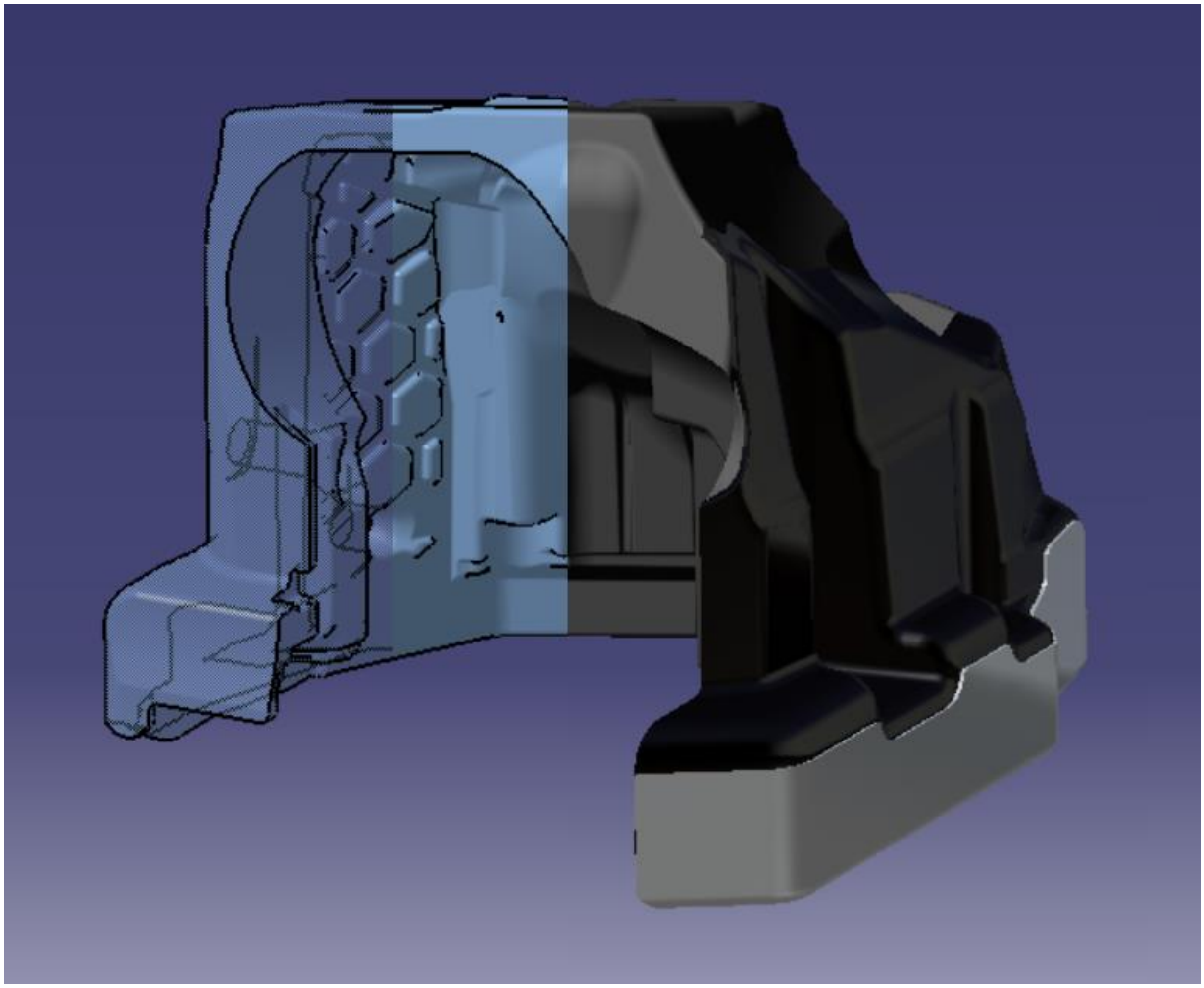
CHALMERS
UNIVERSITY OF TECHNOLOGY



Development of an optimization tool for engine encapsulations

For heat transfer using Finite Volume Method

Master's thesis in Product Development

MARTIN JÖNSSON
DANIEL TALANI

Department of Product and Production Development
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

MASTER'S THESIS IN PRODUCT DEVELOPMENT

Development of an optimization tool for engine encapsulations

For heat transfer using Finite Volume Method

MATRIN JÖNSSON
DANIEL TALANI

Department of Product and Production Development
Division of Product Development
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2015

Development of an optimization tool for engine encapsulations
MARTIN JÖNSSON
DANIEL TALANI

Master's Thesis
ISSN
Department of Product and Production Development
Division of Product Development
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Model of an encapsulation transforming from calculation to prototype

[Printing Service]
Gothenburg, Sweden 2015

ABSTRACT

Volvo Car Group is constantly trying to improve the development processes. The 202020 plan is a current goal of being able to developing brand new cars within 20 months, starting the year 2020. In order to reach this goal, development processes need to be improved covering all departments within research and development.

This projects has been focusing on the development of engine encapsulations and the aim of improving heat test methods. The results of the projects show a way of replacing a physical heat test with a calculated method. The mathematical method used to calculate the heat transfer is the Fully Implicit Finite Volume Method in one dimension, which has been implemented into a MATLAB script. A graphical user interface has been created and compiled into a Windows standalone software in order to help users with no programming knowledge to be able to use this method and using the software on their workstations without need of MATLAB access. The test can also be executed in an earlier stage of the development process with no need of physical prototypes.

The calculations have been verified by executing heat tests and analyzing the results. A case study was made by comparing an existing test done with the new calculation method. The aim was to measure the time spent on heat testing with the old and new method and the variation of results. According to the case study, the heat test procedure can be reduced from 42 hours to about 1 hour and 20 minutes, which is a reduction of 97%. The average temperature variation between tested- and calculated max temperature was 1.86%. This proves that the new method improves the development process of engine encapsulations.

Keywords: Engine, Encapsulation, Optimization, Heat Transfer, Finite Volume Method, Design of Experiments, Automotive, MATLAB, Process Development, Engineering.

ACKNOWLEDGEMENTS

First of all, we would like to thank everyone at Volvo Car Group, specially group 97596 with Per Burström as group manager, for giving us the opportunity to conduct this project and to make us feel like a part of the group.

Secondly, we would like to thank our examiner Kristina Wärmefjord and supervisor Magnus Bengtsson at Chalmers University of Technology for all the feedback and support during this project.

Last but not least, we would like to thank friends and family for all the support and encouragement to help us through this time.

Martin Jönsson and Daniel Talani
Gothenburg, Sweden 2015

ABBREVIATIONS

VCG	Volvo Car Group
PUR	Polyurethane Foam
PA66GF30	Glass fiber reinforced polyamide
EE	Engine Encapsulation (Group)
FVM	Finite Volume Method
DoE	Design of Experiments
CAD	Computer Aided Design
CFD	Computer Fluid Dynamics
GUI	Graphical User Interface
MCR	MATLAB Compiler Runtime

NOMENCLATURE

Q	Energy	$[J]$
k	Thermal Conductivity	$[W/mK]$
A	Area	$[m^2]$
T	Temperature	$[C]$
L	Length	$[m]$
h	Heat Transfer Coefficient	$[W/m^2K]$
e	Emissivity	[-]
σ	Stefan-Boltzmann constant	$[W/m^2K^4]$
m	Mass	$[kg]$
C	Specific Heat Capacity	$[J/kgK]$
ρ	Density	$[kg/m^3]$
V	Volume	$[m^3]$
P	Node	
I	Time integral over the temperature in node P	
θ	Weight variable	
ε	Length	$[m]$

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background of Volvo Car Group

1.1.1. History

Volvo Car Group (VCG) was born on April 14th, 1927 when the first car "Jakob" left the factory in Gothenburg, Sweden. Founded by Assar Gabrielsson and Gustaf Larsson, the company was formed on a background of quality and safety which were both of paramount importance, a concept that still applies to the Volvo cars of today[1]. In 2009, Zhejiang Geely Holding Group bought VCG from Ford and a new era began.

1.1.2. Goals and Values

Development of cars is a time consuming process that requires a lot of resources[2]. Car manufacturers are trying to optimize the development process in order to minimize the time spent on development. The danger of a long development period is the lack of innovating technology updates in the car. A car could be outdated even before it is available on the market[3]. A shorter development process would result in a faster iterations of new designs that would make the car brand innovative compared to its competitors[4]. The knowledge and experience would be gained in a faster way from earlier projects. It is also hard to follow trends when the development time is long, there is a risk of releasing cars that feel old fashioned. It is also hard to predict future trends. By shortening the time, it would be easier to find a scope that is modern. Newer technology will be available with shorter development times.

VCG has always been a leading car brand in quality and safety but a new vision was born when Geely bought VCG, and that is to become the leading brand in quality and safety as a premium car. In order to achieve that vision, VCG must shorten the development time to get the benefits mentioned above. In order to do that, VCG has a plan called 202020 were the aim is to develop new cars in 20 months in the year 2020. This plan have resulted in a higher priority in operational development, trying to optimize each department and actively work on efficiency in the processes[5].

1.2. Background of Project

There are many departments in VCG that can cut down on their development process in order to make VCG a more competitive brand among the premium cars. One of these departments is Engine Engineering, more specifically, *Engine Encapsulations (EE).* Engine encapsulations are used in a car, covering engine parts, to lower the noise while driving to get a more satisfying driving experience, which is one of the key functions of a premium car, but developing an encapsulation takes too long and that results in delays of the engine development.

In **Figure 1.1**, encapsulations, that are highlighted, cover parts of an engine that generate a lot of noise. The encapsulations are made out of *Polyurethane (PUR*) which is a foam material with good sound absorbing and isolating properties. PUR cannot withstand the high temperatures that some of the parts in the engine room give away and therefore, the heat-exposed surfaces of the encapsulations need to be protected with another material as an outside shell. That material can vary depending on the encapsulation, but for the most of them, *Glass fiber reinforced Polyamide (PA66GF30)* is used as protection against heat.

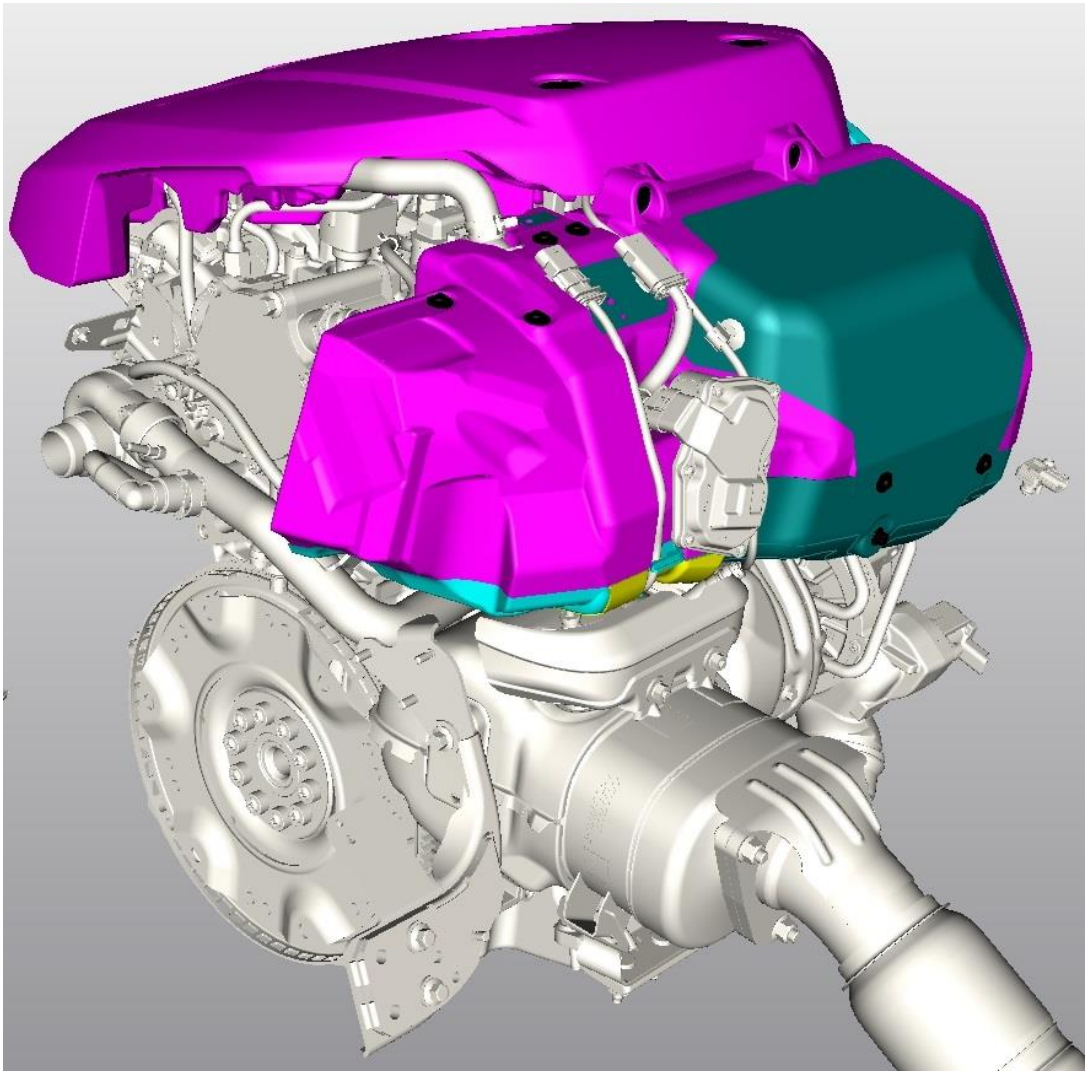


***Figure 1.1**: Engine with highlighted encapsulations*

1.3. Problem Description

Heat tests are required to be done on the encapsulations during the development process. This is to ensure that the sound isolating material, in this case PUR, does not break and loses its absorbing ability. These tests can be made several times on one encapsulation in an iteration process to get a good result. But these tests usually take a long time and cannot be performed until a late stage in the development process because it requires a physical model of an encapsulation and the changes that can be made afterwards are very limited because of the material choice and design have already been done.

1.4. Objective

To get the EE group in the 202020 path, the project will focus on finding methods in the optimization area to develop a user friendly tool, which will be used to reduce steps in the development process.

1.5. Limitation

There are 7-10 encapsulations included in the engines used at VCG. This project will focus on a more general layer based model to calculate the wall design at certain points with different heat impact. Any design suggestions will not be presented, the main focus will be on creating the optimization tool. The tool should be created and documented in a way that simplifies the process of implementing this way of thinking for all encapsulations.

The optimization tool will not replace the "trial-and-error" process entirely, the aim is to reduce steps in the iteration process. The tool will be focused on specific areas of the product and not include the entire geometry in the calculations, this because of the possibility of simplifying the model in order to capture results on these areas.

Depending on the complexity of the calculations, it will not be exactly the same result as in reality. The aim is to find a solution as close to reality as possible and to be able to compare different designs and find pros and cons for each.

The project is a 30 credit course, which is equivalent to 20 weeks full time given all the holidays and breaks.

1.6. Approach

The project was executed for Engine Engineering department at VCG in Torslanda, Sweden.

The pre-study was performed by a literature study and data collection at VCG. To find the user requirements, the different stakeholders will be interviewed. The concept development will be performed by the stage gate method, allowing the project to continue but have to pass different checkpoints in order to the next phase in the development. The software used to develop the tool is not decided yet.

Sound isolation tests and heat transfer tests will be executed in lab environments at Volvo Cars and Chalmers.

1.7. Outline of the report

This report is divided in to chapters that are organized in a structured way to follow the project in a pedagogical way. The chapters are:

- Chapter 2: Theory: Information over relevant subjects used in this project.
- Chapter 3: Methodology: Summary of methods used to execute the project.
- Chapter 4: Results: Valuable information found during the Methodology.
- Chapter 5: Analysis and Discussion: Discussion regarding the results.
- Chapter 6: Conclusions and Recommendations: Recommendation for further use and development of this project.

2. THEORY

Some subjects in this project might not be known to the reader of this report, therefore in this chapter, information about each subject have been gathered to give the reader a deeper knowledge in order to fully understand the project.

2.1. Sound

Sound is vibrations that propagate through mediums in the shape of waves. This waves can be recognized by humans and perceived by the brain as sound. A human can recognize sound with frequencies between 20 and 20.000 Hertz, the upper limit will decrease with age[6]. Sound is transmitted in three ways; airborne, through impact of material and by flanking.

Noise

Aural stimuli are differentiated into two categories; sound and noise. Sound is considered as desirable and tolerable, while noise on the other hand is considered as unwanted and could lead to damage, distraction and discomfort. It is important to categorize sound and noise created in the engine. Some sounds would be interpreted with power and quality, such as good engine sound. There are methods used to minimize the noise; active and passive noise reduction. Active noise reduction is the technology of creating anti-noise, i.e. sound waves with the same frequency as the noise, but with inverted phase. The sum of both sounds will then decrease the amplitude of the wavelength and by that also decrease the decibel produced[7].

Passive noise reduction is to isolate the sound source from the exposed environment. An example of passive noise control is absorption tiles that could be placed in an office to minimize the spread of sound. Passive noise reduction cancels the noise because of the physical geometry and properties of the design. Engine encapsulation is a passive noise reduction method that encapsulates the sound at the source. In the engine encapsulation design it will be the material distribution and thickness that will control the amount of sound energy transformed into other kind of energy, such as heat, when the sound waves travels through the layers.

Physical and mental effects on noise

Being exposed to noise can in both long- and short term lead to human damage[8]. Sound pressure can be hazardous for the ears and result in damages such as tinnitus and decreased hearing ability. Even small amounts of noise can be dangerous when being exposed during a long period of time and car users can be driving for several hours a day. Being exposed to noise is hard to control, because it's impossible to shut of your ears, as it is with eyes when you close them. Noise can also affect the mental ability of operating. Noise can divide the mental resources and be a distraction towards other important stimuli and also increase tiredness. A focused driver is important in order to maintain the safety of vehicle operation. Distractions can result in an extended responsiveness to sudden occurring obstacles and a perception neglecting important stimuli in traffic, which will increase the risk of accidents.

The increased amount of simultaneous stimuli to the driver is important to consider when dealing with noise. The mental workload possible for the user is limited and adding to many stimuli would result in overstimulation, leading to a stressful environment. A stressed driver is dangerous in traffic due to the risk of performing mistakes.

Quality effects on noise

Volvo Cars tries to compete in the premium car category. Customers in this category have high demands on quality of the cars. A car that creates noise is an irritation not tolerable by the customers. Reducing noise level in the car is a highly prioritized objective during the development process and not to aim for decreasing the noise can result in severe consequences in the reputation of the VCG brand. The car industry is an open business with several competitors fighting for the same customers. Unsatisfied customers can easily pick other car brand that satisfies all requirements.

2.2. Heat Transfer

If two objects with different temperatures are placed next to each other the warmer object will heat up the colder object. This is exactly what is happening in the engine room with hot components like the catalyst and colder parts like the encapsulations. Heat transfer, as this physical phenomena is called, is when heat, as a form of energy, is transferred from a normally high temperature object to a lower temperature object according to the laws of thermodynamics[9]. The heat will transfer to the colder object until a temperature equilibrium is met, in other words, when both objects are at the same temperature. Transferring the heat from one object to another object through some kind of medium or material takes time, and depending on what medium the heat transfers through the time will be different. The factor that decides the heat transfer rate in a material is called thermal conductivity. A material with high thermal conductivity will transfer the heat faster than a material with low thermal conductivity. If the warmer object is too hot and warm up the colder object to a temperature higher than the service temperature, the mechanical properties will change in that object and it might no longer be suitable for the original purpose.

Heat transfer can be calculated for steady state or transient heat. This sort of project requires a transient calculation, which implies a differential equation over time.

There are three different ways for heat to transfer from one object to another object, *conduction*, *convection* and *radiation*. The amount of energy, Q, emitting from the warm source can be calculated for each kind of heat transfer, see **Figure 2.1**.

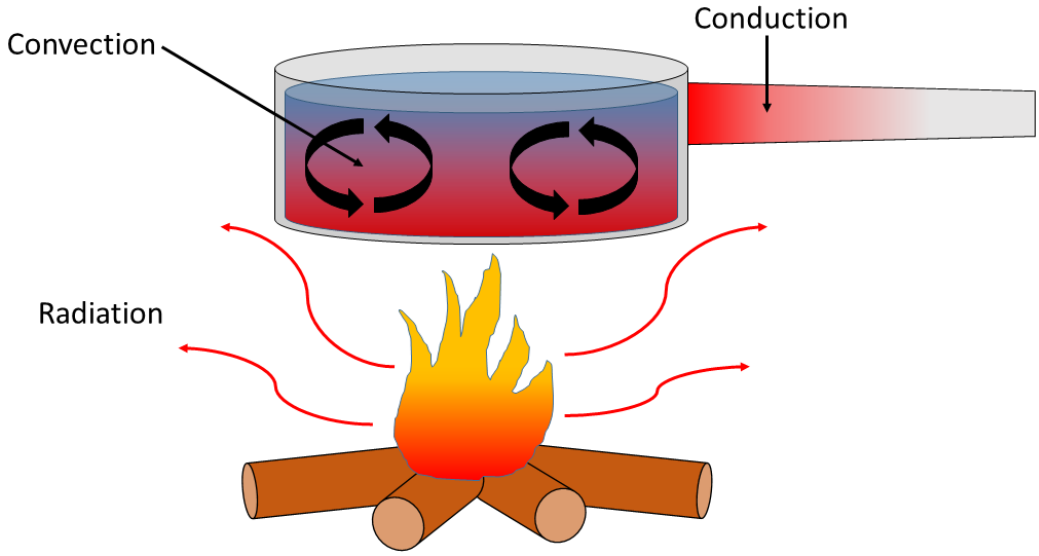


***Figure 2.1**: Conduction, Convection and Radiation explained in one picture*

2.2.1. Conduction

Conduction is when heat transfers across through a stationary solid or fluid medium. The particles in the hot part of the medium have higher speed than in the cold part of the medium and the high speed particles will collide with the slower ones and energy will transfer to the cold part.

The amount of energy transferred from the warm area could be calculated with the formula,

$$\dot{Q} = \frac{k*A*\Delta T}{L} \qquad \textit{(eq. 2.1)}$$

where k is the thermal conductivity constant, A is the cross section area normal to the heat direction, ΔT is the temperature difference and L is the length between the mediums.

2.2.2. Convection

Another way of transferring heat is convection. The heat is transported as energy by fluids such as air or water. The fluid, let's take air as an example, near a hot object will heat up and expand so it will get less dense and rise away from the heat source and the energy will move away with the air to another location.

The steady state equation for convection heat transfer is,

$$\dot{Q} = h * A * \Delta T \qquad \textit{(eq. 2.2)}$$

where h is heat transfer coefficient.

2.2.3. Radiation

Radiation is energy transferred by the emission of electromagnetic waves which carry energy away from the emitting object. An example of heat transfer through radiation is the sun, which transfers heat to the earth through vacuum environment. The steady state heat transfer through radiation is,

$$\dot{Q} = e * \sigma * A * T^4 \qquad \textit{(eq. 2.3)}$$

Where, e is the emissivity and σ is Stefan-Boltzmann constant.

2.2.4. Steady state vs transient heat transfer

A steady state model explains the system in heat equilibrium when the energy transfer is zero, i.e. $\ddot{Q} = 0$. A transient heat transfer model explains the temperature variation over a time period. With constant heat exposure, the transient method will be steady when time goes towards infinity[10].

2.3. Problem Decomposition

Dividing a complex problem into sub-problems and categorizing them into tiers is called problem decomposition[11]. A problem decomposition makes it possible for the user to get an overview of the entire system and simplifies the possibilities of interpreting causes and effects of sub-problems. By decomposing a complex problem, a sub-problem causing a "bottleneck effect" of the system can be found. Solving this sub problem can result in major improvements of the entire system.

Using proper input, a simplified model can be created out of a complex model that represents the real model in its real environment.

2.4. Optimization

In mathematics, optimization is the selection of the best result from a set of variables. An example of an optimization would be to find max- and/or min-values of a formula by finding the best combination of variable values. The mathematical relation of a complex system is hard to find in reality and would demand a mathematical model that requires a set of variables that is almost impossible to find. A tool to find a mathematical relation of a system is *Design of Experiment (DoE)*[12]. This can be used in optimization methods and is further explained in *Chapter 2.4.1*.

2.4.1. Design of Experiment (DoE)

Executing a set of experiments with different settings, the output of the experiment can be set in relation to the settings. Increasing the number of experiments would result in relations that is closer to reality. A full factorial DoE is when all possible settings are tested.
If a test has two variables that each has three different settings, the full factorial model would include $3^2 = 9$ tests. A fractional factorial design is when only some variable variations are combined in order to reduce the number of tests needed. This is possible when the relation between some variables and the output is known or assumed to be of minor importance.

2.5. Finite Volume Method (FVM)

Several existing computer fluid dynamics software, such as COMSOL and ANSYS uses the *Finite Volume Method (FVM)* approach in order to analyze *Computer Aided Design (CAD)* models. The finite volume method is based on transient heat conduction within a mesh. The method is possible in one, two and three dimensions.

