41,1);
% t_2 = ( 0.5-p2(2) )/p2(1) ;
%
%% elevation at cell 2 and cell 41
% cell2  = Length(1)+Length(2)/2
% cell41 = Length(1)+Length(2)*38+ Length(40)/2
%
% Zero_P_vel = (cell41-cell2)/(t_2-t_1)

toc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                                 PLOTS                                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if setting == 1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                             Pressure                                 %%%
    figure
    axis([0 485 0 500000])
    hold on
%     plot(x_data(:),y_pressure(:,1),'b-',x_data(:),y_pressure(:,2),...
%     'g-', x_data(:),y_pressure(:,3),'r--',Bench_Press_outlet(:,1),...
%         Bench_Press_outlet(:,2),'k-')
    plot(x_data(:),y_pressure(:,3),'LineWidth',1.5,'Color','b')
    plot(Bench_Press_outlet(:,1), Bench_Press_outlet(:,2),'LineWidth',...
        1.5,'Color','k','LineStyle','-.')
    grid on
    xlabel('Time [s]')
    ylabel('Pressure [Pa]')
    title('System Pressure')
    legend('Used Pressure','PREMIUM Pressure')
    hold off
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                            Fluid Temp                                %%%
    figure
    axis([0 485 0 440])
    hold on
    plot(x_data(:),y_temp_f(:,1),'LineWidth',1.5,'Color','b');
    plot(Bench_Tcool_inlet(:,1),Bench_Tcool_inlet(:,2),'LineWidth',1.5,...
        'Color','k','LineStyle','-.');
    grid on
    xlabel('Time [s]')
```

```
    ylabel('Temperature [K]')
    legend('Used inlet temp at tmdpvol','Benchmark inlet')
    title('Feedwater Temperature')
    hold off
%     figure
%     h=plot(x_data(:),y_temp_f(:,1),x_data(:), y_temp_f(:,2),...
%         Bench_Tcool_inlet(:,1),Bench_Tcool_inlet(:,2),'LineWidth',2);
%     set(h(1),'linewidth',2);
%     set(h(2),'linewidth',2);
%     set(h(3),'linewidth',2);
%     grid on
%     xlabel('Time [s]')
%     ylabel(strcat(info_temp_f(2,1),' [',info_temp_f(1,1),']'))
%     legend('Calc. inlet temp at tmdpvol','Calc. temp at cell 27',...
%         'Benchmark inlet')
%     title(info_temp_f(2,1))

    %%     %
    %%     % h=plot(x1,y1,x2,y2,'LineWidth',1);
    %%     % set(h(1),'linewidth',1);
    %%     % set(h(2),'linewidth',2);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                          POWER                                %%%

    Power  =[2.0e5 2.0e5 1.957e5 1.887e5 1.777e5 1.702e5 1.594e5 1.547e5...
        1.427e5 1.354e5 1.284e5 1.225e5 1.226e5 1.222e5 1.221e5];
    % Version 2 of the changed boundary condition
    Time_P =[-1.0 1.75 6.0 13.06 29.27 50.5 81.06 100.56 150.56 187.0 ...
        250.41 287.65 356.03 448.92 600.0];
    figure
    axis([0 485 0 250000])
    hold on
    plot(Time_P,Power,'LineWidth',1.5,'Color','b')
    plot(Bench_Power(:,1), Bench_Power(:,2),'LineWidth',1.5,'Color','k',...
        'LineStyle','-.')
    grid on
    xlabel('Time [s]')
    ylabel('Power [W]')
    legend('Actual used Power','Benchmark Power')
    title('Bundle Power')
    hold off
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                       Flow Velocity                           %%%
    figure
    axis([0 485 0 50])
    hold on
    plot(x_data(:),1000*y_Fluid_vel(:,1),'LineWidth',1.5,'Color','b')
    plot(Bench_Floodvel(:,1), 1000*Bench_Floodvel(:,2),'LineWidth',1.5,...
        'Color','k','LineStyle','-.')
    grid on
    xlabel('Time [s]')
    ylabel('Velocity [mm/s]')
    legend('Calc. Flow vel','Benchmark Flow vel')
    title('Liquid average flow velocity')
    hold off
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                          Massflow                             %%%
```

```
%     figure
%     plot(x_data(:),y_massflow(:,1),'-','color',[0,0,0])
%     grid on
%     xlabel('Time [s]')
%     ylabel(strcat(info_massflow(2,1),' [',info_massflow(1,1),']'))
%     legend('Calc. massflow for cell 1 in pipe')
%     title('Mass flow rate in pipe')
%
%     figure
%     hold on
%     plot(x_data(:),y_massflow_tmdvol(:,1),'-','color',[0,0,1])
%     plot(x_data(:),y_massflow_tmdvol(:,2),'-','color',[0,1,0])
%     grid on
%     xlabel('Time [s]')
%     ylabel(strcat(info_massflow(2,1),' [',info_massflow(1,1),']'))
%     legend('Calc. mass flow inl-tmdvol','Calc. mass flow in outl-tmdvol')
%     title('Mass flow rate in tmdvol')
%     hold off