2.5.1. Derivation of FVM

Imagine a wall with constant area, density and specific heat and the heat transfer appears in the normal direction of the cross section area. Cutting a thin plate with thickness dx as **Figure 2.2** shows, the energy of the plate can then be expressed as,

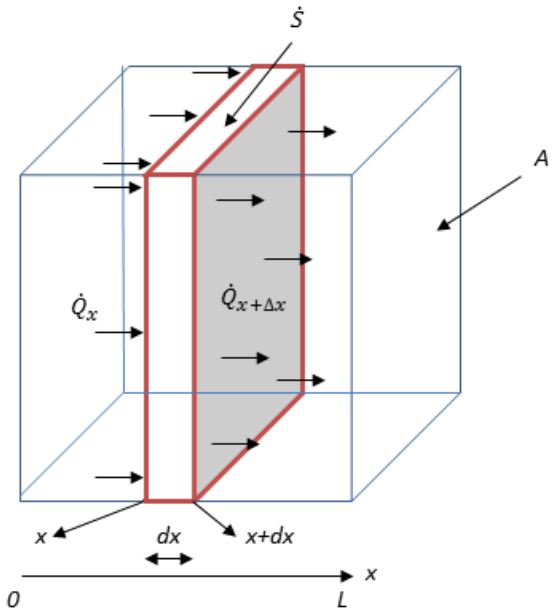


***Figure 2.2**: Heat transfer in a thin plate*

$$\dot{Q}_x - \dot{Q}_{x+\Delta x} + \dot{S}_{element} = \frac{\Delta Q_{element}}{\Delta t} \qquad \textit{(eq. 2.4)}$$

where

$$\Delta Q_{element} = Q_{t+\Delta t} - Q_t = mC(T_{t+\Delta t} - T_t) = \rho CA\Delta x(T_{t+\Delta t} - T_t) \qquad \textit{(eq. 2.5)}$$

$$\dot{S} = \dot{s}A\Delta x \qquad \textit{(eq. 2.6)}$$

Inserting *(eq. 2.5)* and *(eq. 2.6)* into *(eq. 2.4)* and divide with $A * \Delta x$ would result in,

$$-\frac{1}{A}\frac{\dot{Q}_{x+\Delta x} - \dot{Q}_x}{\Delta x} + \dot{s} = \rho C \frac{T_{t+\Delta t} - T_t}{\Delta t} \qquad \textit{(eq. 2.7)}$$

Setting $x \to 0$ and t $\to$ 0 and according to Fourier's law,

$$\lim_{\Delta x \to 0} \frac{\dot{Q}_{x+\Delta x} - \dot{Q}_x}{\Delta x} = \frac{\partial \dot{Q}}{\partial x} = \frac{\partial}{\partial x}\left(-kA\frac{\partial T}{\partial x}\right) \qquad \textit{(eq. 2.8)}$$

(eq. 2.9) would then represent the heat conduction at Δx and Δt.

$$\rho C \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(kA \frac{\partial T}{\partial x}\right) + \dot{S} \qquad \textit{(eq. 2.9)}$$

Unsteady heat conduction in one dimension is governed with *(eq. 2.9)*, were both time t and length x are divided into finite elements in the model. The number of time steps regulates the number of iterations that will be calculated between start and end time. The size of ∂x will create a number of nodes with equal distance between each other through the thickness of the material, see **Figure 2.3** showing a node called *P* with its west node W and its east node E. Each node can then use its own material data, which implies that an infinite amount of materials could be implemented in the code with selected thickness. By adding the number of nodes, the node distribution between materials will then be more accurate to the real scenario. The idea is to let a specific temperature prevail the entire control volume at a specific time period. By minimizing ∂t and ∂x the calculation will be more accurate, but requires more iterations. The aim is to implement this calculation into a computer software, to calculate the large number of iterations efficiently.

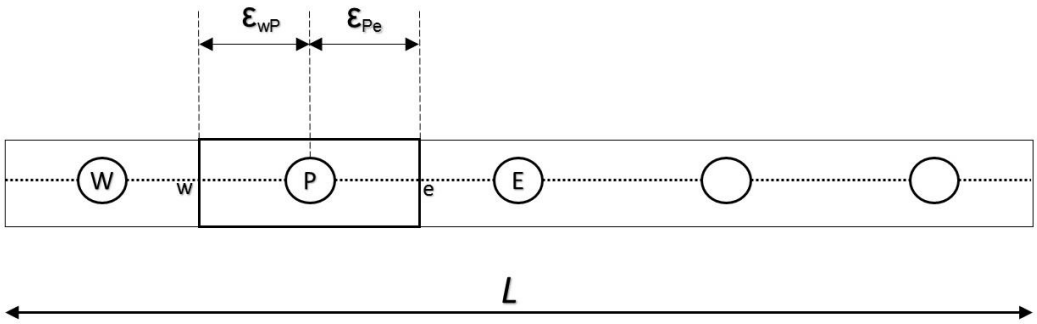


***Figure 2.3**: Length of material divided in five nodes*

Applying the *Control Volume (CV)* from **Figure 2.3** into *(eq. 2.9)* between the time period t and
$t + \Delta t$, *(eq. 2.9)* can be written as,

$$\int_t^{t+\Delta t} \int_{CV} \rho C \frac{\partial T}{\partial t} dV dt = \int_t^{t+\Delta t} \int_{CV} \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) dV dt + \int_t^{t+\Delta t} \int_{CV} S dV dt \qquad \textit{(eq. 2.10)}$$

which can be written to fit **Figure 2.3** as in *(eq. 2.11)*, were A is the face area of the control volume and $\bar{S}$ is the average source strength.

$$\int_w^e \left[\int_t^{t+\Delta t} \rho C \frac{\partial T}{\partial t} dt\right] dV = \int_t^{t+\Delta t} \left[\left(kA \frac{\partial T}{\partial x}\right)_e - \left(kA \frac{\partial T}{\partial x}\right)_w\right] dt + \int_t^{t+\Delta t} \bar{S} \Delta V dt \qquad \textit{(eq. 2.11)}$$

By letting the temperature prevail equally over the entire CV, the left side could be written as *(eq. 2.12)*, were T_P^0 refers to the temperature at time t.

$$\int_{CV} \left[\int_t^{t+\Delta t} \rho C \frac{\partial T}{\partial t} dt\right] dV = \rho C (T_P - T_P^0) \qquad \textit{(eq. 2.12)}$$

The right side of *(eq. 2.11)* could be written as,

$$\rho C(T_P - T_P^0)\Delta V = \int_t^{t+\Delta t}\left[\left(k_e A\frac{T_E - T_P}{\varepsilon_{PE}}\right) - \left(k_w A\frac{T_P - T_W}{\varepsilon_{WP}}\right)\right]dt + \int_t^{t+\Delta t}\bar{S}\,\Delta V dt \qquad \textit{(eq. 2.13)}$$

The right side need to consider the variation of T_P, T_E and T_W with time. The time integral could be calculated using the time t or at time $t + \Delta t$ or even a combination of the both with a specific weight parameter to consider both. The time integral over the temperature in *P* that will be created is shown in *(eq. 2.14)*.

$$I_T = \int_t^{t+\Delta t} T_P dt = [\theta T_P + (1-\theta)T_P^0]\Delta t \qquad \textit{(eq. 2.14)}$$

Several *Computer Fluid Dynamic (CFD)* software uses the Fully Implicit FVM in their calculation, which means that $\theta = 1$. The source term will be linearized as $\overline{S} = S_U + S_P T_P$ This will give $I_T = T_P \Delta t$. Implementing I_T into *(eq. 2.13)* and dividing with $A * \Delta T$ we get *(eq. 2.15)*, deleting $A * \Delta T$ is possible due to the use of point temperature input instead of energy, which is not based on cross sectional area and the iteration time is set to one second in the calculations.

$$\rho C\left(\frac{T_P - T_P^0}{\Delta t}\right)\Delta x = \left[\frac{k_e(T_E - T_P)}{\varepsilon_{PE}} - \frac{k_w(T_P - T_W)}{\varepsilon_{WP}}\right] + \bar{S}\Delta x \qquad \textit{(eq. 2.15)}$$

which can be rearranged into,

$$\left[\rho C\frac{\Delta x}{\Delta t} + \frac{k_e}{\varepsilon_{PE}} + \frac{k_w}{\varepsilon_{WP}}\right]T_P = \frac{k_e}{\varepsilon_{PE}}T_E + \frac{k_w}{\varepsilon_{WP}}T_W + \rho C\frac{\Delta x}{\Delta t}T_P^0 + \bar{S}\Delta x \qquad \textit{(eq. 2.16)}$$

The coefficients of T_W and T_E could now be identified as a_w and a_E and the equation can be rewritten in a simpler form,

$$a_P T_P = a_W T_W + a_E T_E + a_P^0 T_P^0 + \bar{S}\Delta x \qquad \textit{(eq. 2.17)}$$

where

$$a_P = a_W + a_E + a_P^0 - S_P \qquad \textit{(eq. 2.18)}$$

$$a_P^0 = \rho C\frac{\Delta x}{\Delta t} \qquad \textit{(eq. 2.19)}$$

$$a_W = \frac{a_W}{\varepsilon_{WP}} \qquad \textit{(eq. 2.20)}$$

$$a_E = \frac{k_e}{\varepsilon_{PE}} \qquad \textit{(eq. 2.21)}$$

Implementing this calculations into a MATLAB script were done the following way:

Each node will get its own coefficients that follows **Table 2.1**

Node	**a_W**	**a_E**	**S_P**	**S_u**
First node	*0*	$k/\Delta x$	$-2k/\Delta x$	$(2k*T_A)/\Delta x$
The i:th node	$k_i/\Delta x$	$k_i/\Delta x$	*0*	*0*
Last node	$k/\Delta x$	*0*	$-2k/\Delta x$	$(2k*T_B)/\Delta x$

***Table 2.1**: FVM for each node*

Here, $\varepsilon_{PE} = \varepsilon_{WP} = \Delta x/2$. T_A and T_B are vectors containing the temperature at each time iteration at first and last node.

Depending on the boundary conditions on the source terms S_P and S_U are set to 0 if the surface is isolated and has no external heat source. The inside of the encapsulations react on the environment in different ways and needs to be considered for each calculation with either an imported temperature curve for the entire time period or just the initial temperature with a natural temperature variation depending on the surrounding environment.
Using the coefficients set in **Table 2.1** and apply it to *(eq. 2.17)*, a simple equation can be created for a selected number of nodes and time periods, see equation 14. In this case we show a 5 node system, but it could be adjusted to selected quantity.

$$\boldsymbol{M} * \boldsymbol{T} = \boldsymbol{N} \qquad \textit{(eq. 2.22)}$$

where:

$$\boldsymbol{M} = \begin{bmatrix} a_{P1} & a_{E1} & 0 & 0 & 0 \\ a_{W1} & a_{P2} & a_{E2} & 0 & 0 \\ 0 & a_{W2} & a_{P3} & a_{E3} & 0 \\ 0 & 0 & a_{W3} & a_{p4} & a_{E4} \\ 0 & 0 & 0 & a_{W4} & a_{P5} \end{bmatrix} \qquad \textit{(eq. 2.23)}$$

$$\boldsymbol{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} \qquad \textit{(eq. 2.24)}$$

$$\boldsymbol{N} = \begin{bmatrix} a_{P1}^0 T_{P1}^0 + S_{u1} \\ a_{P2}^0 T_{P2}^0 + S_{u2} \\ a_{P3}^0 T_{P3}^0 + S_{u3} \\ a_{P4}^0 T_{P4}^0 + S_{u4} \\ a_{P5}^0 T_{P5}^0 + S_{u5} \end{bmatrix} \qquad \textit{(eq. 2.25)}$$

The new temperature for each node is calculated by $\boldsymbol{T_i} = \boldsymbol{M_i} \backslash \boldsymbol{N_i}$ where $\boldsymbol{T_i}$ is a vector with temperature at each nodes at the time i[1314].

3. METHODOLOGY

Chapter 3 describes the procedure used to execute the project in order to get the desired results for the goal set in the introduction.

3.1. Research

To first understand the problem and gain knowledge to solve the problem, research methods were used to gain as much information as possible. Research methods such as interviews and literature studies were used.

3.1.1. Interviews

Non structured interviews were executed with VCG employees. These interviews were important in order to find information from different stakeholders of the development process. The interviews were unstructured because it was part of the research period of the project, which implied that the interviewers did not have enough knowledge to ask proper questions. The interviews was also a way to improve the relations with the employees in order to ask questions and get help during the entire project.

3.1.2. Literature Studies

Research data was collected through a literature study with existing course books, articles and journals that cover this kind of framing of questions. Relevant information from the interviews were looked up in different literature and analyzed for further use in the project.

3.2. Calculation

The solution for the problem was to find a suitable mathematical model that represents the physical test, which was accurate enough to present output for the encapsulation designer to create a good model.

3.2.1. Heat Transfer

A proper calculation model that represents the existing model and environment was selected from the gained knowledge from the research study. This method was then transferred into a MATLAB script in order to handle heavy calculations in an organized way. The final calculation method was selected using a heat test to understand the physical phenomenon that appears in the engine environment.

Regular steady state heat calculations was not enough in this project because of the time variable and therefore the *Finite Volume Method (FVM)* was selected to get the right prediction of the heat passing through different layers of material over time. More about the theory on FVM can be found in *Chapter 0*.

3.2.2. Sound Isolation

Even though heat is the main problem in this project, a method for optimizing with respect to heat only was not suited for the encapsulation, so the sound had to be included to get as good noise reduction as possible with protection against heat.

After interviewing different sound experts at VCG and researching literatures in the subject, a decision was made that a calculated model from a mathematical method similar to FVM will not be enough to get a realistic behavior of the sound traveling through material. A different optimization method was used to develop a model for the sound absorption and isolation based

on testing. Samples from the materials PUR and PA66GF30 in different thicknesses were put together in many combinations and put in a tube to test absorption and isolation for each combination. The result was analyzed by using *Design of Experiments (DoE)* and a mathematical model was formed after that.

3.3. Validation

Results created from calculation with a simplified calculation model is risky, because it could present output that does not represent the real scenario. Tests was performed in order to compare calculated results with results measured from experiments. This difference should then be analyzed to gain knowledge of the model- and environmental behavior and to calibrate the calculations to minimize the variations. Two kind of validations was executed, one material test executed as a part of the thesis work at the Material Center, VCG, and a case study of an existing test, explained further in *Chapter 3.3.1* and *Chapter 3.3.2*.

3.3.1. Material Center Test

Material plates with dimensions 300 mm in height and 300 mm in width was ordered with a variation of material and thicknesses. These plates was used in a material lab and was exposed to heat in a climate lab with heat guns. The tests was measured with thermo-elements that will measure the temperature at a specific point every second. This data was transferred into a software program that records the temperatures from the thermo-elements at every seconds. These recordings was compiled into graphs that is easy to interpret. The material plates was assembled in different combinations to resemble the material variations in encapsulations.

The equipment that was used in the test at Material Center was, a heat gun model Leister Hotwind 9c1 3400 W and thermo-elements of Type K that measures approximately 41 microvolts per degree Celsius. The range of temperature it can be used in is from minus 200 degree Celsius to 1350 degree Celsius. ComTest, which is a software used in the Material Center at VCG, was logging the temperature from the thermo-elements every second.

The testing process was then executed with thermic behavioral trends learned from studying engine encapsulations in its real environment in a car during a natural driving scenario. The heat guns was adjusted manually to control the energy amount exposed on the test material. One heat gun was simulating the outside temperature caused by other engine parts surrounding the encapsulation and another heat gun was pointed inside a square tube and that tube was located on the backside of the test material. This to simulate the heat generated from the part that the encapsulation will cover.

3.3.2. Case Study

To verify the reliability of the optimization model, an existing test procedure for encapsulations was used and then executed with the code. Ten tests have been executed with different glass fiber combinations, see **Figure 3.1**. The tested encapsulation was the one covering the Bypass Resonator and the reason for testing was to analyze the effect of how different glass fiber thicknesses would affect the temperature on the polyurethane material, see **Figure 3.1**. Both time spent on each process and the difference in result were analyzed. The test were executed with the same method as the tests in *Chapter 3.3.1*.
The test started with a constant temperature on the outer and inner surface to create a heat equilibrium in the material. Next step was to start log the temperature from the thermo-elements

during a 25 minute period. When the test ended, the material needed to cool down for about 2 hours and the glass fiber was replaced.

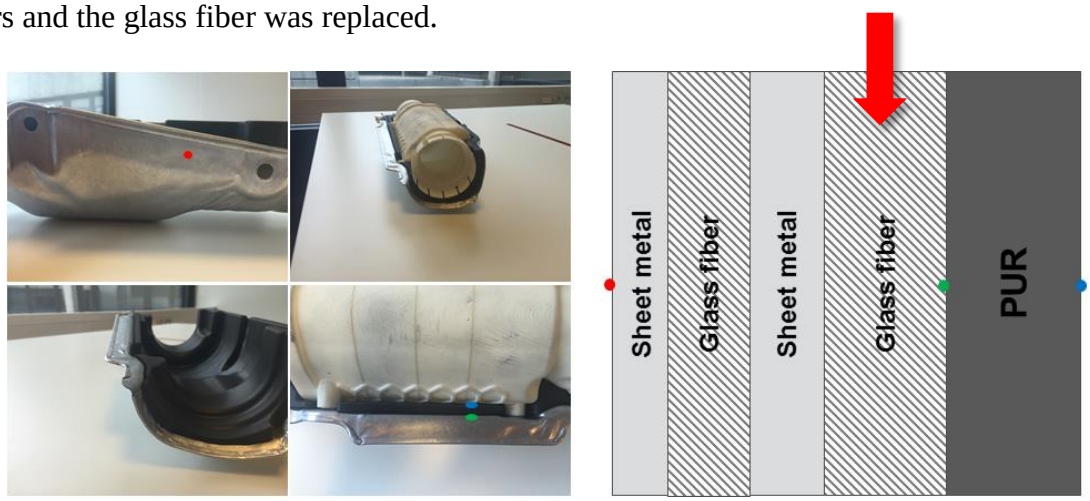


Figure 3.1*: Bypass Resonator Encapsulation and Cross Section view over test area. Red arrow shows the material layer that changes during each test.*

Calculation procedure

The calculated process was executed with the same boundary conditions as the case study. The results from the case study was unavailable during the test period to ensure an unbiased procedure during the calculated tests. The same kind of output as for the existing test was created to make it possible to measure the difference in precision and time in an accurate way. The aim was to use the calculations in the exact same way as it would have been done if the existing test never had appeared.

3.4. Optimization Tool User Interface

Handling code is both complicated and risky. The access to the entire script could result in changes made that destroys the functions. An interface was created to limit the access to the raw code and to simplify the usage, which will make it possible for users with small or no coding experience to use this optimization method.

3.4.1. Graphical User Interface (GUI)

The interface will be programmed using the MATLAB plug-in *Graphical User Interface (GUI)*. To develop an interface that fits the requirements list, a brainstorming session was performed with equipment such as papers, post-its and pens. The aim of the brainstorming session was to create concepts of interfaces that solve the requirements and help the users in their interaction with the software. These concepts was then analyzed and the best parts of the concepts was collected into a programmed interface, presented to the users at VCG. Their feedback was interpreted and resulted in an improved interface. This iteration process was proceed until the users were satisfied with the result. The raw codes functions was transferred into the GUI to remove the need of accessing the raw code.

3.4.2. MATLAB Compiler Runtime, MCR

All of the stakeholders of this software will not have access to MATLAB at their workstations. The code was compiled into a Windows standalone software using the free MATLAB plug-in *MATLAB Compiler Runtime, MCR.*

3.4.3. Key User

One or a couple of key users was selected within the department of engine encapsulations. The reason for choosing key users is to educate them into the software in order to be able to learn new users when the project ends and the access to the developers will be gone.

3.4.4. Manual

The functions of the software will be explained in a manual, to ensure the independence of usage and to store the information within the company in a documented way. The manual will also explain the procedure of handling and storing data in an organized way, to avoid differences between different user actions.

4. RESULTS

In this chapter, the results from the used methods will be presented along with relevant figures.

4.1. Research

In order to fully understand the problem and how to solve it, interviews were made with different stakeholders and literatures were reviewed.

The first stakeholders being interviewed were the group of engineers working at the Engine Encapsulation department at VCG. The engineers explained that the PUR material used to lower the noise in the engine does not withstand higher temperatures that coming from other surrounding engine parts. Therefore, other materials are used as protection to cover the PUR foam. But before the encapsulation can be produced and assembled into a car engine in the factory it needs to be tested in the Material Center and Wind Tunnel. In the Material Center, a heat gun with high temperature is aimed at the encapsulation and then the PUR surface is analyzed. To get a more accurate result, the whole engine is assembled with all parts, including the encapsulations, and tested at different driving scenarios in the Wind Tunnel. If the encapsulation fails to meet the temperature requirements, it has to be redesigned and go through all these tests again. Unfortunately, all these tests take a long time and the Encapsulation group fails to deliver on time.

The engineers at the Encapsulation group also explain that the largest risk for a temperature too high for the encapsulation occurs during the heat soak period. The heat soak period is the time after a car have been driven in a harsh scenario and then stopped really quick and turned off. When the car stops and turns off there's no natural air flow cooling the engine from the generated heat.

To fully understand this heat soak period, an interview with the Thermodynamic group in the Wind Tunnel is conducted and the explanation is shown in **Figure 4.1**. The figure represents a driving scenario in the Wind Tunnel where the car first runs at the speed of 110 km/h as shown in section A of the figure. The different curves are temperature measured on various measurement points on encapsulations. After the car has been driving for a while, it is turned off and the fans turn on to cool the engine, and that is section B in the figure. Now the heat soak period is the period after the fans turn off, section C, and that is the worst period for the encapsulations due to the high temperatures.

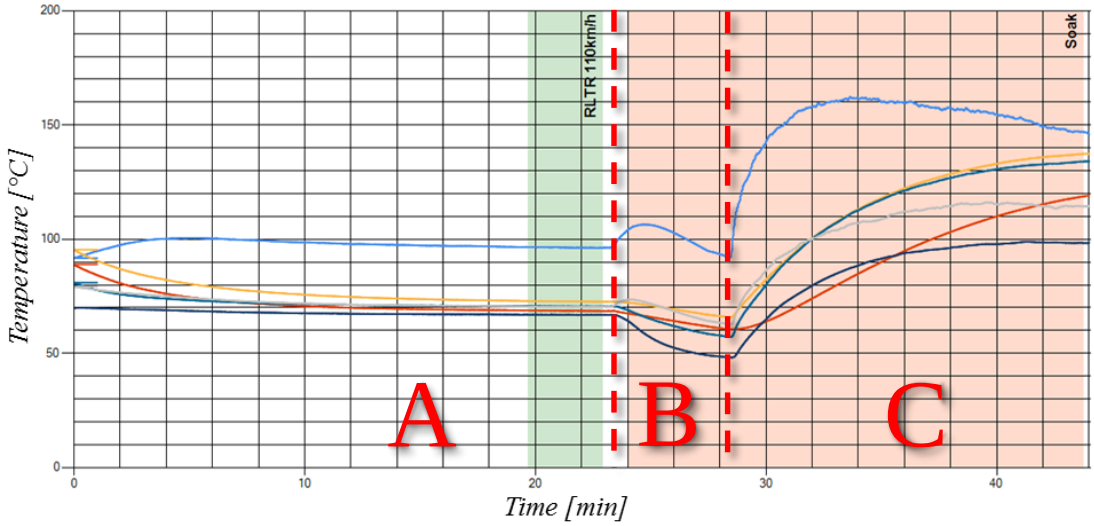


***Figure 4.1**: Plot from Wind Tunnel test over a driving schedule with SOAK*

After the interviews at VCG, the aim was to create a mathematical tool for the heat tests performed by the group, but one important information from the Thermodynamics group in the Wind Tunnel was that simulating a whole heat test in a computer is almost impossible to do because of the variance in how the heat travels from the different components. The EE group works with creating encapsulations and are not experienced in setting up 3D simulating environments so after a discussion with the group, it was decided that the heat testing tool should be simplified to a one dimensional layer based model. In that way the group will have a tool to use when designing an encapsulation that guides them to a design that can lead to good results in the heat test right away.

To find out how to design this simplified mathematical tool, consultations with experts at Chalmers University of Technology in the areas Optimization and Thermodynamics were booked. The information gained from the thermodynamic expert was to use a mathematical model name Finite Volume Method, which is the math behind computer simulated fluid dynamics used in several commercial software. But that math is not easy to handle in an optimizing method. To get a better understanding in how a method can be created with FVM, an optimization expert at Chalmers explained how the math can be integrated in MATLAB to run many calculations in just a couple of minutes.

The aim now was to create a mathematical optimization tool using FVM to calculate the heat transfer over time through the material covering the PUR encapsulation. The FVM uses the temperature on the outer surface of the covering material and the temperature on the inside surface of the encapsulation generated from the part being covered. A wish from the group was to see how the sound absorption and isolation properties would be when setting up a model with different materials and thicknesses in the optimization tool, but interviews with Noise, Vibration and Harshness group at VCG made it clear that there is no possibility of simplifying a mathematical model for the sound transfer through the material. Even though the math for sound could not be done, there was another way to generate an equation for the sound behavior, and that was through a method called DoE, explained in *Chapter 2.4.1*.

With the knowledge gained from the research, a requirements list could be formed, see **Table 4.1**. This list will work as a guide to form the optimization tool.

	Requirement	D/W	Validation
1.	Optimize against heat	D	Testing
2.	Optimize against sound	W(3)	Testing
3.	User friendly tool	D	Let a group member use a beta version of the tool.
4.	Work on materials used today by the group	D	Test calculation for specific material data
5.	Work on all materials	W(5)	Test calculation for specific material data
6.	Trustworthy	D	Compare with old heat test.

Table 4.1*: Requirements List where D = Demand and W =Wish, weighted from 1-5*

4.2. Calculation

4.2.1. Heat Transfer

A decision was made to use the Fully Implicit FVM in one direction to calculate the heat transfer. The external heat sources are simplified as explained in *Chapter 2.5.1*, which leads to a need of using existing external point heat in the inner- and outer surface. The step size is controlled by adjusting the number of nodes needed in the calculation, the step size is then the total thickness divided with the number of nodes selected.

The matrix **TT** will include the temperature for all nodes (columns) and for all time-steps (rows). This matrix will find the vector T_i by using the vector T_{i-1} and execute the FVM calculation. The new vector T_i will then be used to get the values of the vector T_{i+1} and this will continue until the last time iteration is reached.

Boundary Conditions

The first and last nodes have known temperature values because it represents the surface on the inside and outside of the model and these vectors are available as T_A and T_B. To be able to start the calculations, T_i needs to be set when $i = 0$, this is the initial temperature when the calculation starts, see *(eq. 4.1)*, T_1 is then selected to be a linear relation with T_{A_1} as start value and T_{B_1} as the last value.

$$\boldsymbol{TT} = \begin{bmatrix} T_{A_1} & T_{1,1} & T_{1,\dots} & T_{1,m} & T_{B_1} \\ T_{A_\dots} & 0 & 0 & 0 & T_{B_\dots} \\ T_{A_n} & 0 & 0 & 0 & T_{B_n} \end{bmatrix} \qquad \textit{(eq. 4.1)}$$

Matrix **TT**, shown in *(eq. 4.2)*, where all time iterations n have been calculated for all nodes m.

$$\boldsymbol{TT} = \begin{bmatrix} T_{A_1} & T_{1,1} & T_{1,\dots} & T_{1,m} & T_{B_1} \\ T_{A\dots} & T_{\dots,1} & T_{\dots,\dots} & T_{\dots,m} & T_{B\dots} \\ T_{A_n} & T_{n,1} & T_{n,\dots} & T_{n,m} & T_{B_n} \end{bmatrix} \qquad \textit{(eq. 4.2)}$$

The code is split up into two methods, the multilayer calculation and the two-layer optimization. The MATLAB code is available in Appendix B.