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                           Comparison with BC                            %%%
    POS=[0 45 590 1135 1680 2225 2770 3315 3860 4114]';
    AM=4114;
    Nf=41;
    Nc=0;
    CELLPOSf=(1-(POS/AM))*Nf;
    CELLPOSc=(1-(POS/AM))*Nc;
    figure
    axis([0 525 0 1250])
    hold on
    plot(x_data(:),y_temp_c(:,3),'-','color',[0,0,1],'LineWidth',1.5)
    plot(x_data(:),y_temp_c(:,8),'-','color',[0,1,0],'LineWidth',1.5)
    plot(x_data(:),y_temp_c(:,13),'-','color',[1,0,0],'LineWidth',1.5)
    plot(x_data(:),y_temp_c(:,19),'-','color',[0,1,1],'LineWidth',1.5)
    plot(x_data(:),y_temp_c(:,24),'-','color',[1,0,1],'LineWidth',1.5)
    plot(x_data(:),y_temp_c(:,30),'-','color',[1,1,0],'LineWidth',1.5)
    plot(x_data(:),y_temp_c(:,35),'-','color',[0,0,0],'LineWidth',1.5)
    plot(x_data(:),y_temp_c(:,40),'-','color',[0.5,0.5,0.5],'LineWidth',...
        1.5)
    plot(Temp3860(:,1),Temp3860(:,2)+273,'-.','color',...
        [0,0,1],'LineWidth',1.5)
    plot(Temp3315(:,1),Temp3315(:,2)+273,'-.','color',...
        [0,1,0],'LineWidth',1.5)
    plot(Temp2770(:,1),Temp2770(:,2)+273,'-.','color',...
        [1,0,0],'LineWidth',1.5)
    plot(Temp2225(:,1),Temp2225(:,2)+273,'-.','color',...
        [0,1,1],'LineWidth',1.5)
    plot(Temp1680(:,1),Temp1680(:,2)+273,'-.','color',...
        [1,0,1],'LineWidth',1.5)
    plot(Temp1135(:,1),Temp1135(:,2)+273,'-.','color',...
        [1,1,0],'LineWidth',1.5)
    plot(Temp590(:,1),Temp590(:,2)+273,'-.','color',...
        [0,0,0],'LineWidth',1.5)
    plot(Temp45(:,1),Temp45(:,2)+273,'-.','color',...
        [0.5,0.5,0.5],'LineWidth',1.5)
    grid on
    xlabel('Time [s]')
```

```
    ylabel('Temperature [K]')
    legend('Temp at 3860 [mm] (inl)','Temp at 3315 [mm]',...
        'Temp at 2770 [mm]','Temp at 2225 [mm]','Temp at 1680 [mm]',...
        'Temp at 1135 [mm]','Temp at 590 [mm]','Temp at 45 [mm] (outl)')
    title('Measured cladding temperature')
    text('Position',[380 150],'string','- = calculated')
    text('Position',[375 110],'string','-. = benchmark')
    hold off
    %       plot(x_data(:),y_temp_c(:,3),'-','color',[0,0,1])
    %       plot(x_data(:),y_temp_c(:,8),'-','color',[0,1,0])
    %       plot(x_data(:),y_temp_c(:,13),'-','color',[1,0,0])
    %       plot(x_data(:),y_temp_c(:,19),'-','color',[0,1,1])
    %       plot(x_data(:),y_temp_c(:,24),'-','color',[1,0,1])
    %       plot(x_data(:),y_temp_c(:,30),'-','color',[1,1,0])
    %       plot(x_data(:),y_temp_c(:,35),'-','color',[0,0,0])
    %       plot(x_data(:),y_temp_c(:,40),'-','color',[0.5,0.5,0.5])
    %       plot(Temp3860(:,1),Temp3860(:,2)+273,'-.','color',[0,0,1])
    %       plot(Temp3315(:,1),Temp3315(:,2)+273,'-.','color',[0,1,0])
    %       plot(Temp2770(:,1),Temp2770(:,2)+273,'-.','color',[1,0,0])
    %       plot(Temp2225(:,1),Temp2225(:,2)+273,'-.','color',[0,1,1])
    %       plot(Temp1680(:,1),Temp1680(:,2)+273,'-.','color',[1,0,1])
    %       plot(Temp1135(:,1),Temp1135(:,2)+273,'-.','color',[1,1,0])
    %       plot(Temp590(:,1),Temp590(:,2)+273,'-.','color',[0,0,0])
    %       plot(Temp45(:,1),Temp45(:,2)+273,'-.','color',[0.5,0.5,0.5])

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                               Total Mass                               %%%
    figure
    plot(x_data(:),y_emass(:,1),'-k')
    grid on
    xlabel('Time [s]')
    ylabel(strcat(info_mass(2,1),' [',info_mass(1,1),']'))
    title('Total Mass of all liquid and steam in system calculated by RELAP')
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%                                  Mass                                  %%%

    %%%%% Total mass in all cells
    figure
    plot(x_data(:),mass_sum,'LineWidth',1.5)
    grid on
    xlabel('Time [s]')
    ylabel('Mass [kg]')
    title('Total Mass of all liquid and steam in system')
    %%%%% Mass in and mass out
    figure
    plot(x_data(:),Tot_in,x_data(:),-Tot_out,'LineWidth',1.5)
    grid on
    xlabel('Time [s]')
    ylabel('Mass [kg]')
    legend('Inlet mass','Outlet mass')
    title('Mass in and out of the system')
    %%%%% Total mass ERROR in the system
    figure
    plot(x_data(1:end-1),ERROR(1:end-1),'LineWidth',1.5)
    grid on
    xlabel('Time [s]')
```

```
    ylabel('Mass [kg]')
    title('Total mass error in system')

    %%%% Mass error over total mass in time
    ratio_mass = abs(ERROR./mass_sum');
    figure
    plot(x_data(1:end-1),ratio_mass(1:end-1),'LineWidth',1.5)
    grid on
    xlabel('Time [s]')
    ylabel('Mass [kg]')
    title('Mass error ratio in time')