Multi-Layer Calculation

By choosing the number of layers in the system between two and five and the thickness of each layer, the code automatically converts the thickness of each layer to a number of nodes that represents the distribution of materials. The code calculates the matrix **TT** and presents the temperature at each time setting up to the last material.

Two-Layer Optimization

This calculation is limited to calculate to only two layers of materials. By selecting the maximum allowed temperature that the last material could be exposed to and the maximum allowed thickness of the system, the code calculates the thickness distribution that allows a maximum temperature of the last material closest to the inserted maximum allowed temperature. The temperature curve is then presented for this material distribution at each time period. The tool works in such way that the last material used for sound isolation should be as thick as possible.

4.2.2. Sound Isolation

***Figure 4.2**: Sound samples measured for DoE*

To find the equation describing the sound transmission through the materials, a DoE were conducted at VCG Sound Lab. The Lab equipment can be seen above in Figure 4.2, where the material sample used were the small puck shaped pieces. The materials were put in the tube and sound from 500 to 5000 Hz was generated from one end and measued on both sides of the test subject. All materials combined gave a full factorial DoE with 60 number of tests. The measured result from the tests is shown as the red curve in **Figure 4.3**. This figure show the results from one of the 60 combinations but the behavior is similar in all tests. The figure also shows a blue curve that is the result from creating a formula that describes the mathematical relation for this combination from 500 to 5000 Hz.

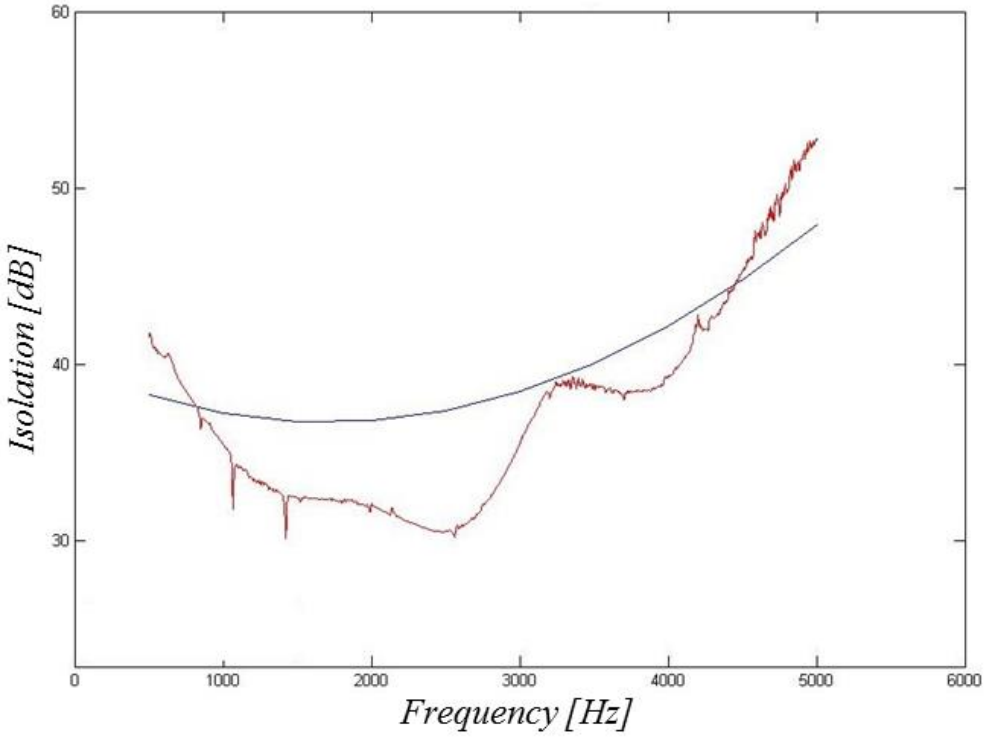


***Figure 4.3**: Sound behavior where the red curve is the result from DoE and the blue curve is the calculated result*

Together with the Encapsulation group, the decision were made to exclude the sound calculation from the Optimization Tool because of following reasons:

- The calculated curve does not represent the DoE curve in an accurate way, see **Figure 4.3**.
- The DoE tests were only performed with PA66GF30 and PUR, does not match with heat calculations that can handle all types of material.
- Flaws in the Sound Lab test that makes it unreliable. Failures such as sound leakage through material samples were found.

4.3. Validation

4.3.1. Material Center Test

There were 21 tests performed in the Material Center lab with different combinations of material and heat exposure. Test 1-4 were performed with thermo-elements mounted to both the plastic side and the PUR side of the transition surface. This was to evaluate the difference in measured results depending on alternative ways of placing the thermo-elements. **Figure 4.4** shows the measured temperatures from Test 2. In all four tests, the thermo-element placed on the plastic surface showed higher values compared to the thermo-elements placed on the PUR surface. The different values from the four first test made it clear that the testing procedure needed to be improved in order to create reliable results.

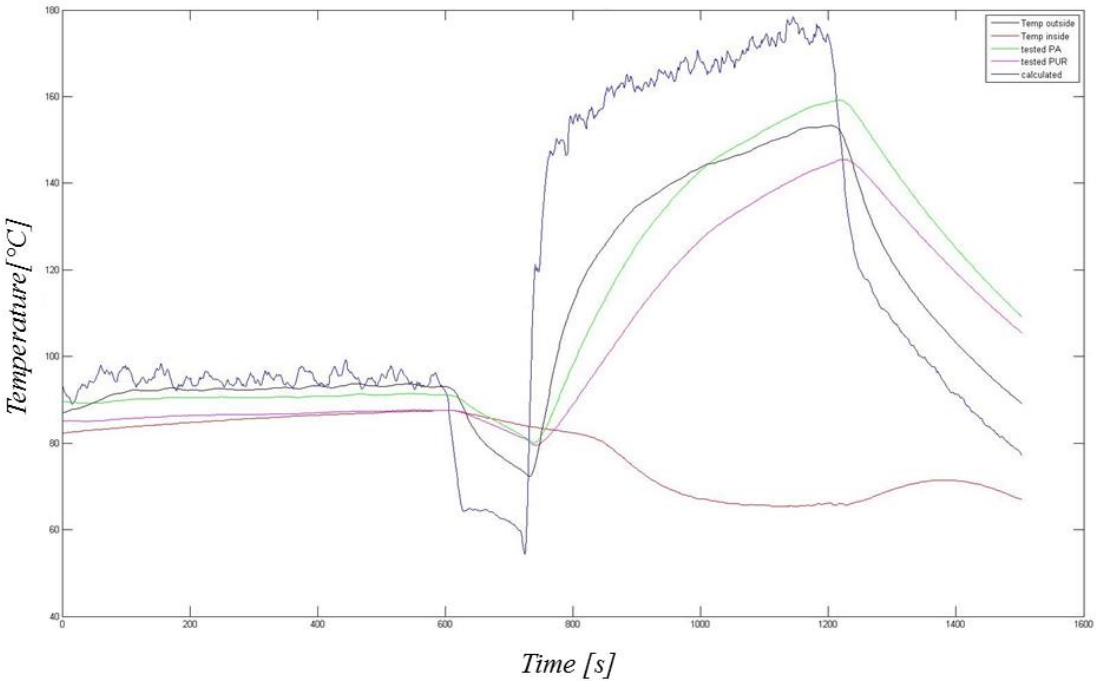


***Figure 4.4**: Plot of Test 2*

At the fifth test, the materials were adjusted to improve the placement of the thermos-elements. By cutting channels in the material, the wires from the thermo-elements were now not creating any air space between the layers. The backside were also covered with aluminum foil to protect the test material from the environment of the climate box. **Figure 4.5** shows the graphs from Test 5.

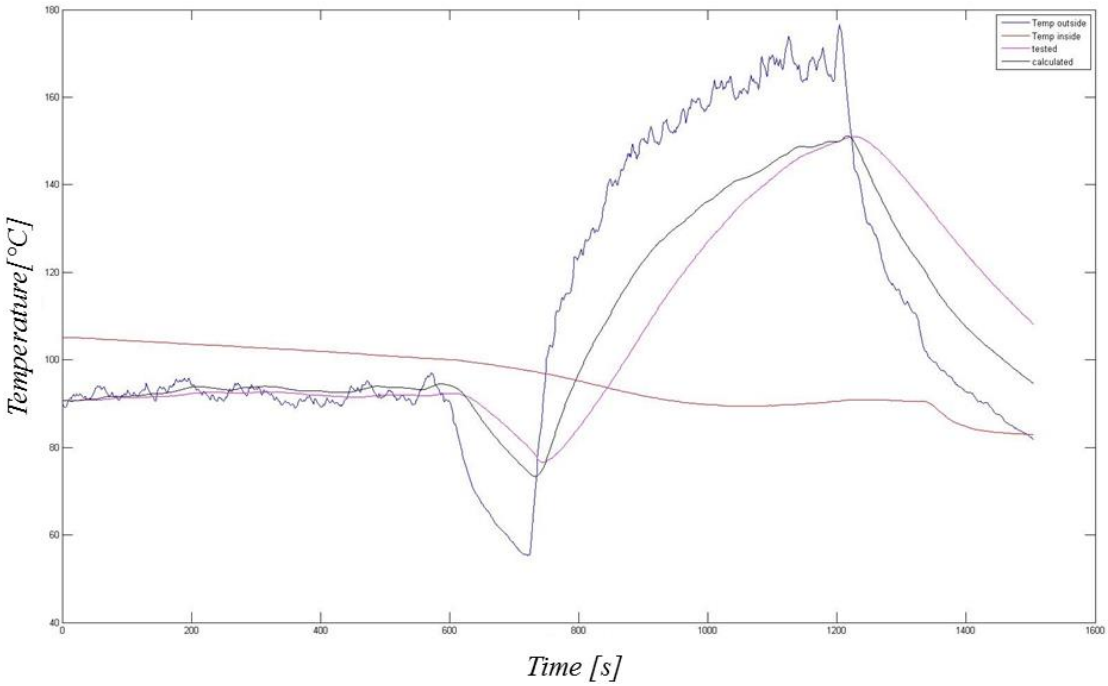


***Figure 4.5**: Plot of Test 5*

Test 9-11 were performed with two kind of thermos-elements; the same as before, and a thicker one. Results from Test 10 show that the new wire did not react as fast as the former wire. The new wire had a bigger diameter and would present values with lower values, but would withstand higher temperatures. While this test did not damage the first thermos-elements, a decision was made to continue with this test equipment.

The tests showed the importance of isolating the materials from the environment to minimize the energy spreading in the surface direction. In reality, there is no air channels between the layers. The tests also showed the result of using a point directed heat source to represent the engine environment. The outer surface emitting the heat will be exposed to a heat maximum in the center of the gun direction and the temperature will decrease in the rest of the material plate. Heat transfer appears between temperature variations, and the test environment indicates high variation in temperatures. The real environment will also result in temperature variation on the outer surface, but it will not be as big as in this test environment.

By comparing the test results with the calculated results, an understanding on the limitations of the calculations were made. The mathematical model used for calculations is one dimensional and isolated in the surface direction, which means that no energy will disappear from the model. The spread of energy through the outer surface is not a part of the calculations and will create deviations from reality when heat variation is big on the outer surface. This test is not suitable for analyzing the behavior on a real product, due to its simple plate geometry and the difficulty of isolating the test rig from external disturbances.

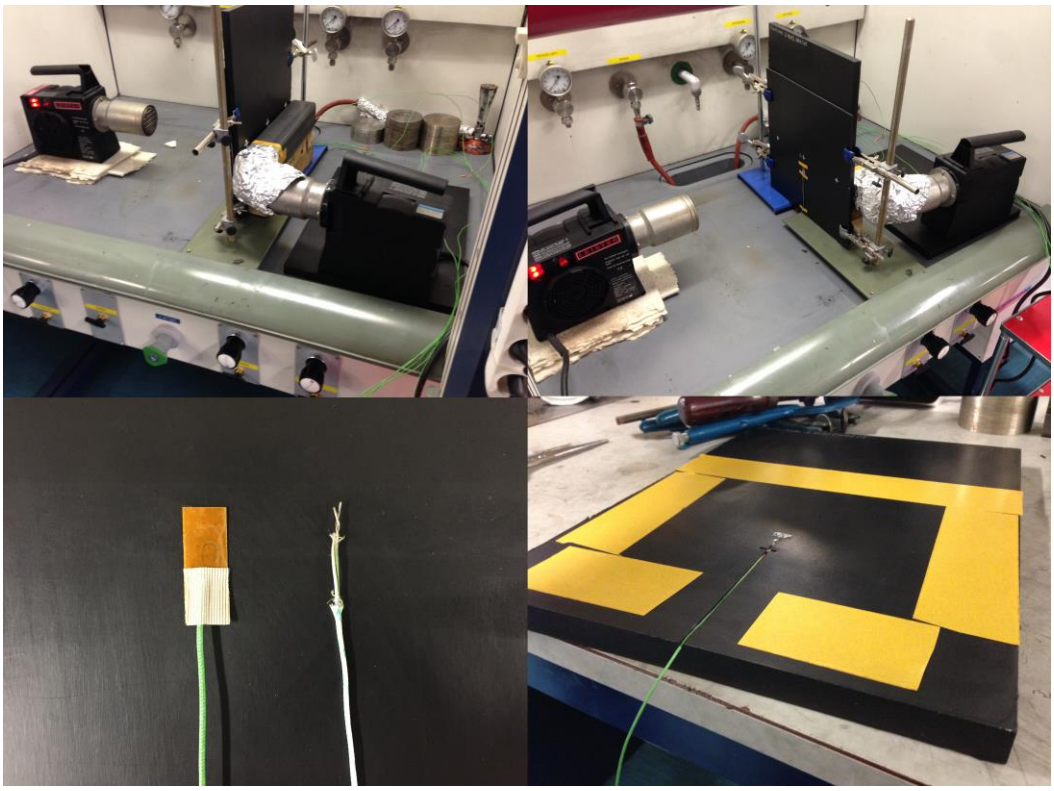


***Figure 4.6**: The two pictures above show the test rig with material in the lab. The lower left picture shows the two kinds of thermo-elements and the lower right shows channels cut out for the thermo-elements.*

The tests are available at Appendix A.

4.3.2. Case Study

Figure 4.7 shows a graph of Case Study Test 1, showing the temperature variation over time for calculated and measured method. The slope trends are similar, but the calculated values increases and decrease in a higher velocity than the measured ones, but they will still have maximum temperature close to each other. This trend appears in all ten tests.

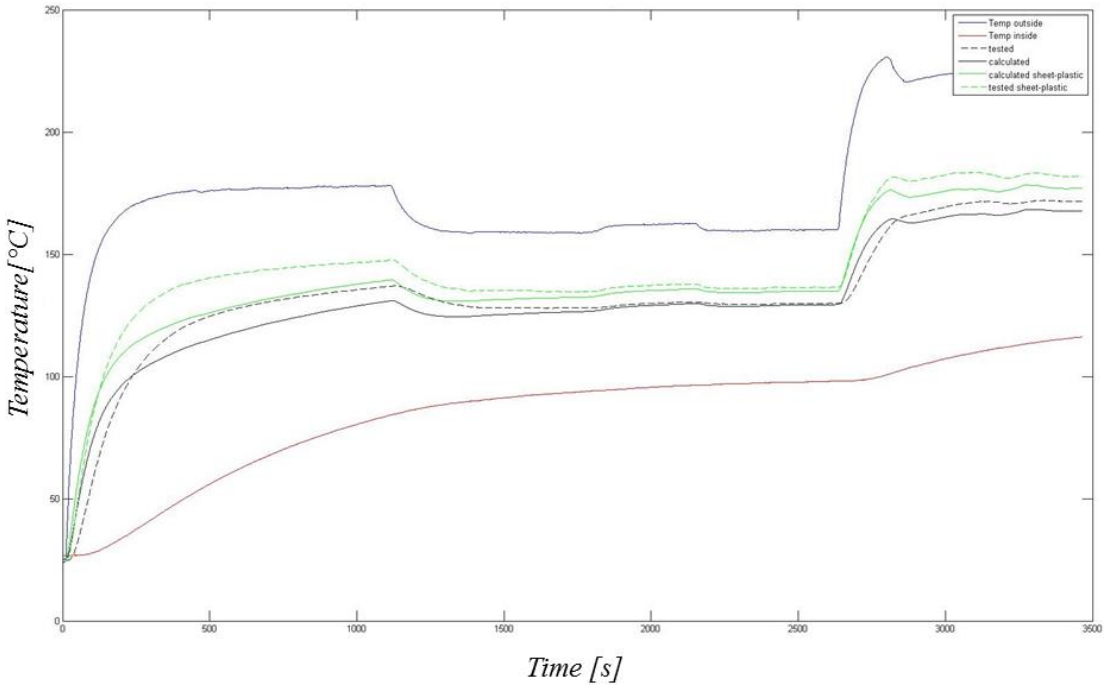


***Figure 4.7**: Plot of Test 1*

Case Study Test 5 and 6 show the tests where the thickest glass fibers were used. The test material were squeezed into the component, which resulted in compressed material. The thickness were then decreased and the material became more compact. Case Study Test 5 and 6 showed the largest deviation between calculation and test, which could be explained by the fact that this compression phenomenon were neglected in the calculation method. This proves the importance of analyzing each test specimen and to be careful when setting input values in the calculation model.

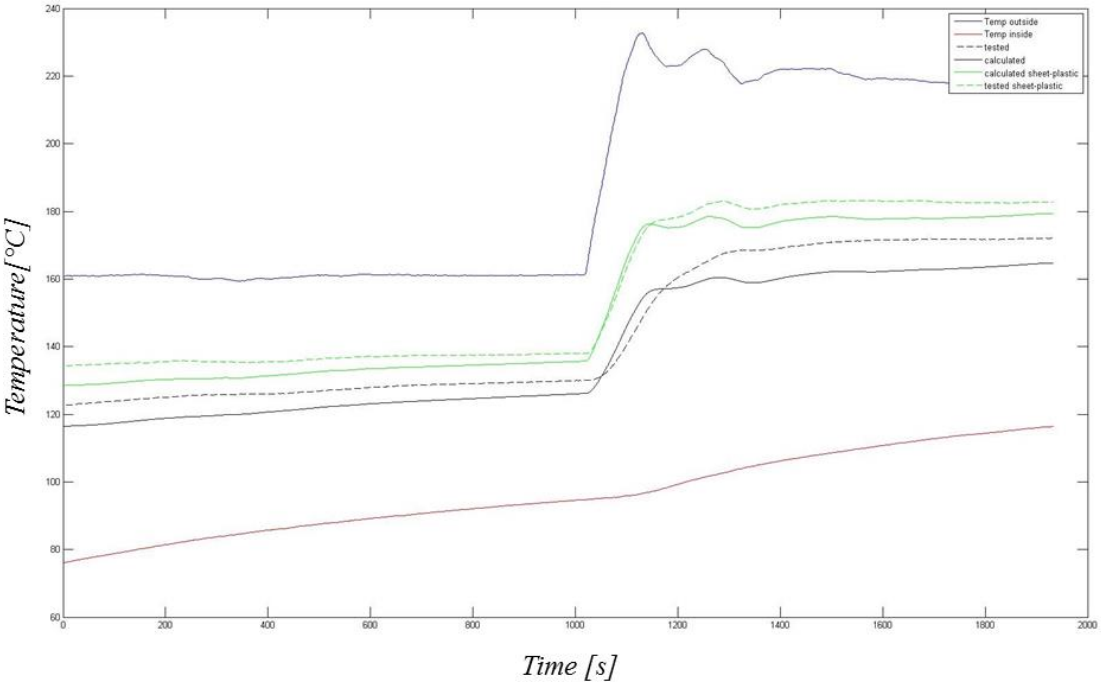


***Figure 4.8**: Plot of Test 5*

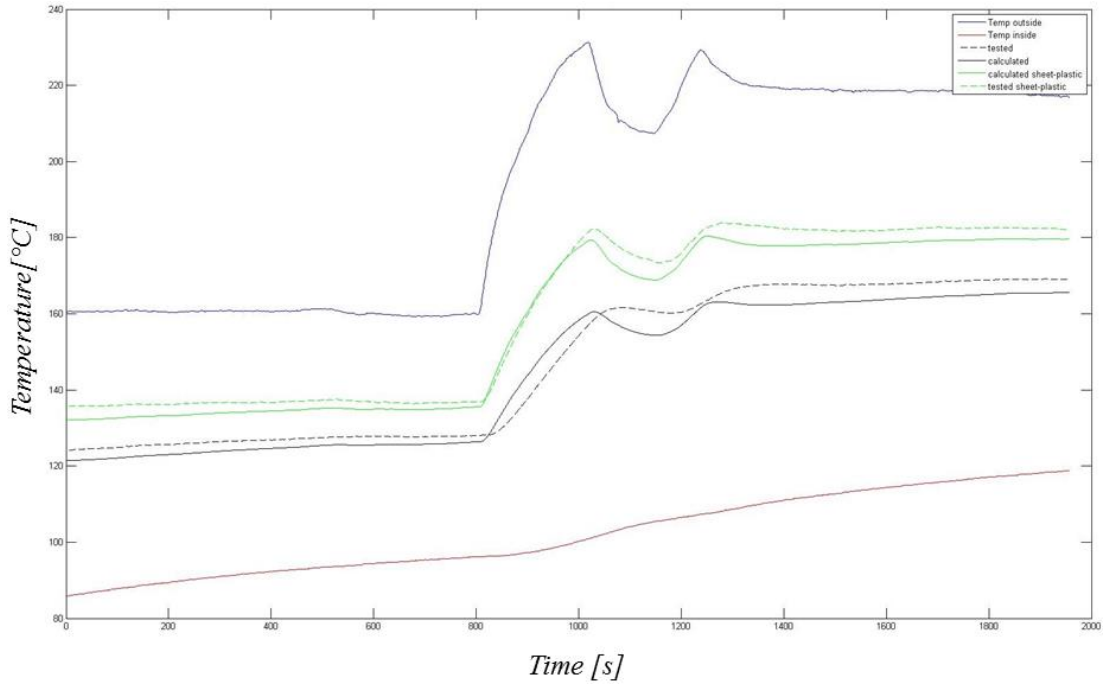


***Figure 4.9**: Plot of Test 6*

The time needed to calculate the ten tests was 1 hour and 20 minutes. Comparing to the 42 hours needed for the real test, a reduction of 97.21 % was made, see **Table 4.2** and **Table 4.3**. The mean difference between calculated and measured maximum temperatures of the ten heat tests were 3.13 °C, which is 1.86 %. See **Table 4.4** for the results of each test.

Testing using real material or prototype in lab (10 Tests)	
Order/Prep. Material:	4 h
Preparation Test Rig:	3 h
Performing Test:	10 h
Cooling Between Tests:	20 h
Compiling Data:	5 h
Total:	**42 h**

***Table 4.2**: Time for ten tests in Material Center*

Testing using Optimization Software (10 Tests)	
Order/Prep. Material:	0 h
Preparation Test Rig:	0.5 h
Performing Test:	0.33 h
Cooling Between Tests:	0 h
Compiling Data:	0.33 h
Total:	**1.17 h**

***Table 4.3**: Time using Optimization Software for ten tests*

Test #	**T max Tested, °C**	**T max Calculated, °C**	**ΔT, °C**	**% ΔT**
1	172.1	168.3	3.7	2.2 %
2	160.8	160.6	0.2	0.1 %
3	161.1	157.2	4.4	2.4 %
4	162.7	160.8	1.9	1.2 %
5	158.0	151.6	6.4	4.1 %
6	172.1	164.7	7.4	4.3 %
7	169.2	165.6	3.6	2.1 %
8	162.5	162.7	0.2	0.1 %
9	161.8	162.8	1.0	0.6%
10	161.3	163.8	2.5	1.5 %

***Table 4.4**: Results comparing real test and calculated test*

4.4. Optimization Tool User Interface

One demand from the group was that the optimization tool should be user friendly, and to work with someone else's MATLAB code can be very hard for someone who have none or very little MATLAB skills. As mentioned in Chapter 3.4.1, MATLAB have a plug-in that allows a person to transform code to a Graphical User Interface, GUI.

But to make sure that the Optimization Tool is represented in the GUI in an accurate way, a brainstorming session with the Encapsulation group was done to generate as many ideas as possible for functions to be used in the software layout. After gathering ideas from the group, A3 posters with a mixture of the functions were made.

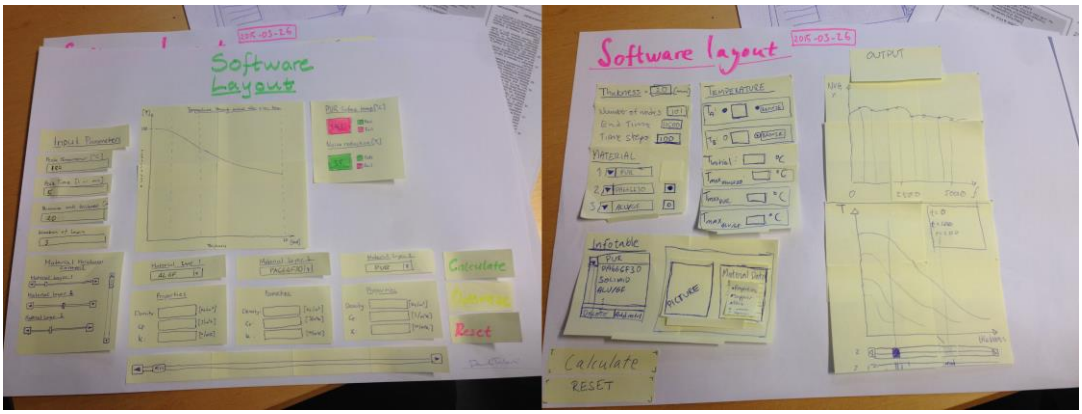


***Figure 4.10**: Brainstorming session for software layout*

These concepts ideas were screened using the requirement list in order to find the most suitable interface. The remaining functions are put in to the GUI plug-in and a software design is slowly starting to form. All the buttons and other functions in the GUI have to work together without any crashes so a lot of coding to link the buttons together had to be done. A name is also decided for the Optimization Tool, Volvo EEO, which is short for Volvo Engine Encapsulation Optimizer.

The layout of the functions is put in a way that the user cannot use the software wrong. The user have to follow a specific path to get the results calculated. This is to prevent errors in the code that a person with no coding experience wouldn't be able to solve.

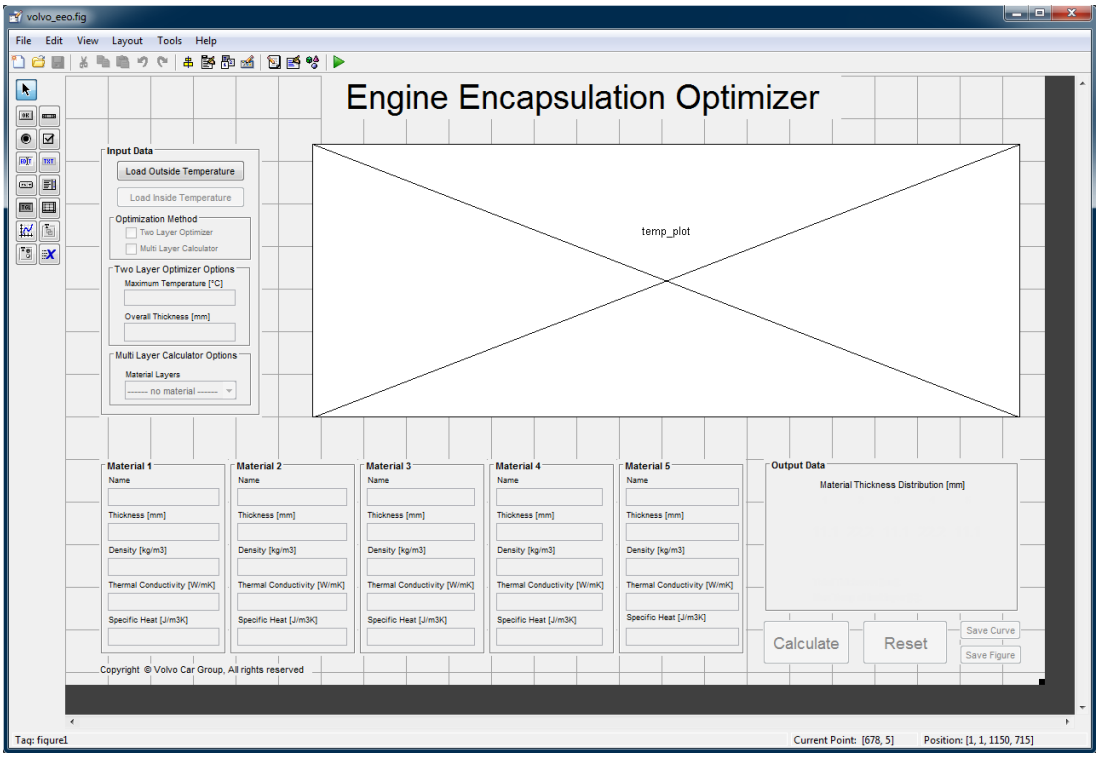


***Figure 4.11**: Designing software layout in MATLAB GUI*

The user was to first load the outer surface temperature curve and then the inside temperature curve. Then the user is given two options for Optimization Method, Two Layer Optimizer or Multi-Layer Calculator, and depending on what the user chooses, several functions open up in the software and the user have to fill in the material properties. The user can enter properties for any kind of material, so the wish for that requirement is fulfilled. When all the input data is entered in the software, the user simply press calculate. A figure with the graphs for the outer temperature (red), inner temperature (blue) and the temperature on the last material surface (green) will be plotted over time. A black dashed line will also show the maximum temperature on the surface for the last material. The same output can be view in the lower right corner of the software but in a scheme form with the total thickness and maximum temperature at the last material surface. A material thickness distribution will also be available and where the different temperature curves occur on the wall section, see **Figure 4.12**.

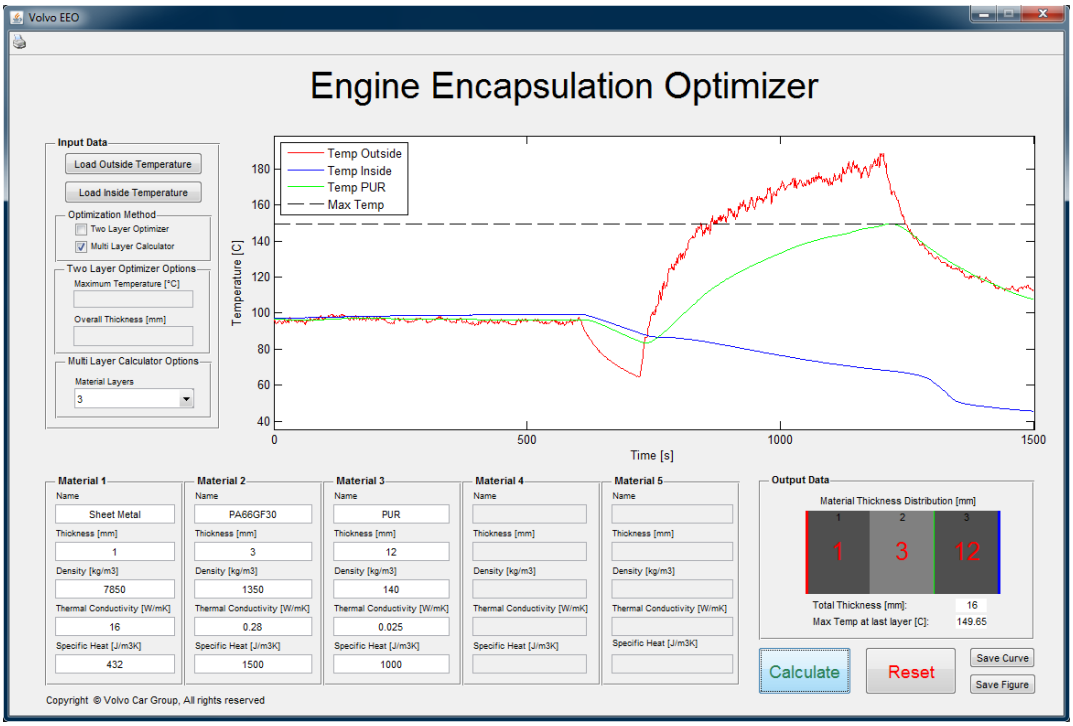


***Figure 4.12**: Volvo EEO software with final design and after compiled with MCR*

The GUI will only work if MATLAB is installed on the computer but there is another plug-in that allows MATLAB to convert the GUI into a standalone software. This plug-in is called MATLAB Compiler Runtime, MCR and turns the MATLAB code in to a software that can be installed to any other computer that doesn't have MATLAB installed. This is a necessary step because the engineers at the group do not use MATLAB but should be able to work with the Optimization Tool.

At first a key user from the group was selected to learn how the Optimization Tool is used but the decision to have a manual was made so that anyone can understand how the software should be used. The manual explains step by step how to get the desired results to use in the design process. A copy of the manual can be found in Appendix X.

5. ANALYSIS AND DISCUSSION

The most difficult part during the research period was to find the right people with knowledge within subjects necessary for the project. The limited knowledge in the beginning of the project made it hard to understand what to search for and how to know what kind of departments that needed to be contacted.

Some research studies lead to a dead end, such as the sound transfer calculations. This study took a long time and lead in the end to a decision to not include it in the project. This research could be seen as a waste of time, but was necessary in order to make a decision about it. By contacting an expert in sound and vibration early in the process, maybe this conclusion could have been reached using less time and effort.

The first scope was to include both heat and sound in the optimization tool. But the simplified calculation model was not including enough input data to contain reliable results from sound. Changing scope during the development process is usual, due to the increasing learning curve over time. With increased knowledge, opportunities and limitations are clearer and it is important to understand that goals set early in the process could be done with bad precision.

Including the DoE results from the sound test into the optimization tool would delimitate the possibilities of heat results and lower the reliability of the software.
It is hard to analyze the effect of the sound created, because of the difficulties to track the sound created from this part of the engine. Sound is energy and will be summarized with the total sound created from the car. That makes it hard to set a specific requirement on noise cancellation, because it is fuzzy to know what the effects will be in the driving environment. But to neglect this function could still result in an increase of noise that would impair the driving situation. Some engine sound is also preferable. Some cars have techniques to increase sound of specific kinds, to overhear high pitched irritating sound. To know if sound created will be good or bad is complicated to know in advance.

Using a requirement list was a good way of implementing the end user's demands and wishes of the end product. Without this information, there is a risk of developing the program after your own knowledge and experience. Since the coding experience differs between the programmers and the end users, this could have resulted in a software not used in the end.

Performing problem decomposition could be both helpful and harmful. It could be used to find the bottlenecks of the entire systems, and solving problems that could affect the end result significantly. But dividing the problems into sub-problems could result in neglecting important information. The simplified model might represent the total system in an inaccurate way. All sub-problems influence each other in some way, both positive and negative. Changing some part of the system could affect parts of the system that has been excluded.

All mathematical models are just approximations of the entire systems. Especially when simplifications have been made. This calculation approach is focusing on a one dimensional isolated model, which should be considered at every usage. The benefit of creating simplified models is their availability in an earlier stage of the process. The final tests could not be executed until the entire engine is completed, when it is too late to implement changes. Using this model and setting a worst case scenario in the input values, an approximation could be made in an early stage. Using the same heat input, different kind of material combinations could be analyzed even though the heat input does not represent the actual case in the end, delta values of the heat transfer could be made between concepts find the best alternative early in the

process. The isolated model will also have zero percent energy loss from the one dimensional model, which will create results higher than reality. With this in mind, the reality would probably show lower temperatures and the results from the calculations would then create a safety factor in its output and present the worst case scenario. The calculation will create temperature deviation with higher change velocity, which would result in sharp peaks in its extreme points. The tests made, did not show if the measurements showed accurate results or if the test rig and its environment influenced on the results. Therefore, the temperature variation could vary in reality. On other hand, the calculation represents the physical tests in an accurate way, and the tests are used today in that manner. With that in mind, the calculation could be used with the same consideration as earlier test method.

6. CONCLUSIONS AND RECOMMENDATIONS

The case study showed that reduction of time spent on heat transfer is possible, while maintaining high accuracy of the output results.

The 202020 plan at VCG encourage reduction of the time needed for developing new cars. This project minimized the time for heat test procedure and is a great example on working with a 202020 point of view. Working with finding bottlenecks in processes at other departments should be done in order to improve the overall development process.

The problem decomposition showed other problems that influence the product, such as vibration and fatigue. This project failed to calculate the sound in a proper way. These areas could with the same effort as heat, improve their procedures. This software could be a base for developing a multi-optimizer software, focusing on more than heat analysis.

According to the objective in *Chapter 1.4*, the goal was to shorten the development time in order to get the EE group in the 202020 path. This project has resulted in a reduction of 97 % of the heat test with an accuracy of 98%. With no physical material needed for the tests, it could be conducted in an earlier stage of the development process.

Our recommendation to VCG is to continue the work of finding ways of replacing physical tests with calculations to predict the behavior in a real car. Physical tests are time consuming and expensive. Using simulation methods instead is the future of automotive development. This needs to be implemented in order compete with competitors.

BIBLOGRAPHY

[1] Volvo Car Group. (2014) *Heritage*, [Online], Available: http://www.volvocars.com/intl/about/our-company/heritage [16 June 2015].

[2] Schwartz, E. (2015) *Transformation 202020*, [Online], Available: http://m.efqm.org/25Volvo.pdf [16 June 2015].

[3] Wheelwright, Steve C. and Clark, Kim B. (1992) *Revolutionizing Product Development*, Free Press.

[4] Wheelwright, Steve C. and Clark, Kim B. (1992) *Revolutionizing Product Development*, Free Press.

[5] Schwartz, E. (2015) *Transformation 202020*, [Online], Available: http://m.efqm.org/25Volvo.pdf [16 June 2015].

[6] Olson, Harry F. (1967) *Music, Physics and Engineering*, Dover Publications.

[7] Moylan, W. (2006) *Understanding and crafting the mix: the art of recording*, Focal Press.

[8] http://www.thelancet.com/journals/lancet/article/PIIS0140-6736(13)61613-X/abstract?cc=y=

[9] Mathias Basner, Dr. M., Babisch, W., Davis, A., Brink, M., Clark, C., Janssen, S. and Stansfeld, S. (2013) '*Auditory and non-auditory effects of noise on health'* [Electronic], vol. 383, no. 9925, pp. 1325-1332, Available: http://www.thelancet.com/journals/lancet/article/PIIS0140-6736(13)61613-X, [16 June 2015].

[10] Incropera, F. and Dewitt, D. (1985) *Introduction to heat transfer*, New York: John Wiley and Sons Inc..

[11] Wheelwright, Steve C. and Clark, Kim B. (1992) *Revolutionizing Product Development*, Free Press.

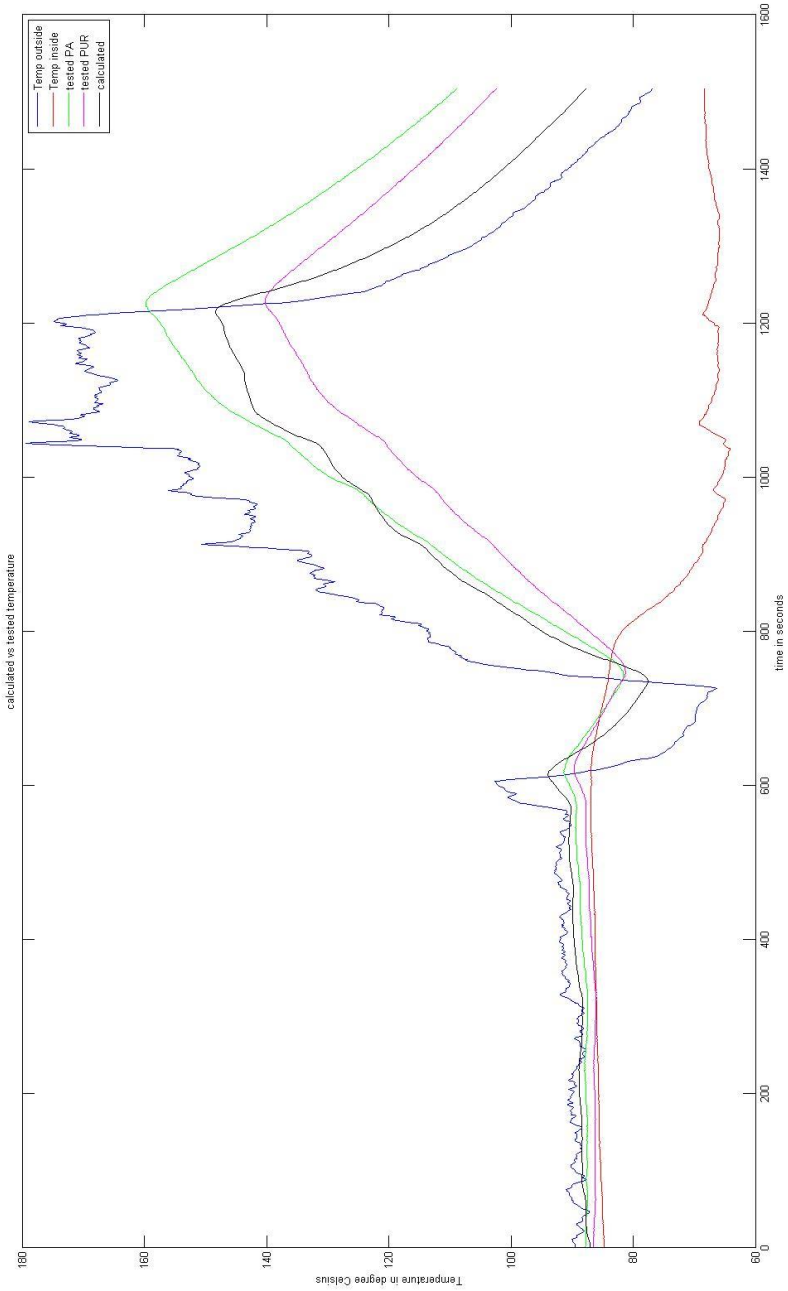
[12] Papalambros, P.Y. and Wilde, D.J. (2000) *Principles of Optimal Design, 2nd Edition*, Cambridge University Press.

[13] Versteeg, H.K. and Malalasekera, W. (2007) *An Introduction to Computational Fluid Dynamics, 2nd Edition*, Harlow: Pearson Education Limited.

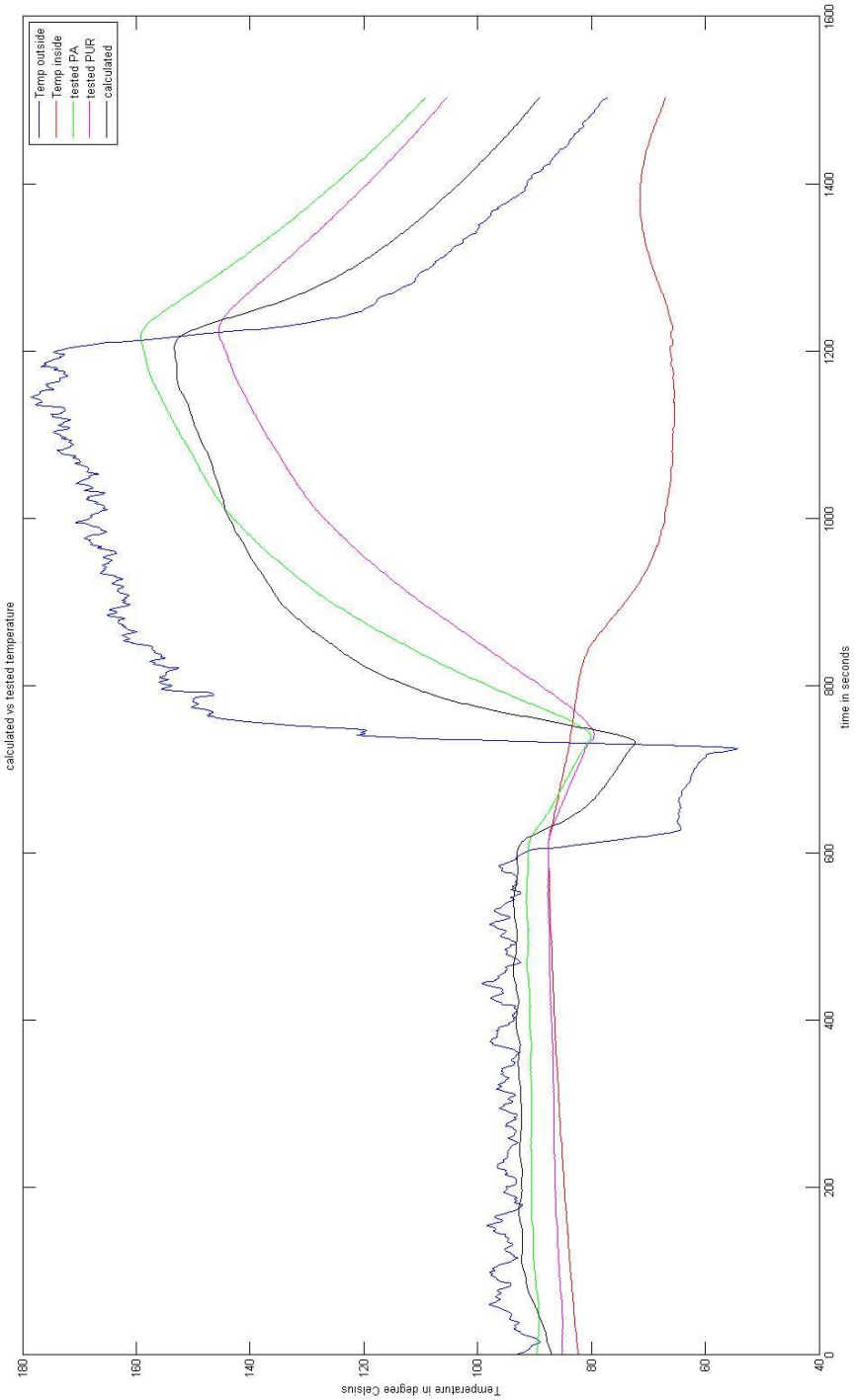
[14] Incropera, F. and Dewitt, D. (1985) *Introduction to heat transfer*, New York: John Wiley and Sons Inc..

APPENDIX A – HEAT CURVES

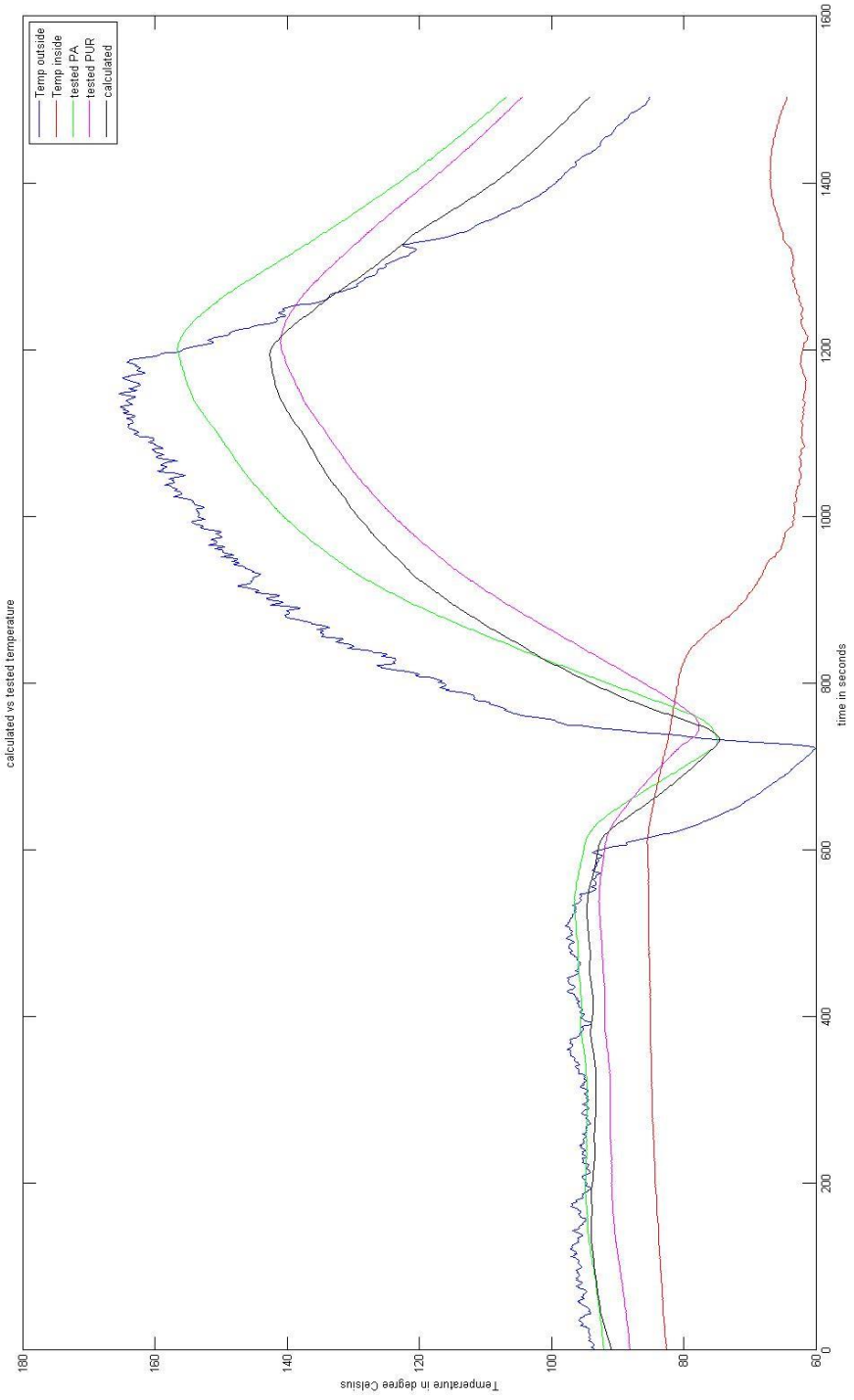
Following figures show the measured data from the validation tests during the project.