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%       Cladding temp dist at all nodes for different times          %%%
x=1:100.3415:4114;
figure
hold on
axis([0  1300 0 4200])
% choose time approximate to [25 100 200 300 375 and ]
plot(y_temp_c(251,:),x,'LineWidth',1.5,'color',[0,0,1])
plot(y_temp_c(1001,:),x,'LineWidth',1.5,'color',[0,1,0])
plot(y_temp_c(2001,:),x,'LineWidth',1.5,'color',[1,0,0])
plot(y_temp_c(3001,:),x,'LineWidth',1.5,'color',[0,1,1])
plot(y_temp_c(3751,:),x,'LineWidth',1.5,'color',[1,0,1])
plot(y_temp_c(4530,:),x,'LineWidth',1.5,'color',[1,1,0])
grid on
xlabel(strcat(info_temp_c(2,1),' [',info_temp_c(1,1),']'))
ylabel('Height')
title('Temp. dist. throughout the pipe')
legend('At 25 [s]','At 100 [s]','At 200 [s]','At 300 [s]', ...
    'At 375 [s]', 'At 452.9 [s]')
hold off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                          Movie with HTTEMP                           %%%
if setting ==2
    numFrames=length(y_temp_c);
    x=1:100.3415:4114;
    movClip = moviein(numFrames);
    %xlim([0 1300])
    step = 2;
    for i=1:step:numFrames;
        plot(y_temp_c(i,:),x,'-k')
        axis([0  1300 0 4200])
        grid on
        xlabel(strcat(info_temp_c(2,1),' [',info_temp_c(1,1),']'))
        ylabel('Height')
        title({'Temp. dist in the pipe'; sprintf('Time %d', x_data(i))})
        %movClip(i)=getframe;
        movClip((i-1)/step+1)=getframe(gcf);
    end
    movie2avi(movClip, 'plot2.avi')
end
```

Appendix B - Matlab code used in the DAKOTA-coupling

This Matlab-file is used for the coupling with DAKOTA and it extracts useful data.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                                                                  %%%
%%%                Matlab file for the Dakota-coupling               %%%
%%%                                by                                %%%
%%%                         Rickard Magnusson                        %%%
%%%                               2013                               %%%
%%%                                                                  %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;
clear all;
clc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                            DYNAMIC WAY                           %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
javaaddpath('/home/rickard/Documents/MATLAB_Masterthesis/matlab.jar');
javaaddpath('/home/rickard/Documents/MATLAB_Masterthesis/relap5.jar');
javaaddpath('/home/rickard/Documents/MATLAB_Masterthesis/trace.jar');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                             STATIC WAY                           %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% add those files to $matlab\toolbox\local\classpath.txt
% C:\Program Files\snap\lib\matlab.jar
% C:\Program Files\snap\plugins\relap5.jar
% C:\Program Files\snap\plugins\trace.jar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
javaclasspath;

import com.cafean.matlab.*
intf=MatlabIntf;
intf.openRelapDmx('demux.dmx');

methods MatlabIntf
intf.help
intf.getVersion
intf.getFileList
intf.getFileInfo
intf.getRelapFileInfo
intf.setUseSIunits( true )
intf.getUseSIunits
intf.getRelapPlotVars(0)
% Amount of timesteps
x = length(intf.getRelapData(0,{'httemp-1000115'}));

T_clad=zeros(41,x,2);
string1='httemp-1000';
string2='httemp-100';
for i=1:41
    a(i)=115+100*(i-1);
    if i<10
        string=[string1 int2str(a(i))];
        T_clad(i,:,:) = intf.getRelapData(0,{string});
    else
        string=[string2 int2str(a(i))];
```

```
            T_clad(i,:,:) = intf.getRelapData(0,{string});
        end
end
size(T_clad)
% Maximum cladding temperature
A1(1)=max(max((T_clad(:,:,2))));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%          CORRESPONDING TIME AND ELEVATION TO MAX. CLAD. TEMP           %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% finding elevation and timing when and where the max value was reached
max_val = max(max(T_clad(:,:,2)));
[rows cols] = find(T_clad(:,:,2)==max_val);
% elev = use the rows
cell1 = 0.139; %[m]
cell2_40 = 0.1;
cell41 = 0.075;

if rows == 41
    elevation = cell1+cell2_40*(rows-2)+cell41;
else
    elevation = cell1+cell2_40*(rows-1);
end
%time = use the cols to search in t_clad(:,:,1)
time = T_clad(rows,cols,1);
%time and elevation at which highest clad temp was achieved
Time_Elevation = [time elevation]

T_elev =[T_clad(24,:,1)' T_clad(24,:,2)'];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                      CREATING DATA FOR ERROR BARS                      %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                     LINEAR EXTRP. AT CERTAIN TIMES                     %%%
pos25 = find(T_clad(1,:,1)<25.42 & T_clad(1,:,1)>24.58);
pos100 = find(T_clad(1,:,1)<100.42 & T_clad(1,:,1)>99.58);
pos200 = find(T_clad(1,:,1)<200.42 & T_clad(1,:,1)>199.58);
pos300 = find(T_clad(1,:,1)<300.42 & T_clad(1,:,1)>299.58);
pos375 = find(T_clad(1,:,1)<375.42 & T_clad(1,:,1)>374.58);

cell = [3 8 13 19 24 30 35 40];
format long
for j = 1:length(cell)
%%% AT TIME 25
    for i = 1:length(pos25)
        T_25(i,j) = T_clad(cell(j),pos25(i),2);
        if j == 1
            time_25(i) = T_clad(1,pos25(i),1);
        end
    end
    p1=polyfit(time_25',T_25(:,j),1);
    T_C_25(j) = p1(2)+25*p1(1);
%%% AT TIME 100
    for i = 1:length(pos100)
        T_100(i,j) = T_clad(cell(j),pos100(i),2);
        if j == 1
            time_100(i) = T_clad(1,pos100(i),1);
        end
```

```
    end
    p2=polyfit(time_100',T_100(:,j),1);
    T_C_100(j) = p2(2)+100*p2(1);
%%% AT TIME 200
    for i = 1:length(pos200)
        T_200(i,j) = T_clad(cell(j),pos200(i),2);
        if j == 1
            time_200(i) = T_clad(1,pos200(i),1);
        end
    end
    p3=polyfit(time_200',T_200(:,j),1);
    T_C_200(j) = p3(2)+200*p3(1);
%%% AT TIME 300
    for i = 1:length(pos300)
        T_300(i,j) = T_clad(cell(j),pos300(i),2);
        if j == 1
            time_300(i) = T_clad(1,pos300(i),1);
        end
    end
    p4=polyfit(time_300',T_300(:,j),1);
    T_C_300(j) = p4(2)+300*p4(1);
%%% AT TIME 375
    for i = 1:length(pos375)
        T_375(i,j) = T_clad(cell(j),pos375(i),2);
        if j == 1
            time_375(i) = T_clad(1,pos375(i),1);
        end
    end
    p5=polyfit(time_375',T_375(:,j),1);
    T_C_375(j) = p5(2)+375*p5(1);
end
T_bench = [T_C_25; T_C_100; T_C_200; T_C_300; T_C_375;];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                          WRITING TO RESULT FILES                          %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dlmwrite('results.out.1',A1);
dlmwrite('results_error_bar.out.2',T_bench);
dlmwrite('results_time_steps.out.3',x);
dlmwrite('results.out.4',T_elev);
dlmwrite('result_time_elevation.out',Time_Elevation);
%%%dlmwrite('results.out.5',check);

dlmwrite('results.tmp',A1);
fclose('all');
exit
```