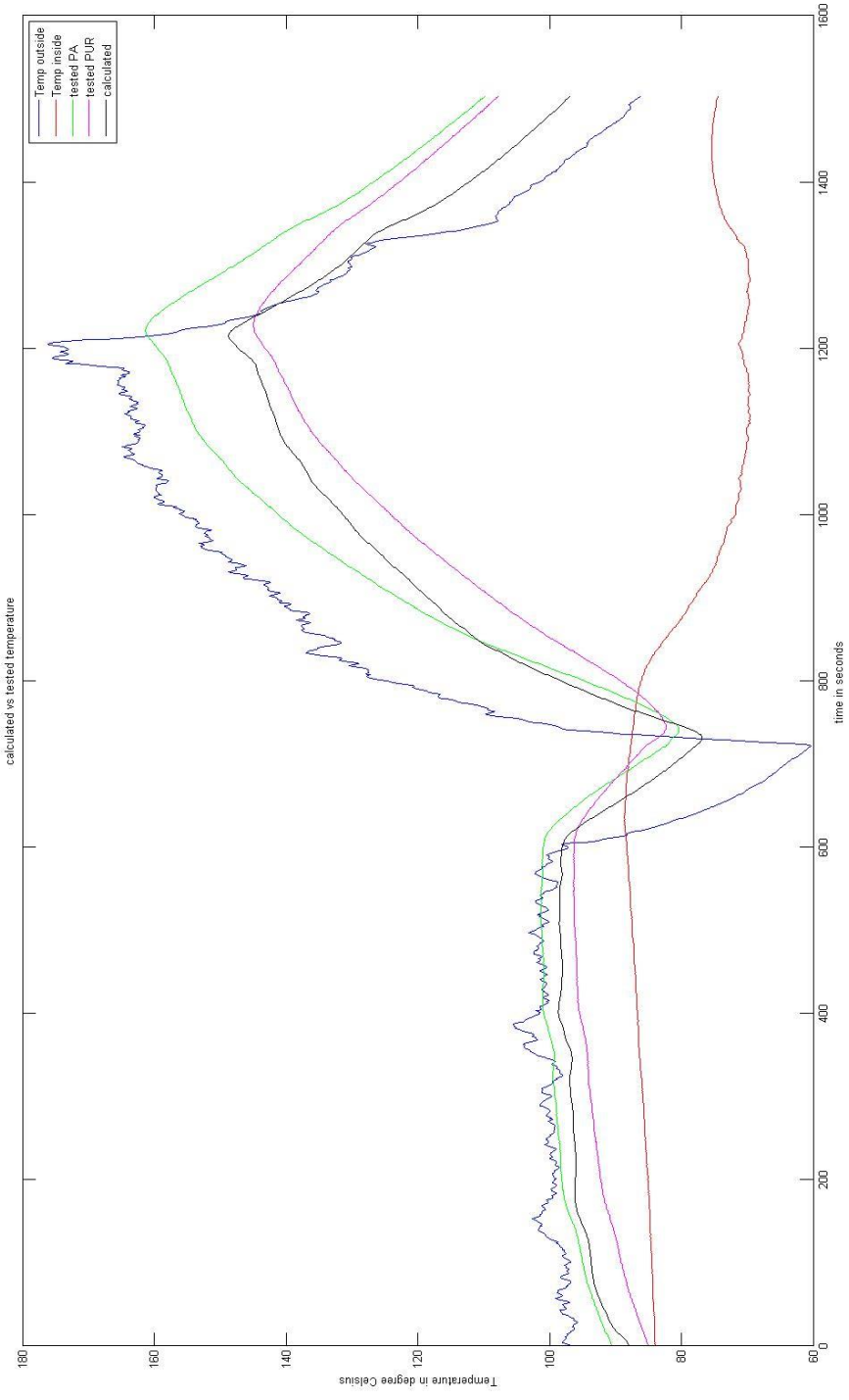
***Plot 1**: Graph from Test 1*



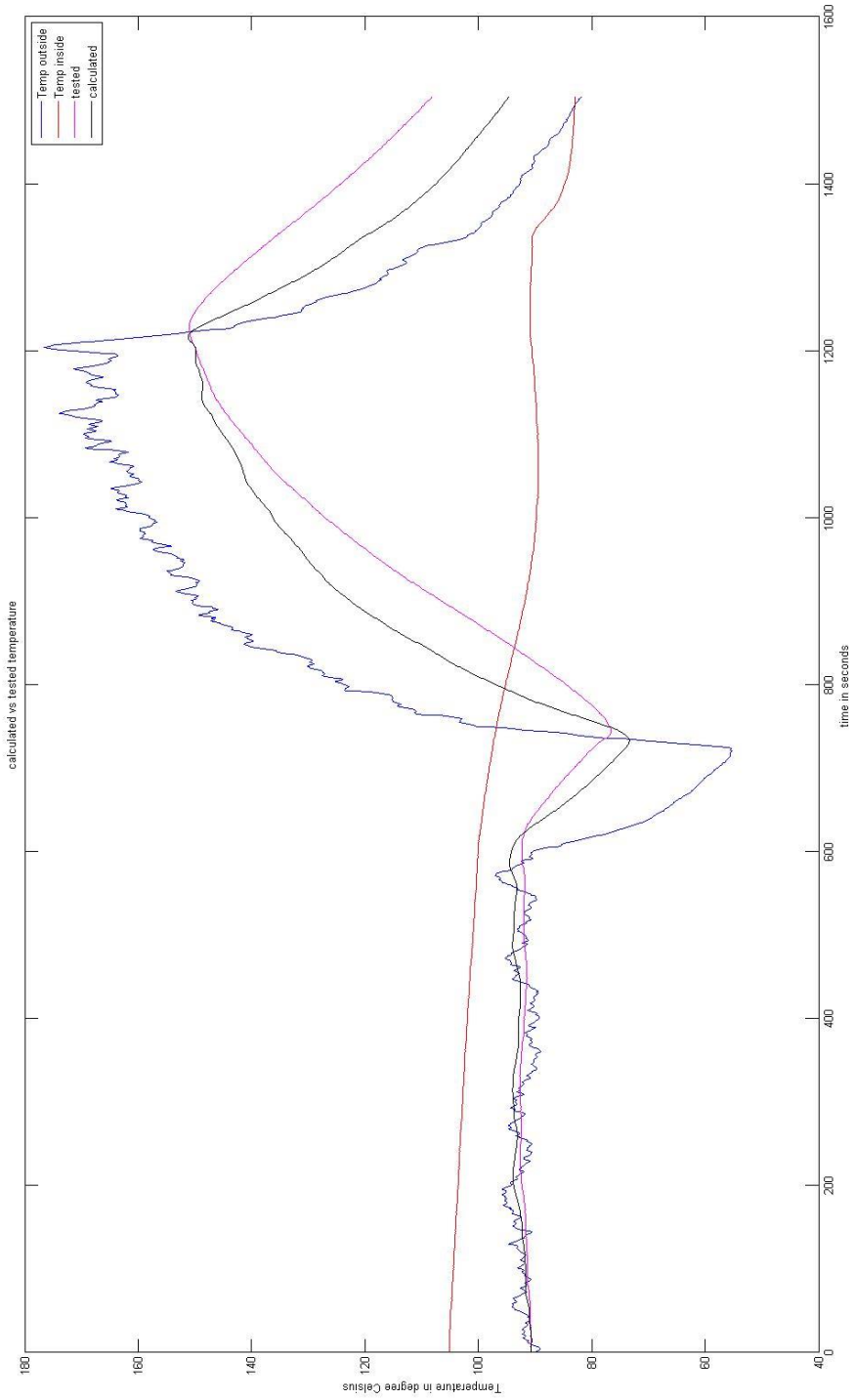
***Plot 2**: Graph from Test 2*



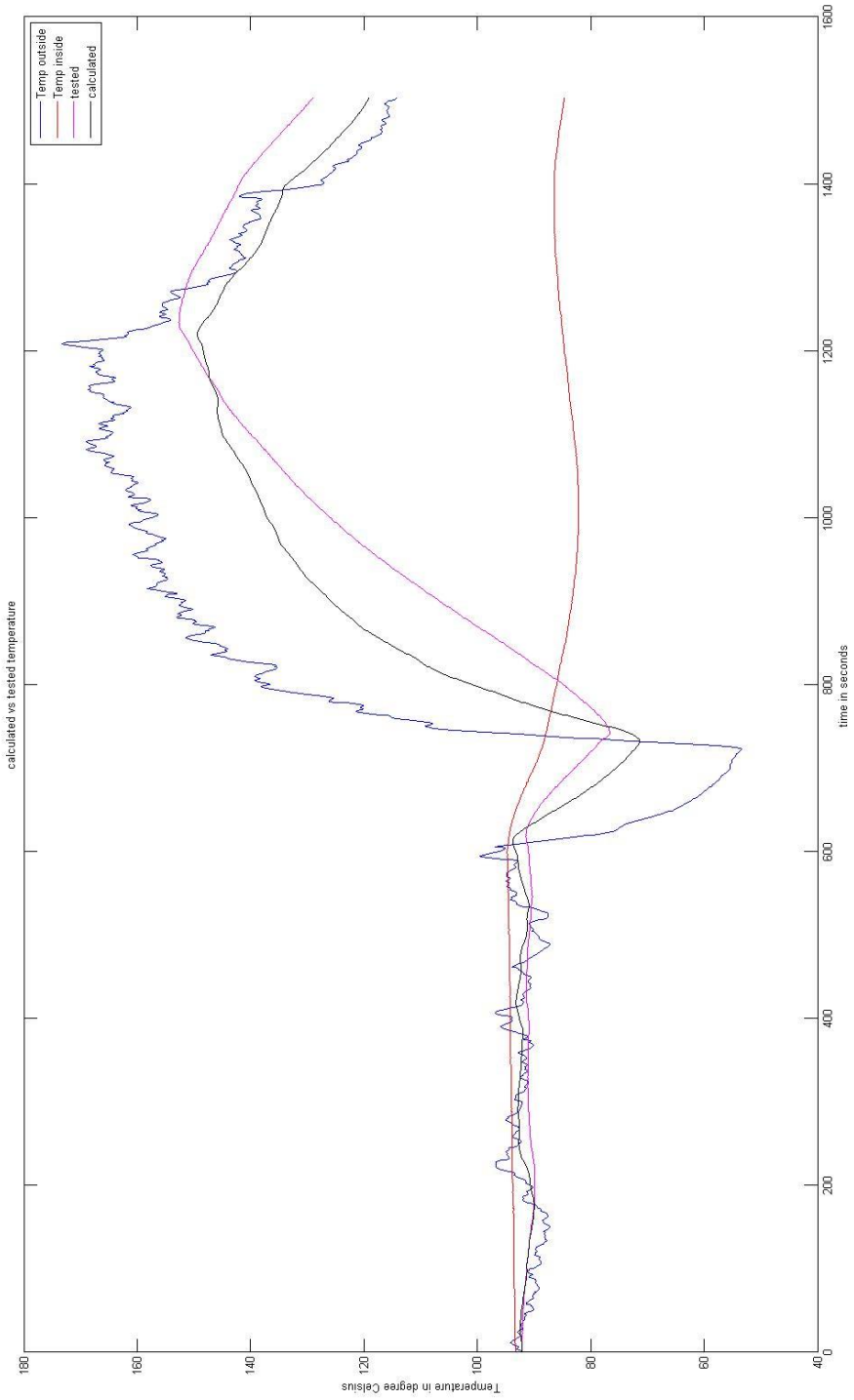
***Plot 3**: Graph from Test 3*



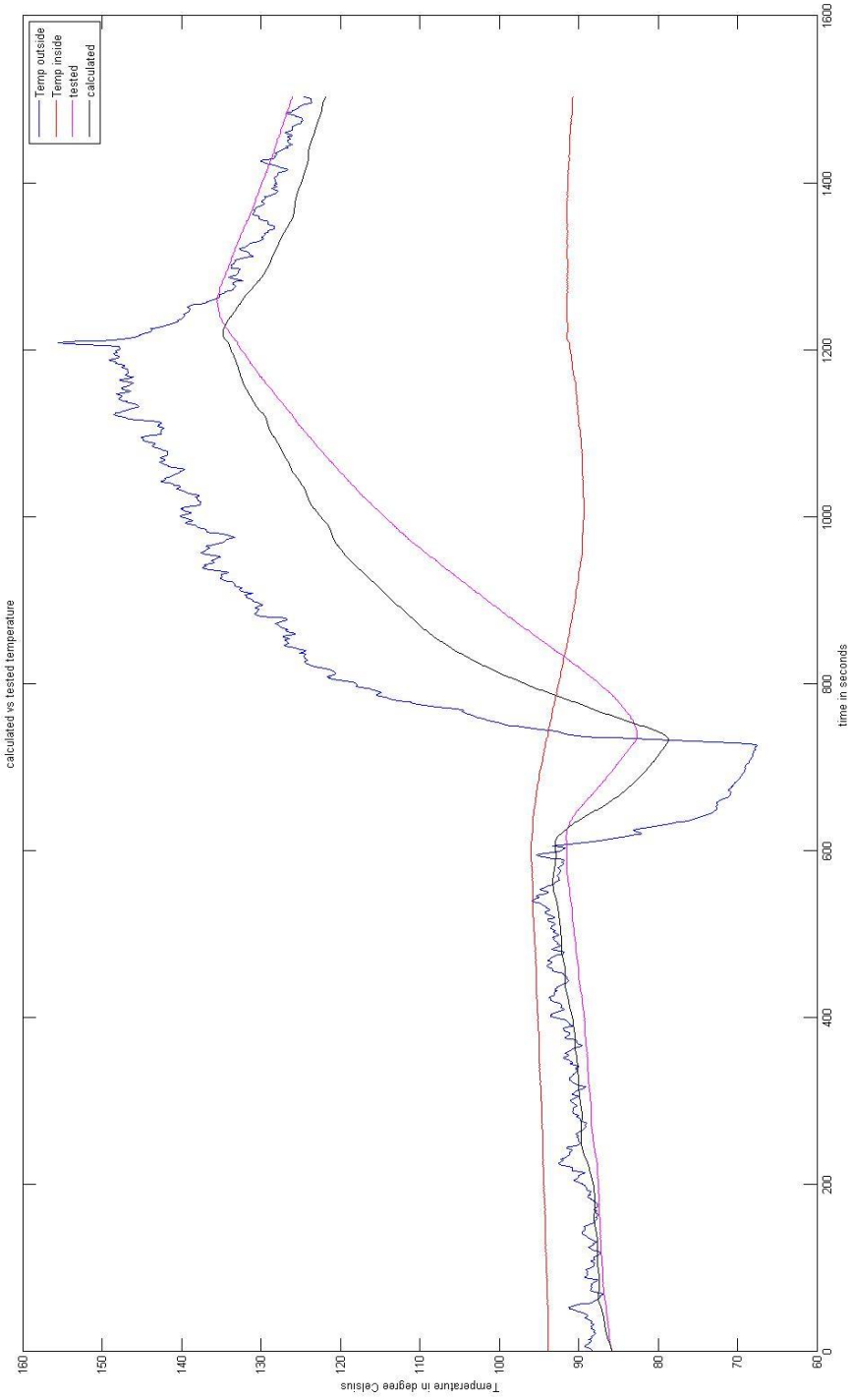
***Plot 4**: Graph from Test 4*



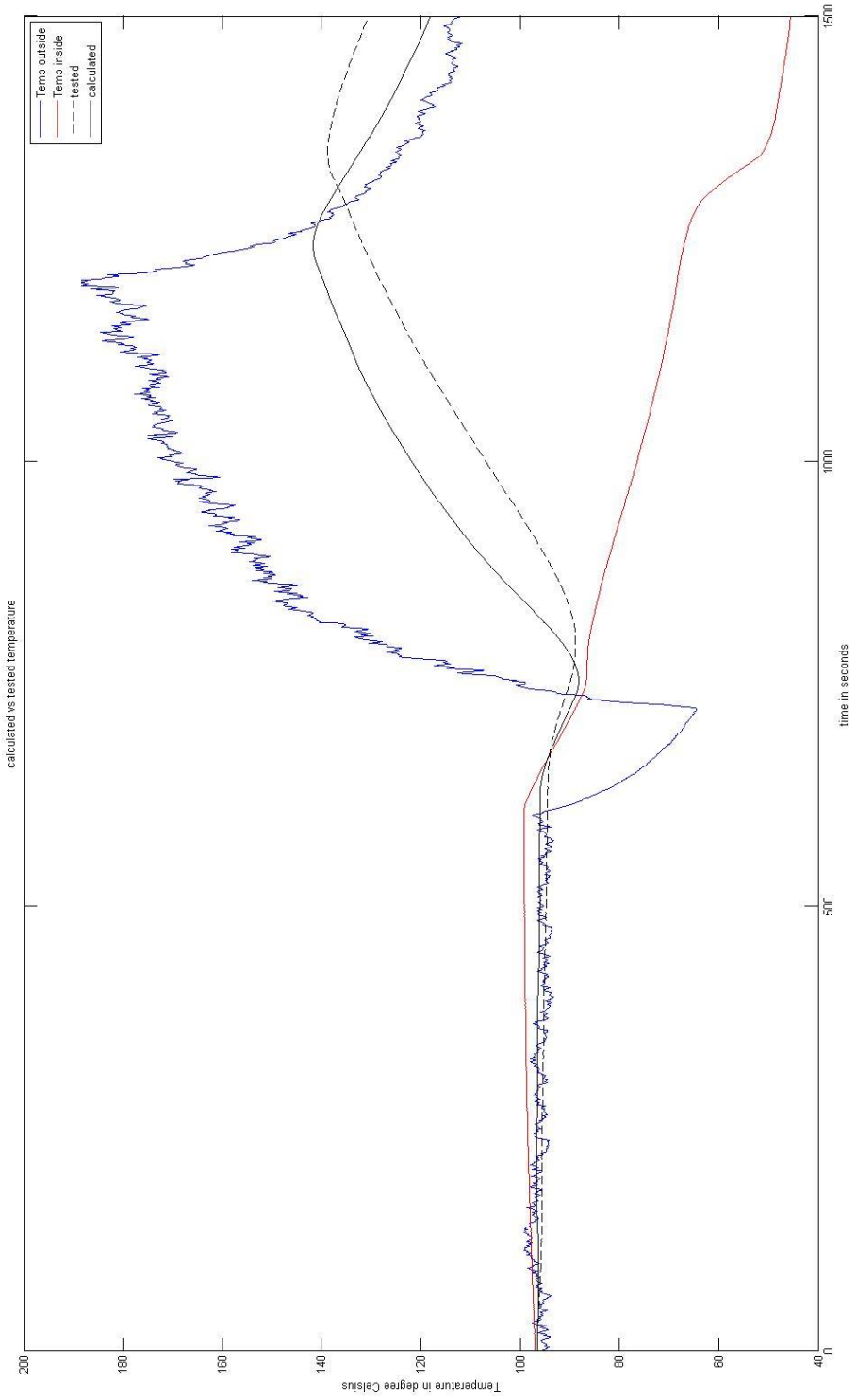
***Plot 5**: Graph from Test 5*



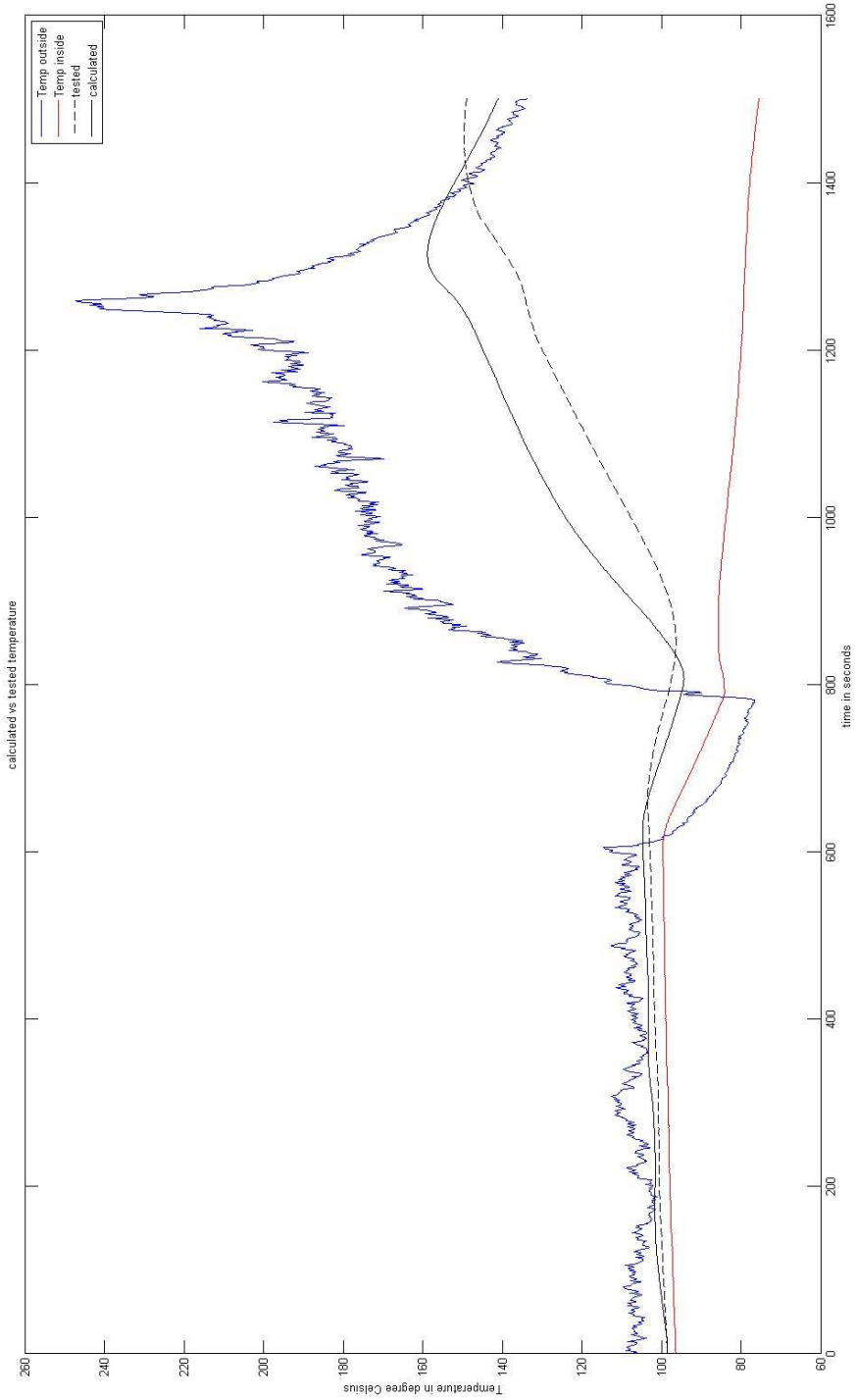
***Plot 6**: Graph from Test 6*



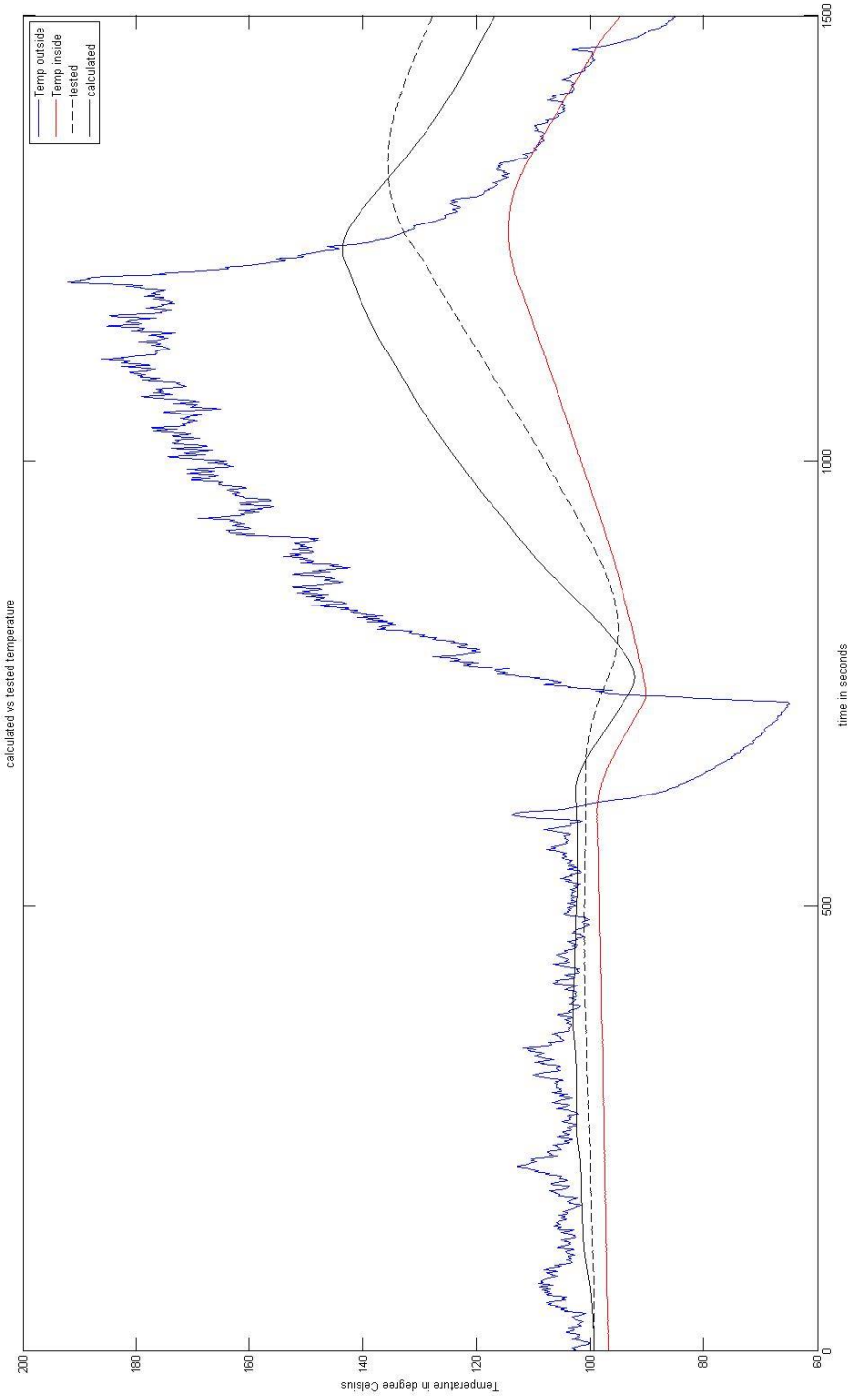
***Plot 7**: Graph from Test 7*



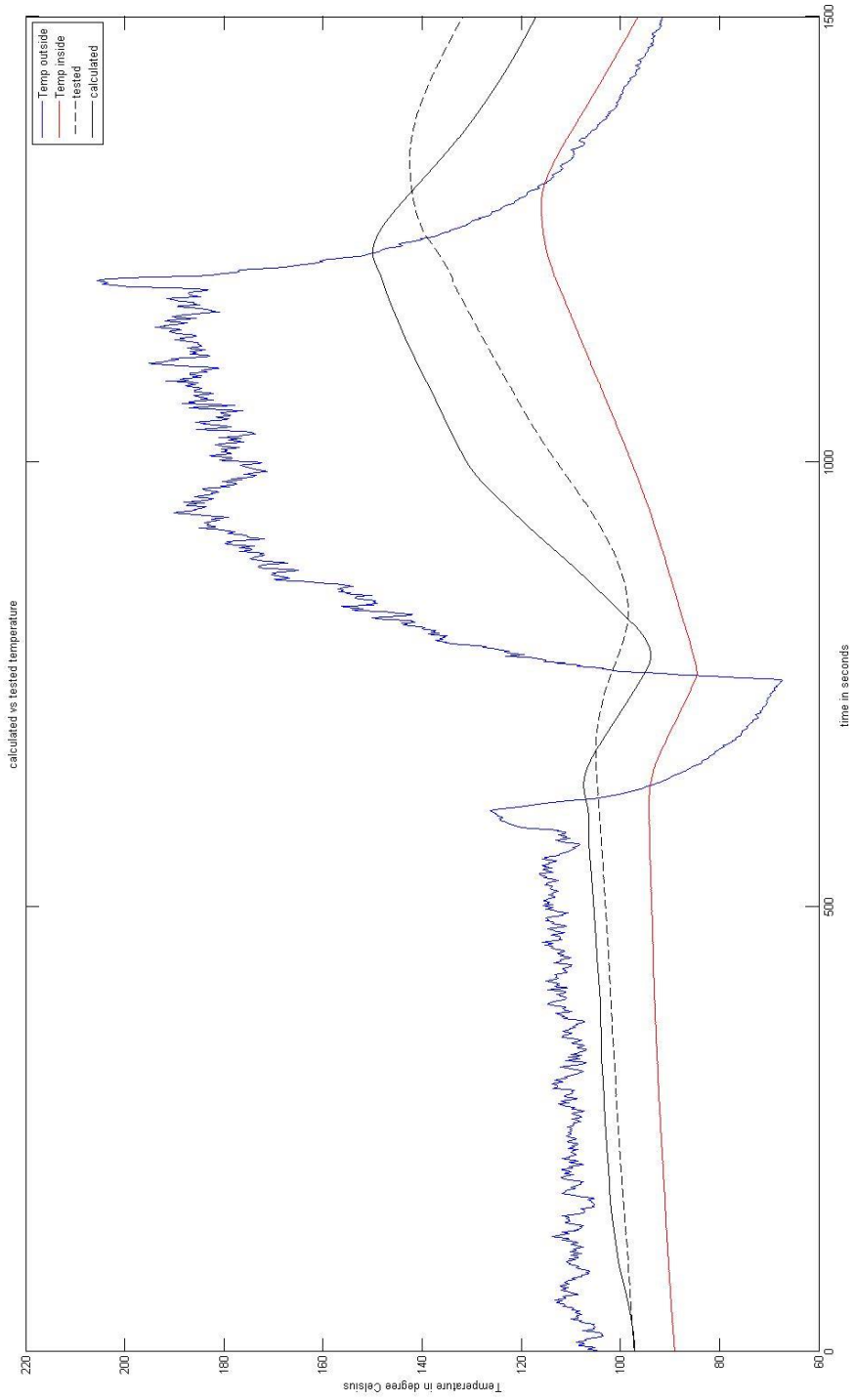
***Plot 8**: Graph from Test 13*



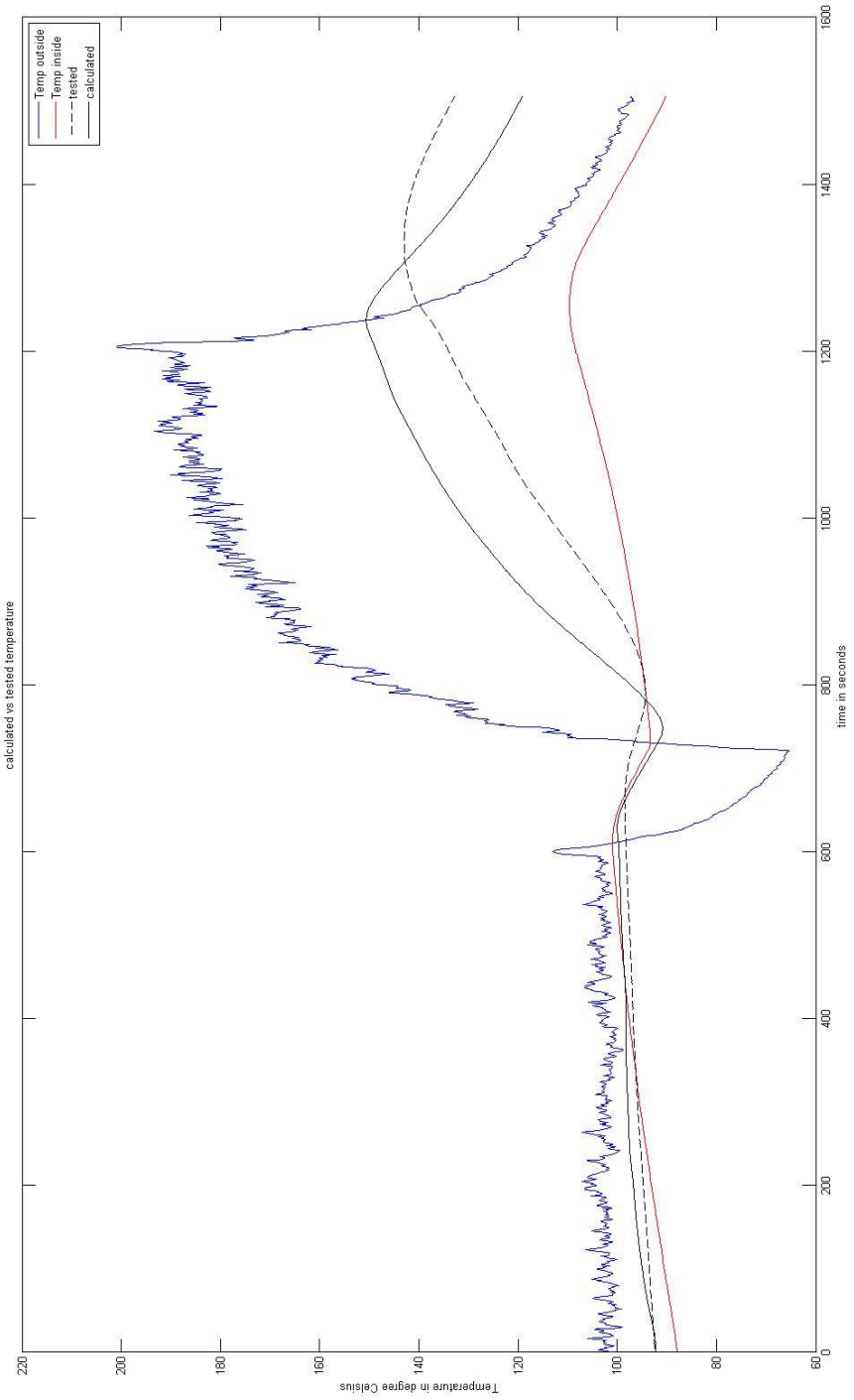
***Plot 9**: Graph from Test 14*



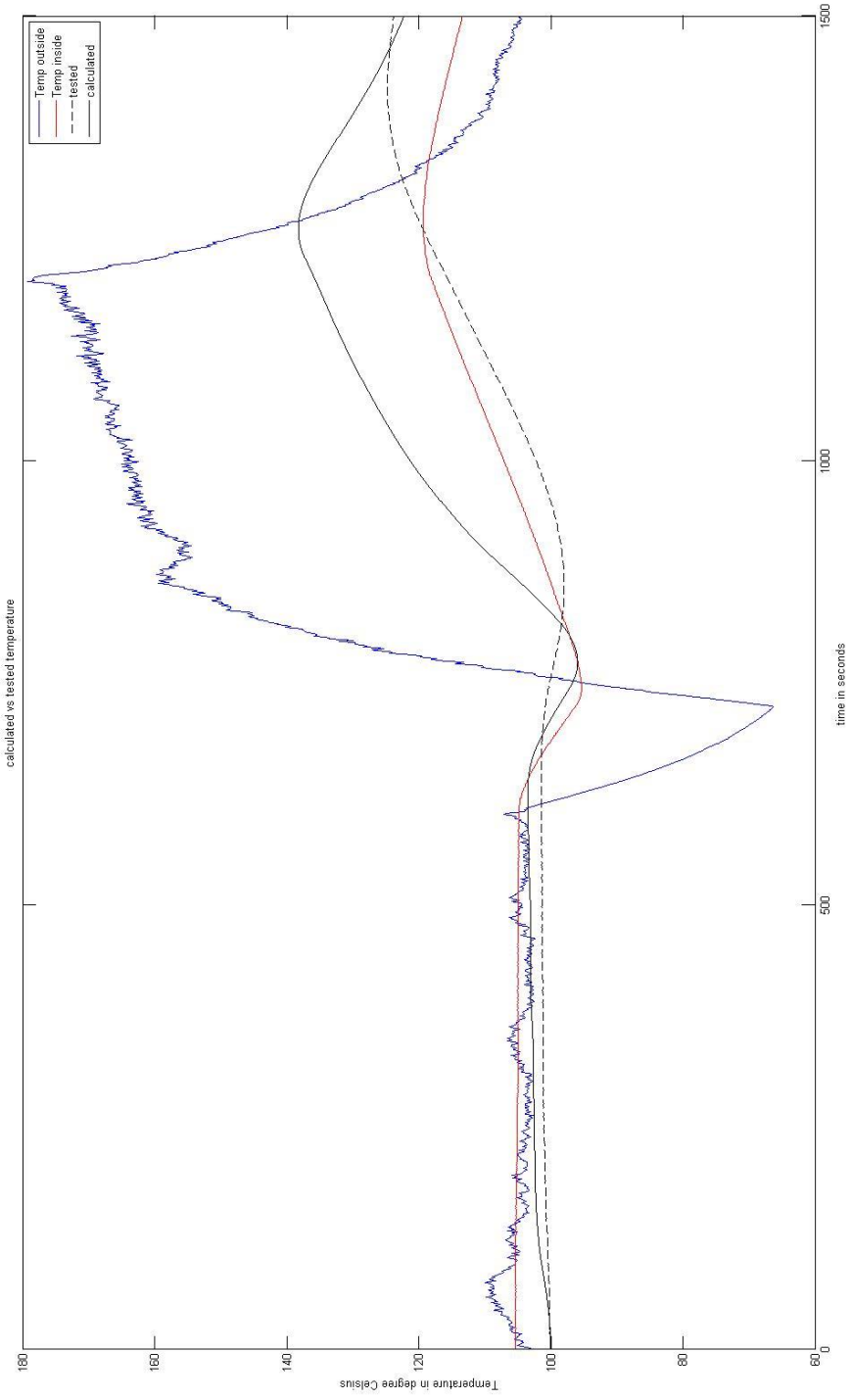
***Plot 10**: Graph from Test 15*



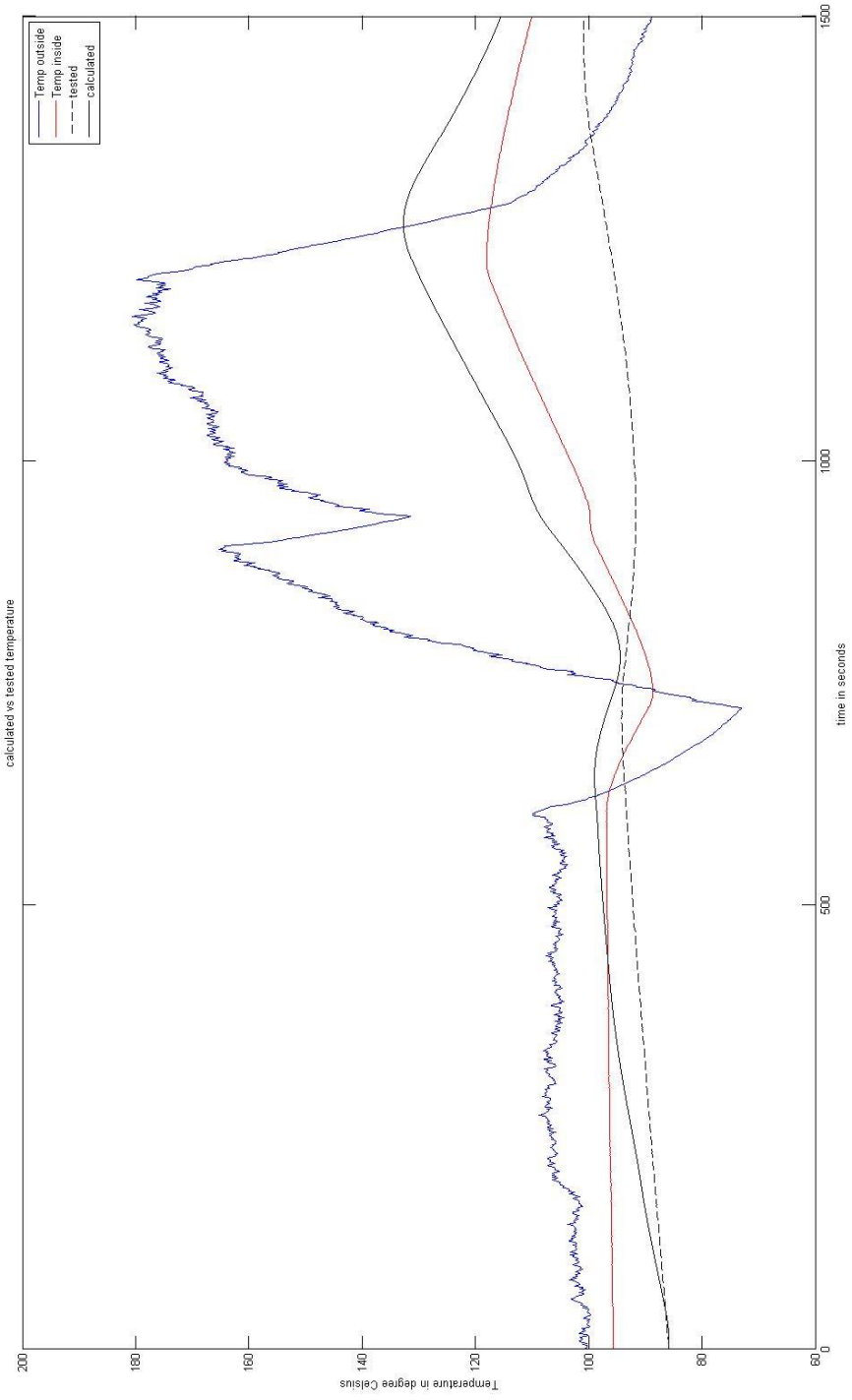
***Plot 11**: Graph from Test 16*



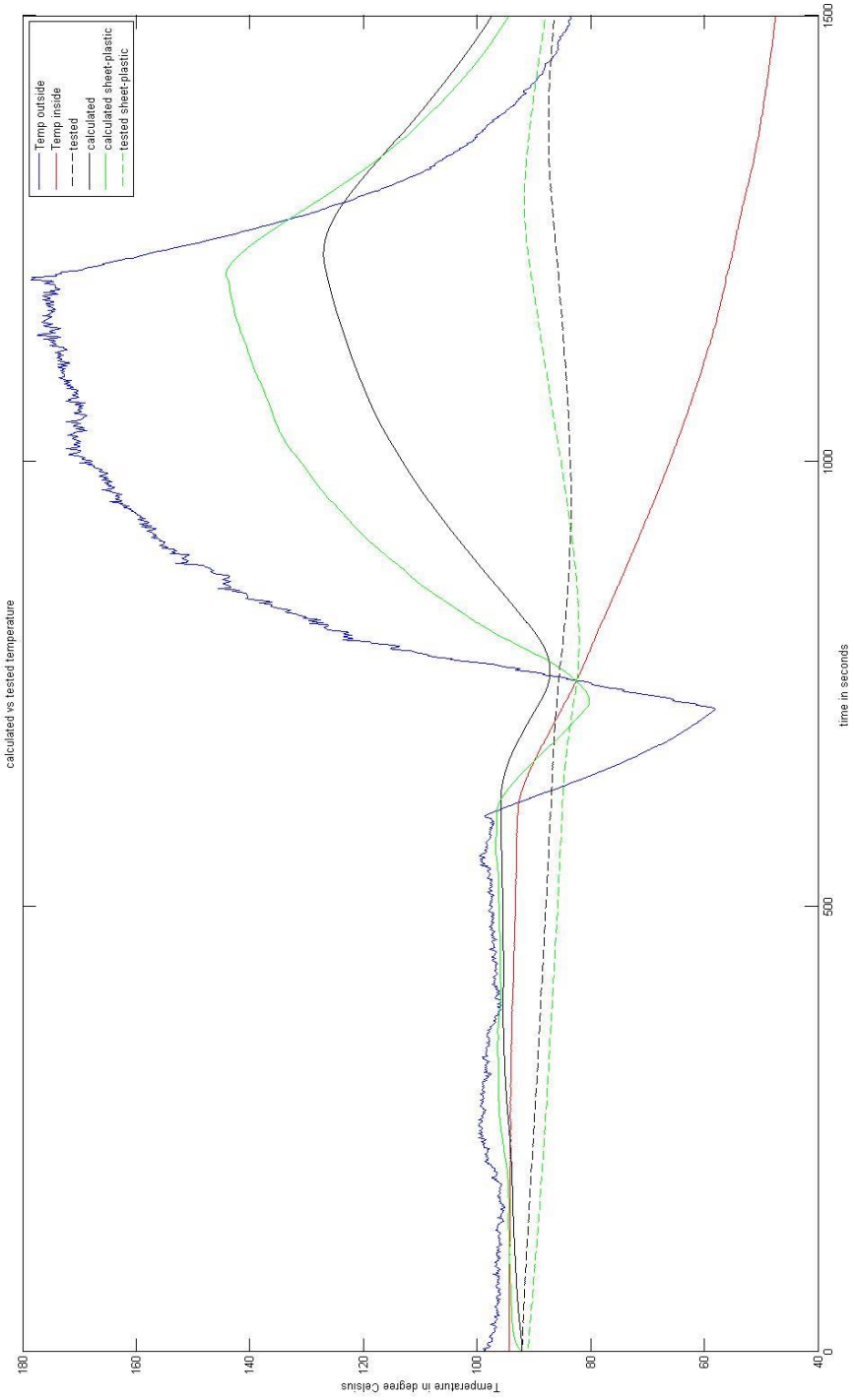
***Plot 12**: Graph from Test 17*



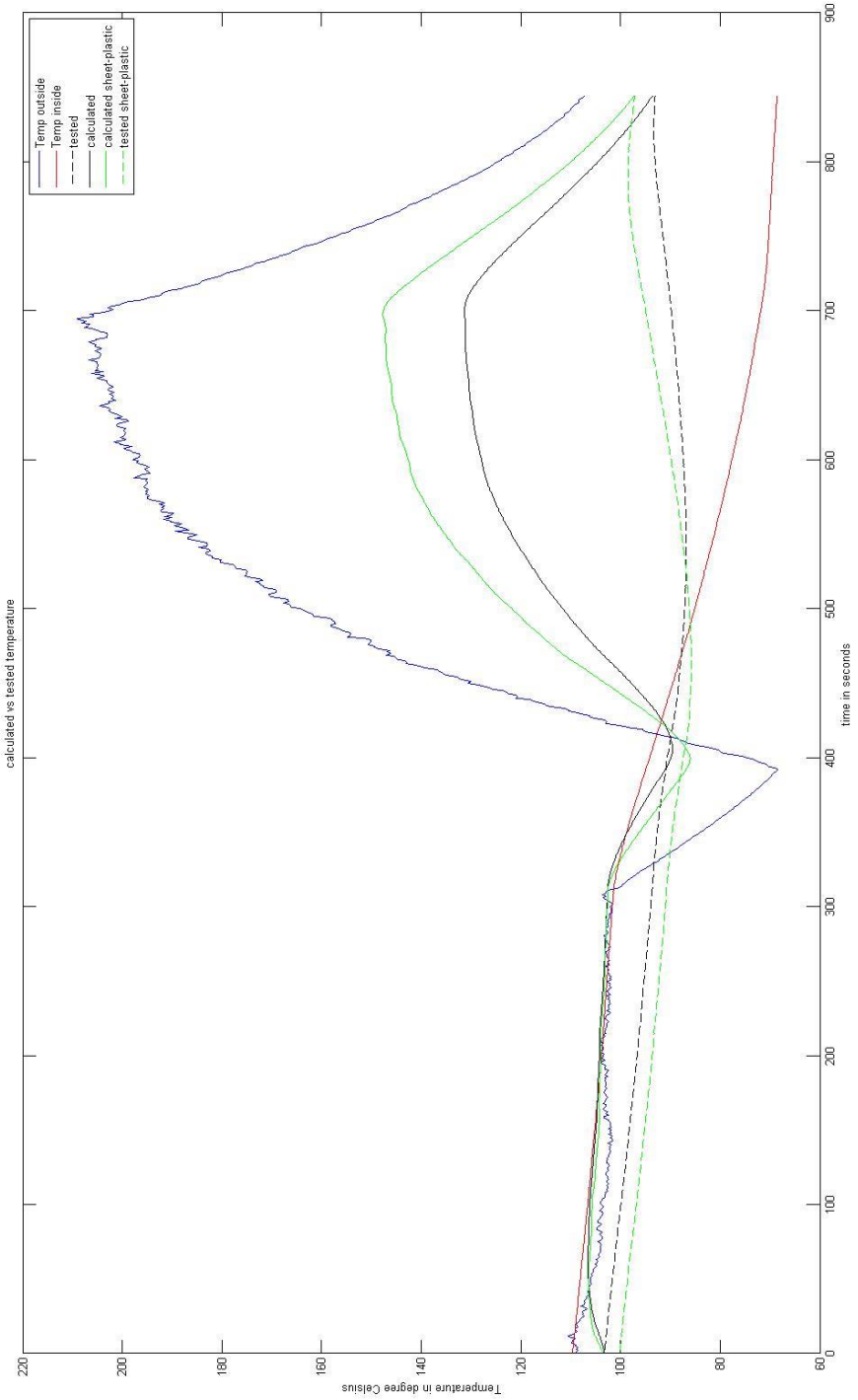
***Plot 13**: Graph from Test 18*



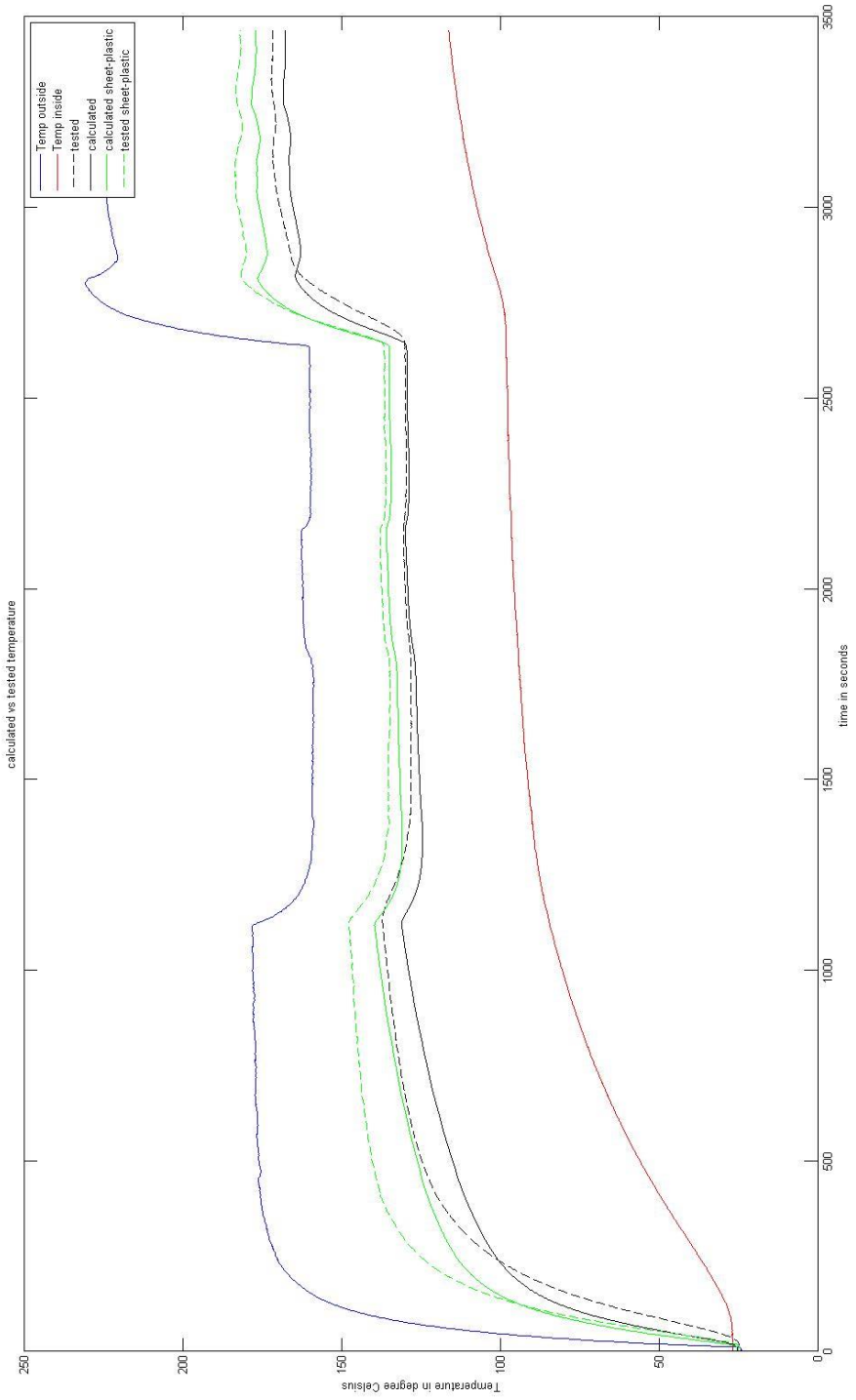
***Plot 14**: Graph from Test 19*



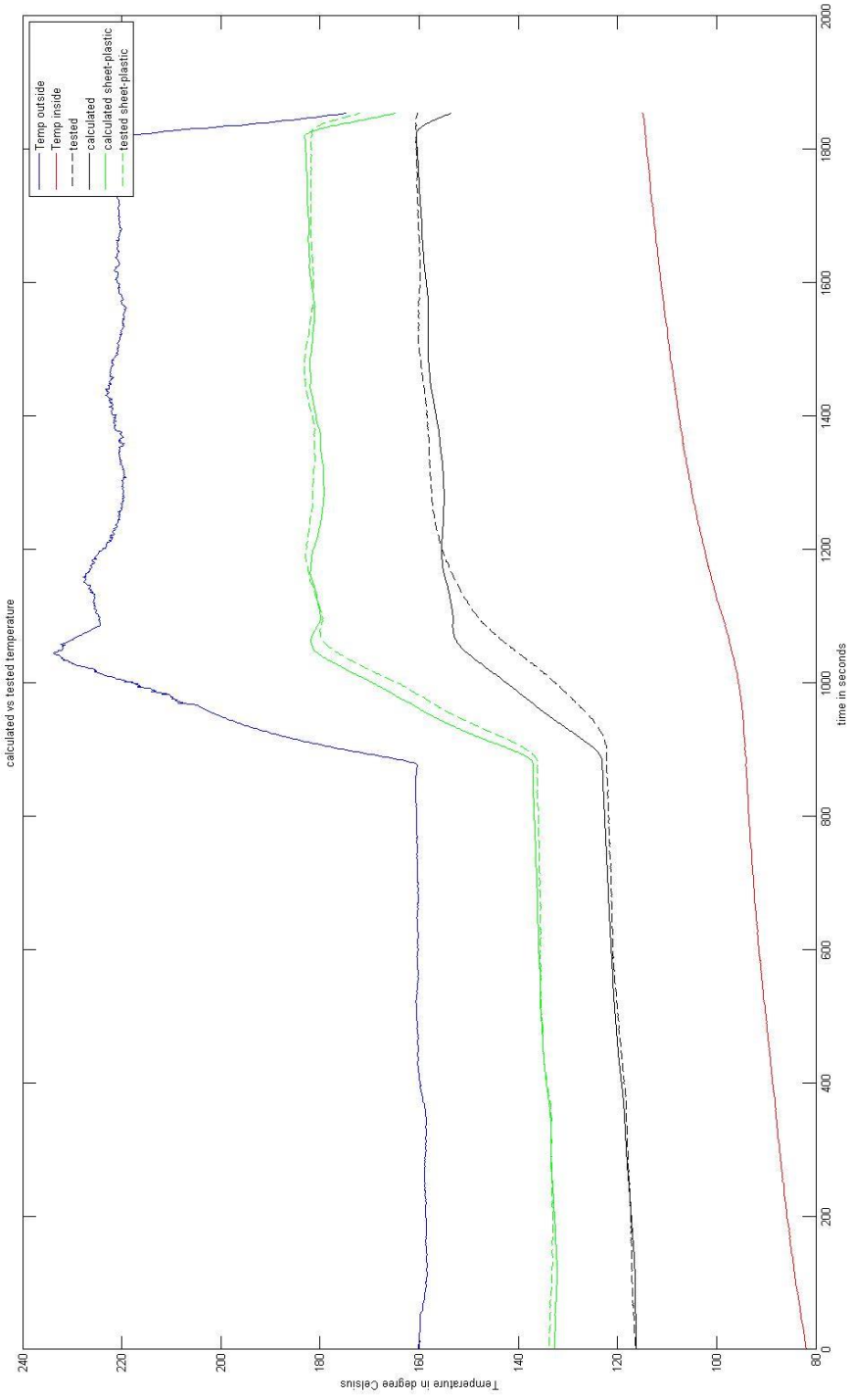
***Plot 15**: Graph from Test 20*



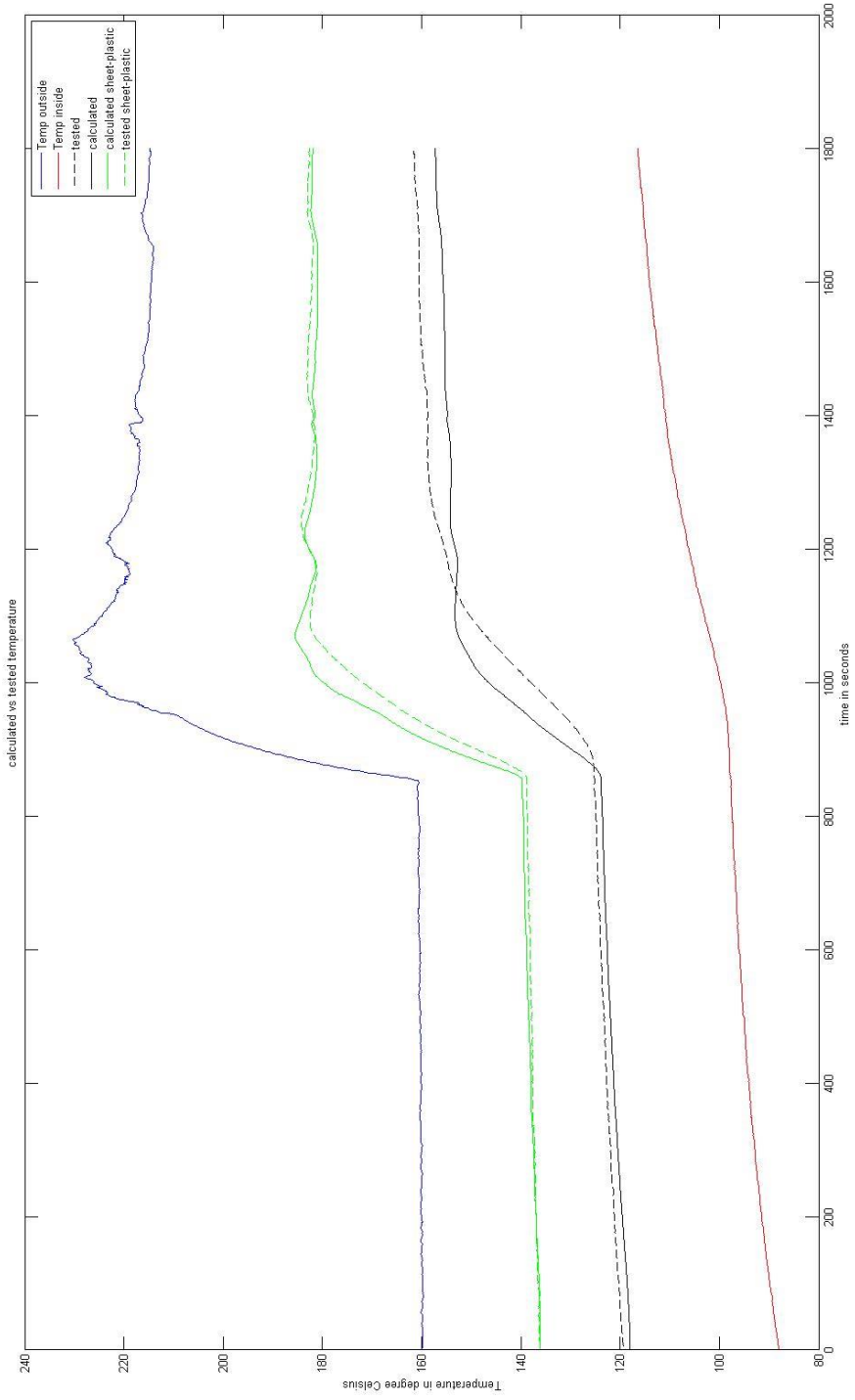
***Plot 16**: Graph from Test 21*



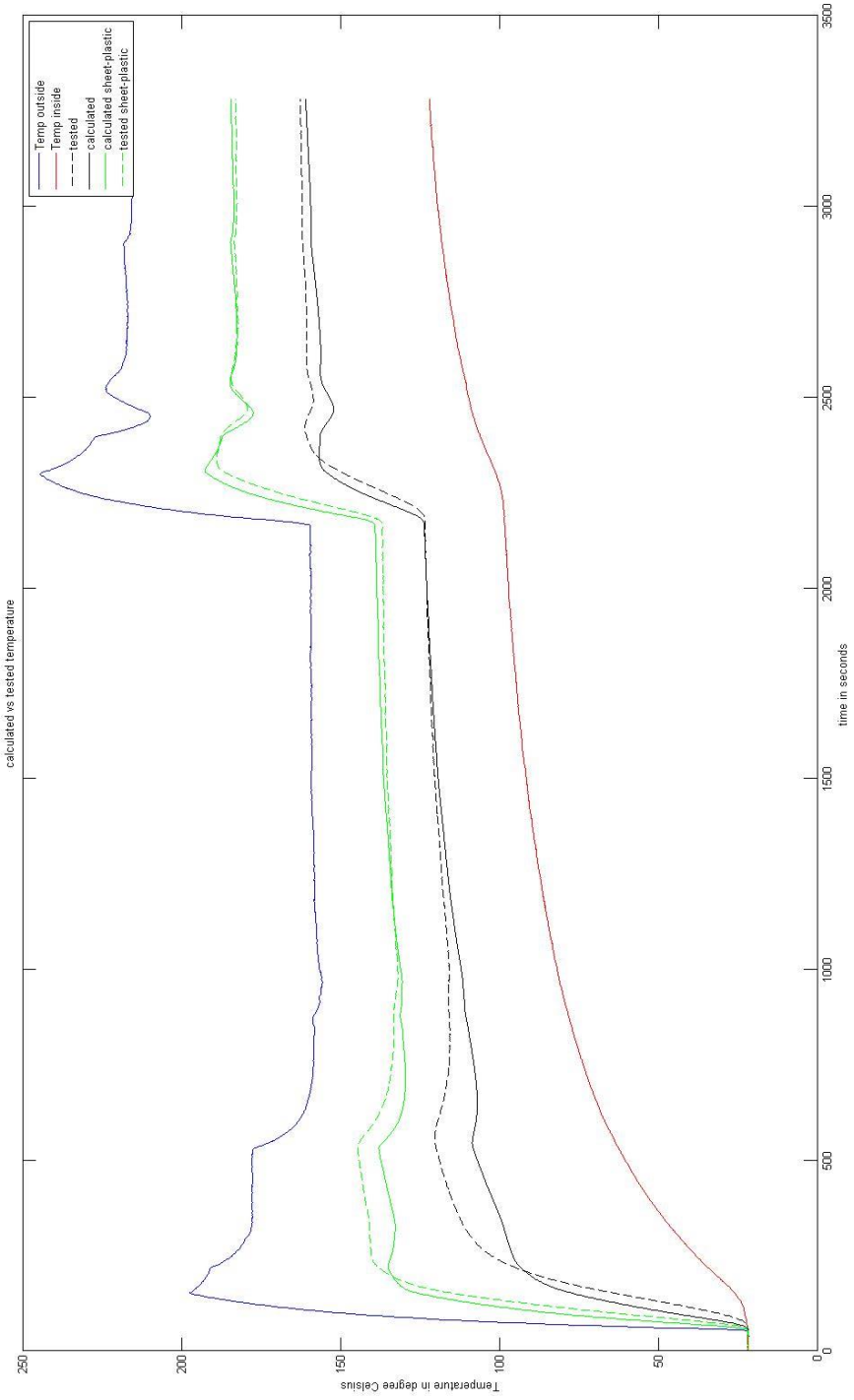
***Plot 17**: Graph from Case Study Test 1*



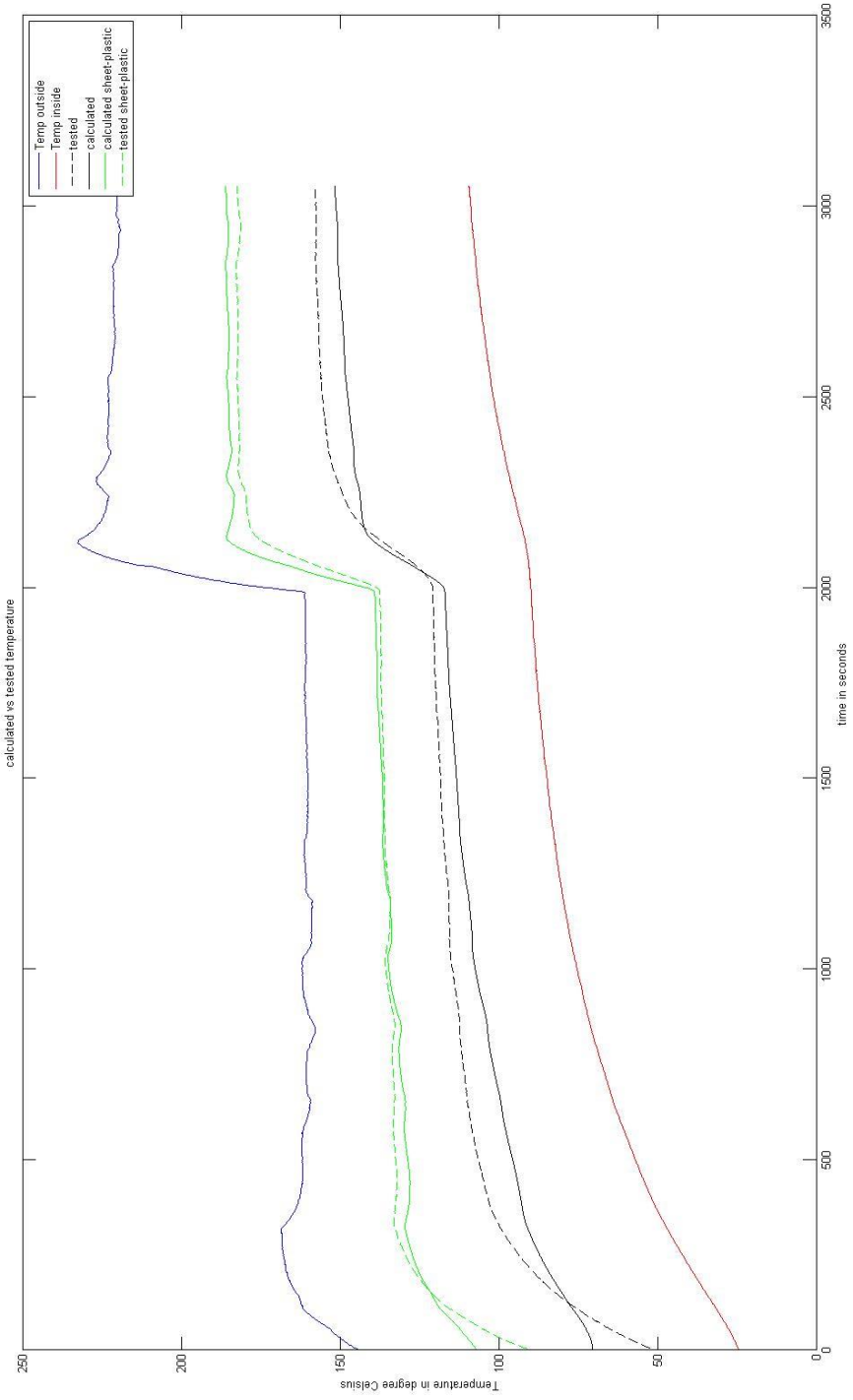
***Plot 18**: Graph from Case Study Test 2*



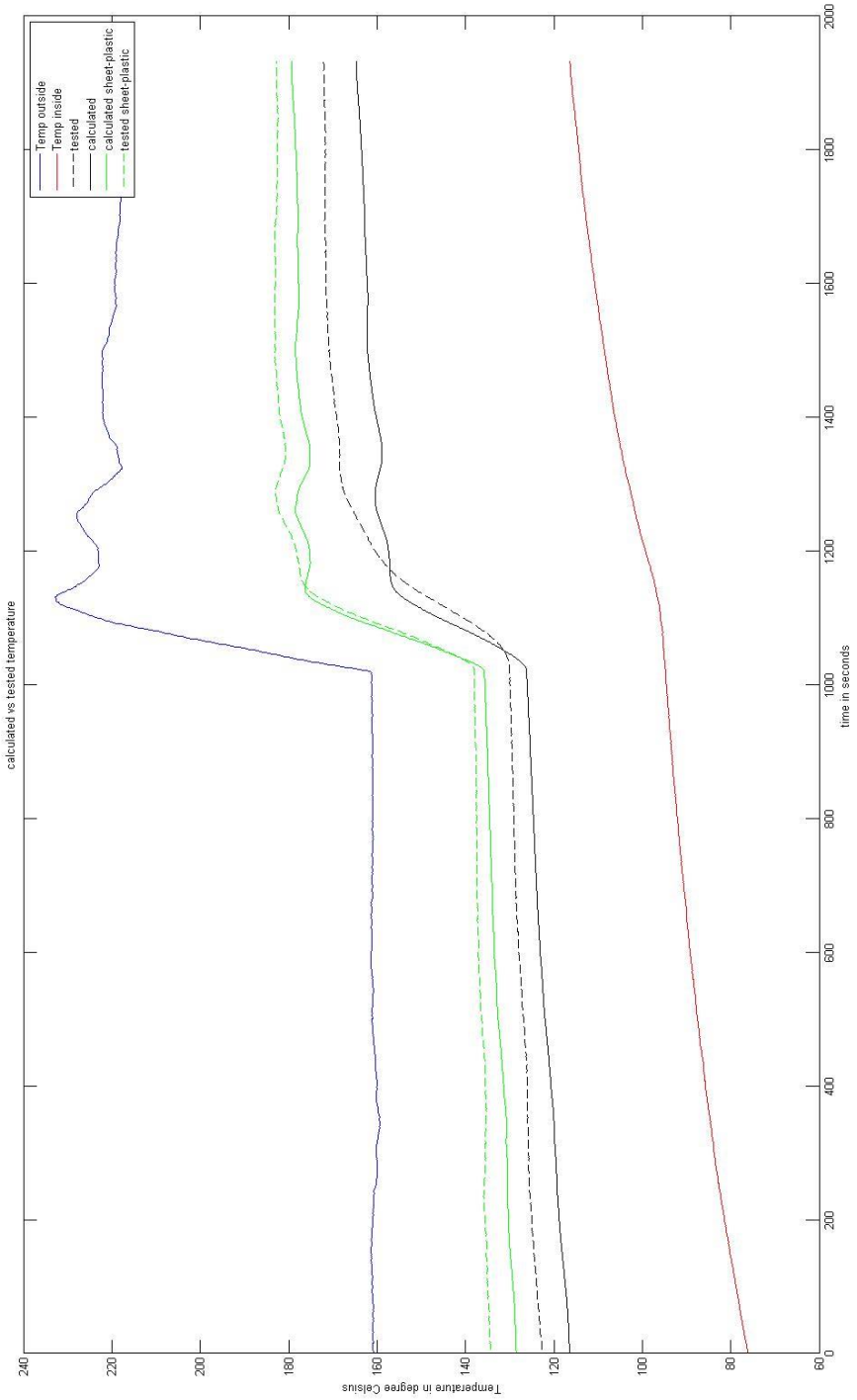
***Plot 19**: Graph from Case Study Test 3*



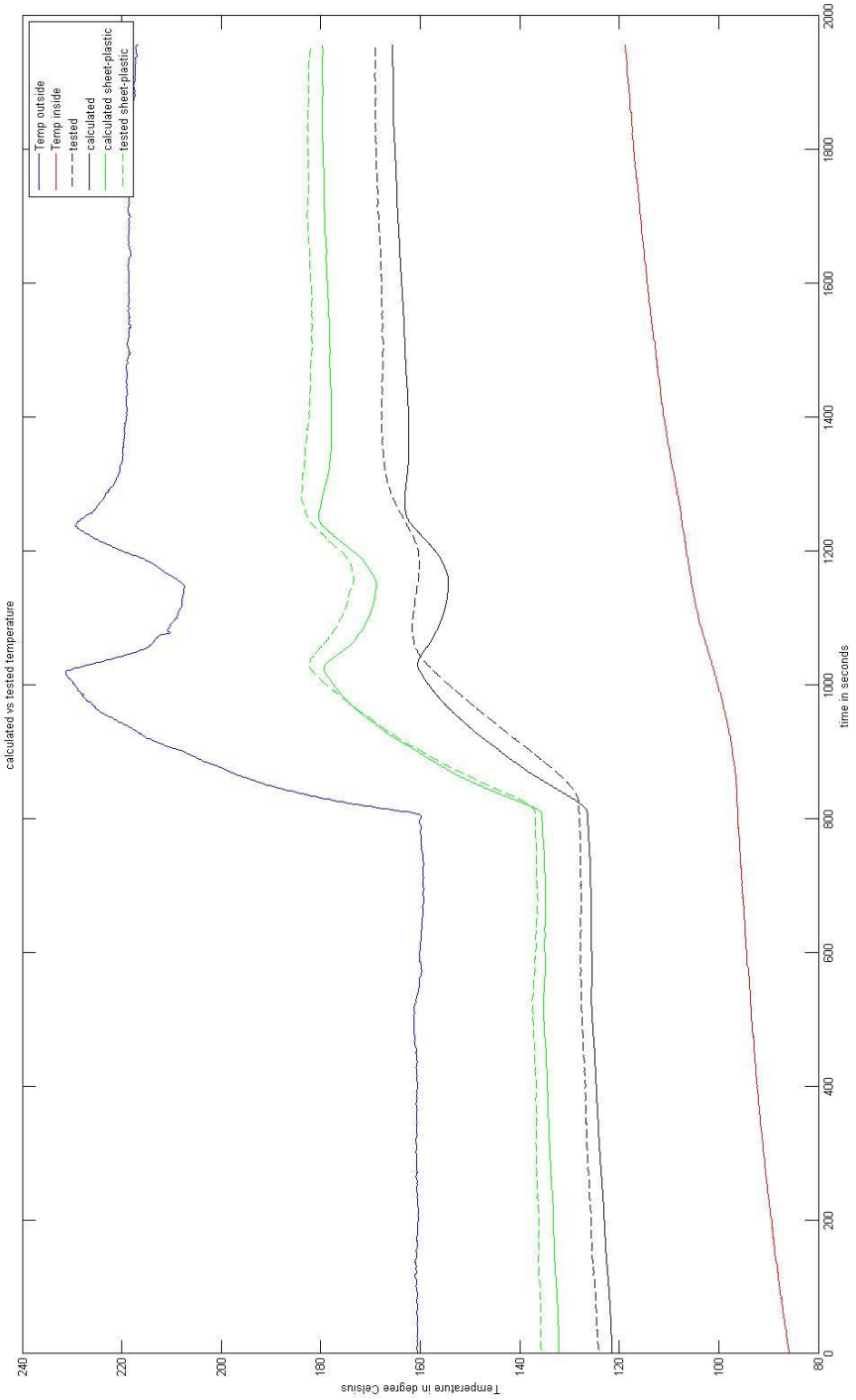
***Plot 20**: Graph from Case Study Test 4*



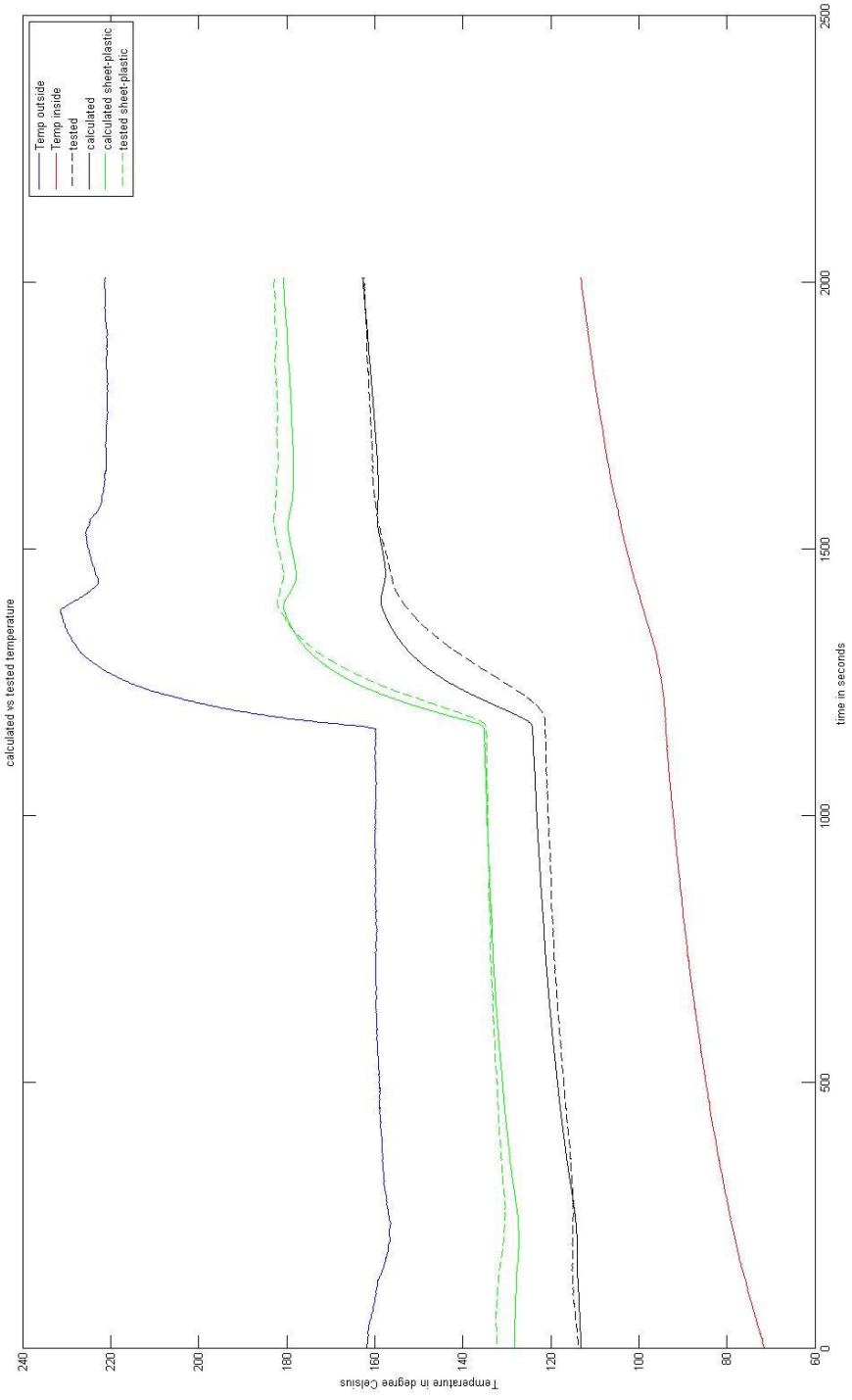
***Plot 21**: Graph from Case Study Test 5*



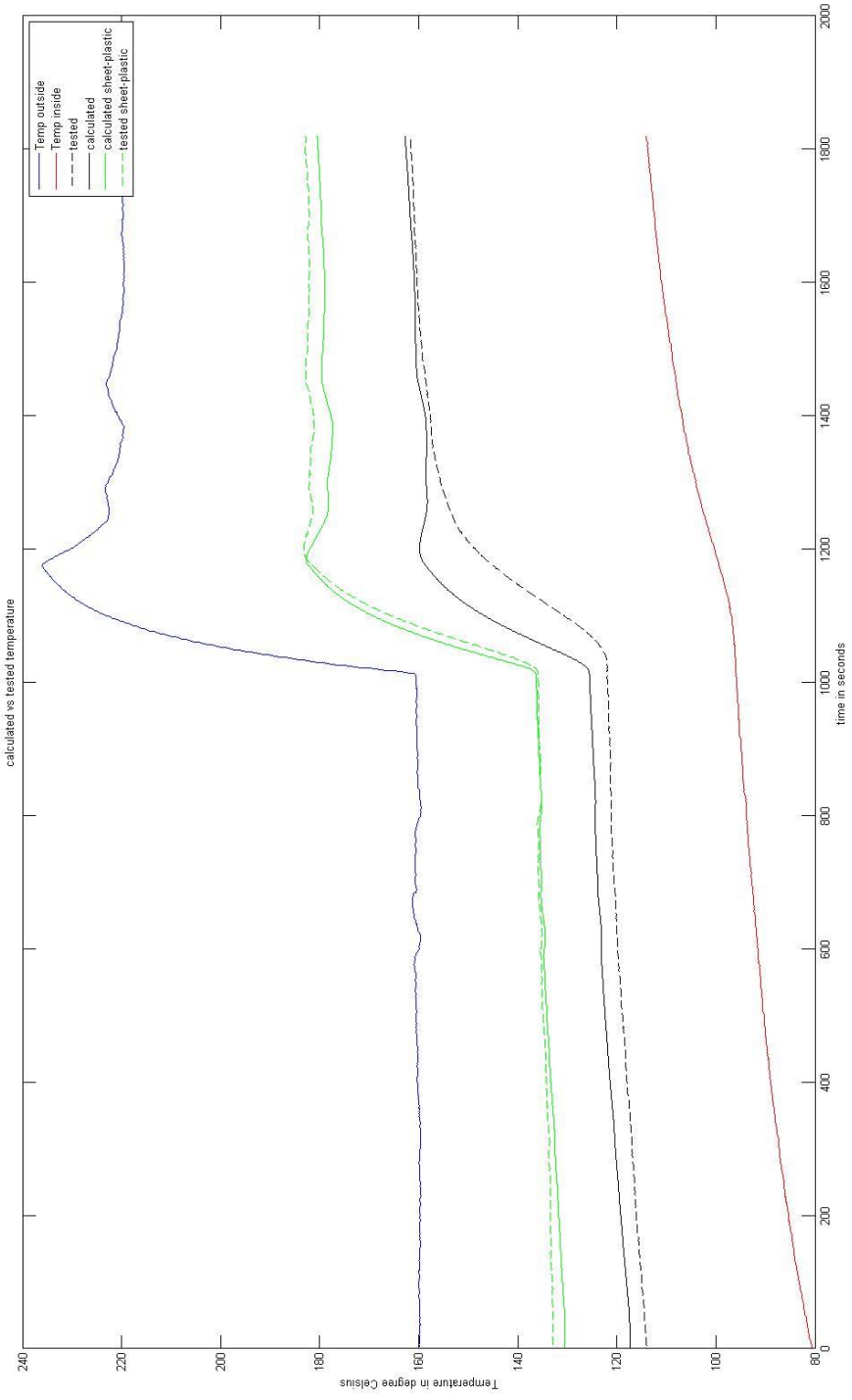
***Plot 22**: Graph from Case Study Test 6*



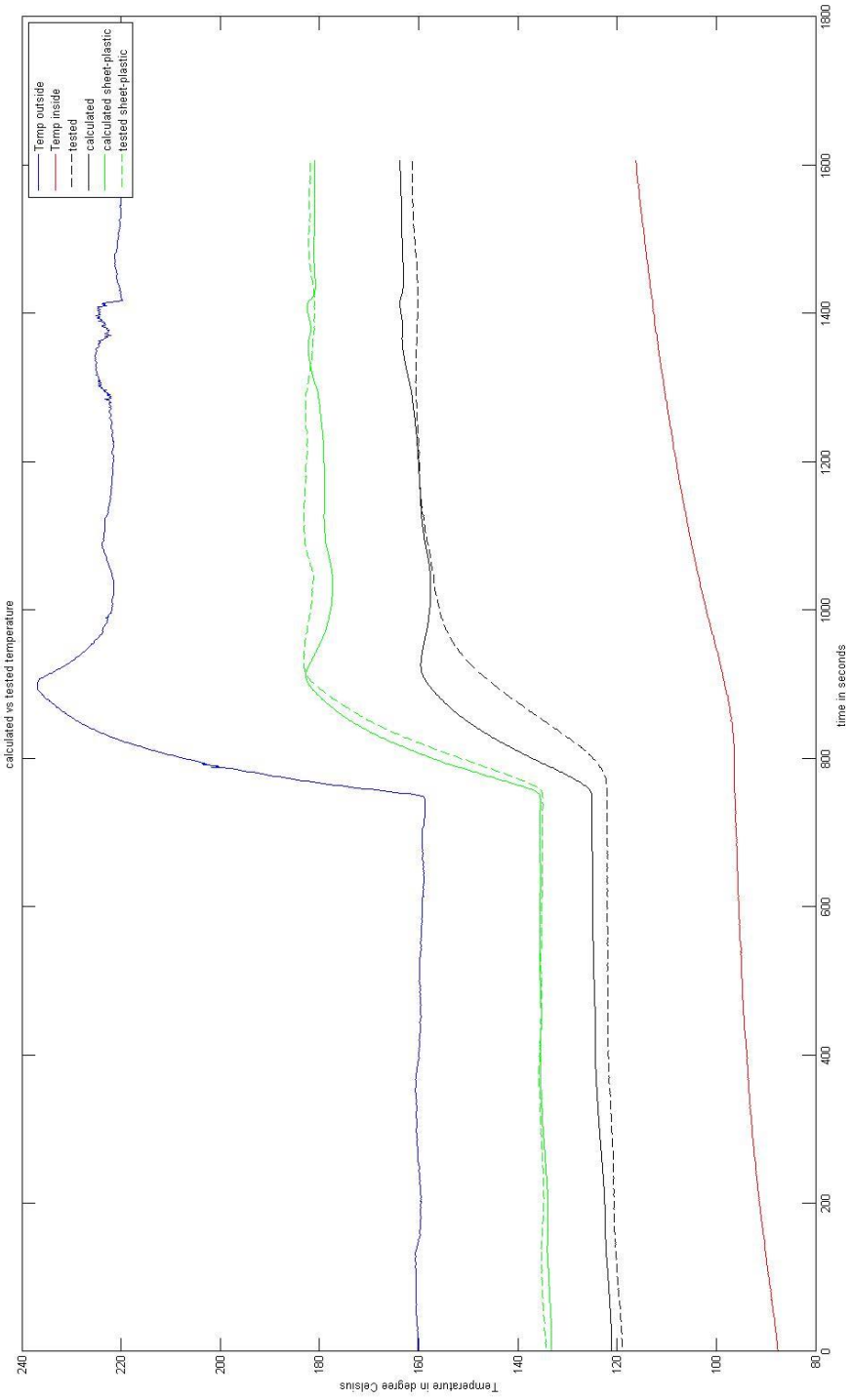
***Plot 23**: Graph from Case Study Test 7*



***Plot 24**: Graph from Case Study Test 8*



***Plot 25**: Graph from Case Study Test 9*



***Plot 26**: Graph from Case Study Test 10*

APPENDIX B – MATLAB CODE

Full code for the Volvo EEO software.

```
function varargout = volvo_eeo(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @volvo_eeo_OpeningFcn, ...
                   'gui_OutputFcn',  @volvo_eeo_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
%
%%% - Volvo Engine Encapsulation Optimizer -
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This software is created as a master's thesis for Volvo Cars by students
% from Chalmers University of Technology.
%
% Name:         Martin Jönsson and Daniel Talani
% Company:      Volvo Car Group
% Department:   Encapsulations, 97695, Engine Engineering
% Manager:      Per Burström
%
% School:       Chalmers University of Technology
% Department:   Product Development
% Examinor:     Kristina Wärmefjord
%
% Date:         2015-06-11
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes just before volvo_eeo is made visible.
function volvo_eeo_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% Choose default command line output for volvo_eeo
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);


% --- Outputs from this function are returned to the command line.
function varargout = volvo_eeo_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUT FUNCTIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in out_data_button.
function out_data_button_Callback(hObject, eventdata, handles)

% Loads data for outer temperature curve

global out_temp

[filename1,filepath1] = uigetfile({'*.*','All Files'}, 'Select Outer Temperature File');
  out_temp = load([filepath1 filename1]);

  if length(out_temp) > 0

     set(handles.reset,'Enable','on')
     set(handles.in_data_button,'Enable','on')

  end

% --- Executes on button press in in_data_button.
function in_data_button_Callback(hObject, eventdata, handles)

% Loads data for inner temperature curve

global in_temp1

[filename2,filepath2] = uigetfile({'*.*','All Files'}, 'Select Inner Temperature File');
  in_temp1 = load([filepath2 filename2]);

  if length(in_temp1) > 0

     set(handles.two_layer_enable,'Enable','on')
```

```
    set(handles.multi_layer_enable,'Enable','on')

  end

% --- Executes on button press in two_layer_enable.
function two_layer_enable_Callback(hObject, eventdata, handles)

% Check box for two layer optimizer method

two_layer_check = get(hObject, 'Value');

if two_layer_check > 0

    cla(handles.temp_plot,'reset');
    set(handles.temp_plot,'XColor', 'White');
    set(handles.temp_plot,'YColor', 'White');
    set(handles.temp_plot,'XTick', []);
    set(handles.temp_plot,'YTick', []);

    set(handles.total_thickness_text, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.max_temp_text2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.total_thickness_disp, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.max_temp_disp, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.total_thickness_disp, 'String', [])
    set(handles.max_temp_disp, 'String', [])

    set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
    set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
    set(handles.material_box_3, 'Position', [51.8 3.7 0.1 7])
    set(handles.material_box_4, 'Position', [51.8 3.7 0.1 7])
    set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])

    set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
    set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
    set(handles.material_thickness_3, 'Position', [51.8 5.3 0.1 4])
    set(handles.material_thickness_4, 'Position', [51.8 5.3 0.1 4])
    set(handles.material_thickness_5, 'Position', [51.8 5.3 0.1 4])

    set(handles.material_box_1, 'BackgroundColor', ([0.31 0.31 0.31]))
    set(handles.material_box_2, 'BackgroundColor', ([0.5 0.5 0.5]))
    set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'BackgroundColor', ([0.31 0.31 0.31]))
    set(handles.material_thickness_2, 'BackgroundColor', ([0.5 0.5 0.5]))
    set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))
```

```
set(handles.material_box_1, 'ForegroundColor', 'black')
set(handles.material_box_2, 'ForegroundColor', 'black')
set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'ForegroundColor', 'r')
set(handles.material_thickness_2, 'ForegroundColor', 'r')
set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'FontSize', 30)
set(handles.material_thickness_2, 'FontSize', 30)
set(handles.material_thickness_3, 'FontSize', 30)
set(handles.material_thickness_4, 'FontSize', 30)
set(handles.material_thickness_5, 'FontSize', 30)

set(handles.material_box_1, 'String', [1])
set(handles.material_box_2, 'String', [2])
set(handles.material_box_3, 'String', [])
set(handles.material_box_4, 'String', [])
set(handles.material_box_5, 'String', [])

set(handles.material_thickness_1, 'String', [])
set(handles.material_thickness_2, 'String', [])
set(handles.material_thickness_3, 'String', [])
set(handles.material_thickness_4, 'String', [])
set(handles.material_thickness_5, 'String', [])

set(handles.max_temp,'Enable','on')
set(handles.thickness,'Enable','on')
set(handles.multi_layer_enable,'Value', 0)
set(handles.material_layer_popup,'Enable','off')
set(handles.material_layer_popup,'Value', 1)

set(handles.max_temp,'String', [])
set(handles.thickness,'String', [])

set(handles.name_1, 'Enable', 'on')
set(handles.thickness_1, 'Enable', 'off')
set(handles.d_1, 'Enable', 'on')
set(handles.k_1, 'Enable', 'on')
set(handles.cp_1, 'Enable', 'on')

set(handles.name_2, 'Enable', 'on')
set(handles.thickness_2, 'Enable', 'off')
set(handles.d_2, 'Enable', 'on')
set(handles.k_2, 'Enable', 'on')
set(handles.cp_2, 'Enable', 'on')
```

```
set(handles.name_3, 'Enable', 'off')
set(handles.thickness_3, 'Enable', 'off')
set(handles.d_3, 'Enable', 'off')
set(handles.k_3, 'Enable', 'off')
set(handles.cp_3, 'Enable', 'off')

set(handles.name_4, 'Enable', 'off')
set(handles.thickness_4, 'Enable', 'off')
set(handles.d_4, 'Enable', 'off')
set(handles.k_4, 'Enable', 'off')
set(handles.cp_4, 'Enable', 'off')

set(handles.name_5, 'Enable', 'off')
set(handles.thickness_5, 'Enable', 'off')
set(handles.d_5, 'Enable', 'off')
set(handles.k_5, 'Enable', 'off')
set(handles.cp_5, 'Enable', 'off')

set(handles.name_1, 'String', [])
set(handles.thickness_1, 'String', [])
set(handles.d_1, 'String', [])
set(handles.k_1, 'String', [])
set(handles.cp_1, 'String', [])

set(handles.name_2, 'String', [])
set(handles.thickness_2, 'String', [])
set(handles.d_2, 'String', [])
set(handles.k_2, 'String', [])
set(handles.cp_2, 'String', [])

set(handles.name_3, 'String', [])
set(handles.thickness_3, 'String', [])
set(handles.d_3, 'String', [])
set(handles.k_3, 'String', [])
set(handles.cp_3, 'String', [])

set(handles.name_4, 'String', [])
set(handles.thickness_4, 'String', [])
set(handles.d_4, 'String', [])
set(handles.k_4, 'String', [])
set(handles.cp_4, 'String', [])

set(handles.name_5, 'String', [])
set(handles.thickness_5, 'String', [])
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

set(handles.calculate, 'Enable', 'on')
```

```
    set(handles.save_curve_button, 'Enable', 'off')
    set(handles.save_figure_button, 'Enable', 'off')

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [51.8 3.7 0.5 7])

else

    cla(handles.temp_plot,'reset');
    set(handles.temp_plot,'XColor', 'White');
    set(handles.temp_plot,'YColor', 'White');
    set(handles.temp_plot,'XTick', []);
    set(handles.temp_plot,'YTick', []);

    set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
    set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
    set(handles.material_box_3, 'Position', [51.8 3.7 0.1 7])
    set(handles.material_box_4, 'Position', [51.8 3.7 0.1 7])
    set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])

    set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
    set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
    set(handles.material_thickness_3, 'Position', [51.8 5.3 0.1 4])
    set(handles.material_thickness_4, 'Position', [51.8 5.3 0.1 4])
    set(handles.material_thickness_5, 'Position', [51.8 5.3 0.1 4])

    set(handles.material_box_1, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_2, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_2, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

    set(handles.material_box_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
```

```
set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'FontSize', 30)
set(handles.material_thickness_2, 'FontSize', 30)
set(handles.material_thickness_3, 'FontSize', 30)
set(handles.material_thickness_4, 'FontSize', 30)
set(handles.material_thickness_5, 'FontSize', 30)

set(handles.material_box_1, 'String', [])
set(handles.material_box_2, 'String', [])
set(handles.material_box_3, 'String', [])
set(handles.material_box_4, 'String', [])
set(handles.material_box_5, 'String', [])

set(handles.material_thickness_1, 'String', [])
set(handles.material_thickness_2, 'String', [])
set(handles.material_thickness_3, 'String', [])
set(handles.material_thickness_4, 'String', [])
set(handles.material_thickness_5, 'String', [])

set(handles.max_temp,'Enable','off')
set(handles.thickness,'Enable','off')
set(handles.max_temp,'String', [])
set(handles.thickness,'String', [])

set(handles.name_1, 'Enable', 'off')
set(handles.thickness_1, 'Enable', 'off')
set(handles.d_1, 'Enable', 'off')
set(handles.k_1, 'Enable', 'off')
set(handles.cp_1, 'Enable', 'off')

set(handles.name_2, 'Enable', 'off')
set(handles.thickness_2, 'Enable', 'off')
set(handles.d_2, 'Enable', 'off')
set(handles.k_2, 'Enable', 'off')
set(handles.cp_2, 'Enable', 'off')

set(handles.name_3, 'Enable', 'off')
set(handles.thickness_3, 'Enable', 'off')
set(handles.d_3, 'Enable', 'off')
set(handles.k_3, 'Enable', 'off')
set(handles.cp_3, 'Enable', 'off')

set(handles.name_4, 'Enable', 'off')
set(handles.thickness_4, 'Enable', 'off')
set(handles.d_4, 'Enable', 'off')
set(handles.k_4, 'Enable', 'off')
set(handles.cp_4, 'Enable', 'off')

set(handles.name_5, 'Enable', 'off')
```

```
    set(handles.thickness_5, 'Enable', 'off')
    set(handles.d_5, 'Enable', 'off')
    set(handles.k_5, 'Enable', 'off')
    set(handles.cp_5, 'Enable', 'off')

    set(handles.name_1, 'String', [])
    set(handles.thickness_1, 'String', [])
    set(handles.d_1, 'String', [])
    set(handles.k_1, 'String', [])
    set(handles.cp_1, 'String', [])

    set(handles.name_2, 'String', [])
    set(handles.thickness_2, 'String', [])
    set(handles.d_2, 'String', [])
    set(handles.k_2, 'String', [])
    set(handles.cp_2, 'String', [])

    set(handles.name_3, 'String', [])
    set(handles.thickness_3, 'String', [])
    set(handles.d_3, 'String', [])
    set(handles.k_3, 'String', [])
    set(handles.cp_3, 'String', [])

    set(handles.name_4, 'String', [])
    set(handles.thickness_4, 'String', [])
    set(handles.d_4, 'String', [])
    set(handles.k_4, 'String', [])
    set(handles.cp_4, 'String', [])

    set(handles.name_5, 'String', [])
    set(handles.thickness_5, 'String', [])
    set(handles.d_5, 'String', [])
    set(handles.k_5, 'String', [])
    set(handles.cp_5, 'String', [])

    set(handles.calculate, 'Enable', 'off')
    set(handles.save_curve_button, 'Enable', 'off')
    set(handles.save_figure_button, 'Enable', 'off')

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [51.8 3.7 0.5 7])


end

% --- Executes on button press in multi_layer_enable.
function multi_layer_enable_Callback(hObject, eventdata, handles)
```

```
% Check box for multi layer calculator method

multi_layer_check = get(hObject, 'Value');

if multi_layer_check > 0

    cla(handles.temp_plot,'reset');
    set(handles.temp_plot,'XColor', 'White');
    set(handles.temp_plot,'YColor', 'White');
    set(handles.temp_plot,'XTick', []);
    set(handles.temp_plot,'YTick', []);

    set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
    set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
    set(handles.material_box_3, 'Position', [49.6 3.7 0.1 7])
    set(handles.material_box_4, 'Position', [49.6 3.7 0.1 7])
    set(handles.material_box_5, 'Position', [49.6 3.7 0.1 7])

    set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
    set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
    set(handles.material_thickness_3, 'Position', [49.6 5.3 0.1 4])
    set(handles.material_thickness_4, 'Position', [49.6 5.3 0.1 4])
    set(handles.material_thickness_5, 'Position', [49.6 5.3 0.1 4])

    set(handles.material_box_1, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_2, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_2, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

    set(handles.material_box_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'FontSize', 30)
    set(handles.material_thickness_2, 'FontSize', 30)
```

```
set(handles.material_thickness_3, 'FontSize', 30)
set(handles.material_thickness_4, 'FontSize', 30)
set(handles.material_thickness_5, 'FontSize', 30)

set(handles.material_box_1, 'String', [])
set(handles.material_box_2, 'String', [])
set(handles.material_box_3, 'String', [])
set(handles.material_box_4, 'String', [])
set(handles.material_box_5, 'String', [])

set(handles.material_thickness_1, 'String', [])
set(handles.material_thickness_2, 'String', [])
set(handles.material_thickness_3, 'String', [])
set(handles.material_thickness_4, 'String', [])
set(handles.material_thickness_5, 'String', [])

set(handles.total_thickness_text, 'ForegroundColor', 'Black')
set(handles.max_temp_text2, 'ForegroundColor', 'Black')
set(handles.total_thickness_disp, 'BackgroundColor', 'White')
set(handles.max_temp_disp, 'BackgroundColor', 'White')
set(handles.total_thickness_disp, 'String', [])
set(handles.max_temp_disp, 'String', [])

set(handles.max_temp,'Enable','off')
set(handles.max_temp,'String', [])
set(handles.thickness,'Enable','off')
set(handles.thickness,'String', [])
set(handles.two_layer_enable,'Value', 0)
set(handles.material_layer_popup,'Enable','on')

set(handles.material_layer_popup,'Value', 1)

set(handles.name_1, 'Enable', 'off')
set(handles.thickness_1, 'Enable', 'off')
set(handles.d_1, 'Enable', 'off')
set(handles.k_1, 'Enable', 'off')
set(handles.cp_1, 'Enable', 'off')

set(handles.name_2, 'Enable', 'off')
set(handles.thickness_2, 'Enable', 'off')
set(handles.d_2, 'Enable', 'off')
set(handles.k_2, 'Enable', 'off')
set(handles.cp_2, 'Enable', 'off')

set(handles.name_3, 'Enable', 'off')
set(handles.thickness_3, 'Enable', 'off')
set(handles.d_3, 'Enable', 'off')
set(handles.k_3, 'Enable', 'off')
set(handles.cp_3, 'Enable', 'off')
```

```
set(handles.name_4, 'Enable', 'off')
set(handles.thickness_4, 'Enable', 'off')
set(handles.d_4, 'Enable', 'off')
set(handles.k_4, 'Enable', 'off')
set(handles.cp_4, 'Enable', 'off')

set(handles.name_5, 'Enable', 'off')
set(handles.thickness_5, 'Enable', 'off')
set(handles.d_5, 'Enable', 'off')
set(handles.k_5, 'Enable', 'off')
set(handles.cp_5, 'Enable', 'off')

set(handles.name_1, 'String', [])
set(handles.thickness_1, 'String', [])
set(handles.d_1, 'String', [])
set(handles.k_1, 'String', [])
set(handles.cp_1, 'String', [])

set(handles.name_2, 'String', [])
set(handles.thickness_2, 'String', [])
set(handles.d_2, 'String', [])
set(handles.k_2, 'String', [])
set(handles.cp_2, 'String', [])

set(handles.name_3, 'String', [])
set(handles.thickness_3, 'String', [])
set(handles.d_3, 'String', [])
set(handles.k_3, 'String', [])
set(handles.cp_3, 'String', [])

set(handles.name_4, 'String', [])
set(handles.thickness_4, 'String', [])
set(handles.d_4, 'String', [])
set(handles.k_4, 'String', [])
set(handles.cp_4, 'String', [])

set(handles.name_5, 'String', [])
set(handles.thickness_5, 'String', [])
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

set(handles.calculate, 'Enable', 'off')
set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.plot_line, 'Position', [51.8 3.7 0.5 7])
```

```
else

    cla(handles.temp_plot,'reset');
    set(handles.temp_plot,'XColor', 'White');
    set(handles.temp_plot,'YColor', 'White');
    set(handles.temp_plot,'XTick', []);
    set(handles.temp_plot,'YTick', []);

    set(handles.total_thickness_text, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.max_temp_text2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.total_thickness_disp, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.max_temp_disp, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.total_thickness_disp, 'String', [])
    set(handles.max_temp_disp, 'String', [])

    set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
    set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
    set(handles.material_box_3, 'Position', [51.8 3.7 0.1 7])
    set(handles.material_box_4, 'Position', [51.8 3.7 0.1 7])
    set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])

    set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
    set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
    set(handles.material_thickness_3, 'Position', [51.8 5.3 0.1 4])
    set(handles.material_thickness_4, 'Position', [51.8 5.3 0.1 4])
    set(handles.material_thickness_5, 'Position', [51.8 5.3 0.1 4])

    set(handles.material_box_1, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_2, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_2, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

    set(handles.material_box_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
```

```
set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'FontSize', 30)
set(handles.material_thickness_2, 'FontSize', 30)
set(handles.material_thickness_3, 'FontSize', 30)
set(handles.material_thickness_4, 'FontSize', 30)
set(handles.material_thickness_5, 'FontSize', 30)

set(handles.material_box_1, 'String', [])
set(handles.material_box_2, 'String', [])
set(handles.material_box_3, 'String', [])
set(handles.material_box_4, 'String', [])
set(handles.material_box_5, 'String', [])

set(handles.material_thickness_1, 'String', [])
set(handles.material_thickness_2, 'String', [])
set(handles.material_thickness_3, 'String', [])
set(handles.material_thickness_4, 'String', [])
set(handles.material_thickness_5, 'String', [])

set(handles.material_layer_popup,'Enable','off')
set(handles.material_layer_popup,'Value', 1)

set(handles.name_1, 'Enable', 'off')
set(handles.thickness_1, 'Enable', 'off')
set(handles.d_1, 'Enable', 'off')
set(handles.k_1, 'Enable', 'off')
set(handles.cp_1, 'Enable', 'off')

set(handles.name_2, 'Enable', 'off')
set(handles.thickness_2, 'Enable', 'off')
set(handles.d_2, 'Enable', 'off')
set(handles.k_2, 'Enable', 'off')
set(handles.cp_2, 'Enable', 'off')

set(handles.name_3, 'Enable', 'off')
set(handles.thickness_3, 'Enable', 'off')
set(handles.d_3, 'Enable', 'off')
set(handles.k_3, 'Enable', 'off')
set(handles.cp_3, 'Enable', 'off')

set(handles.name_4, 'Enable', 'off')
set(handles.thickness_4, 'Enable', 'off')
set(handles.d_4, 'Enable', 'off')
set(handles.k_4, 'Enable', 'off')
set(handles.cp_4, 'Enable', 'off')

set(handles.name_5, 'Enable', 'off')
set(handles.thickness_5, 'Enable', 'off')
```

```
    set(handles.d_5, 'Enable', 'off')
    set(handles.k_5, 'Enable', 'off')
    set(handles.cp_5, 'Enable', 'off')

    set(handles.name_1, 'String', [])
    set(handles.thickness_1, 'String', [])
    set(handles.d_1, 'String', [])
    set(handles.k_1, 'String', [])
    set(handles.cp_1, 'String', [])

    set(handles.name_2, 'String', [])
    set(handles.thickness_2, 'String', [])
    set(handles.d_2, 'String', [])
    set(handles.k_2, 'String', [])
    set(handles.cp_2, 'String', [])

    set(handles.name_3, 'String', [])
    set(handles.thickness_3, 'String', [])
    set(handles.d_3, 'String', [])
    set(handles.k_3, 'String', [])
    set(handles.cp_3, 'String', [])

    set(handles.name_4, 'String', [])
    set(handles.thickness_4, 'String', [])
    set(handles.d_4, 'String', [])
    set(handles.k_4, 'String', [])
    set(handles.cp_4, 'String', [])

    set(handles.name_5, 'String', [])
    set(handles.thickness_5, 'String', [])
    set(handles.d_5, 'String', [])
    set(handles.k_5, 'String', [])
    set(handles.cp_5, 'String', [])

    set(handles.calculate, 'Enable', 'off')
    set(handles.save_curve_button, 'Enable', 'off')
    set(handles.save_figure_button, 'Enable', 'off')

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [51.8 3.7 0.5 7])

end

% --- Sets the maximum Temperature on PUR surface.
function max_temp_Callback(hObject, eventdata, handles)
% -
%-
```

```
% --- Sets the overall thickness of encapsulation.
function thickness_Callback(hObject, eventdata, handles)
% -
% -

% --- Executes on selection change in material_layer_popup.
function material_layer_popup_Callback(hObject, eventdata, handles)

layers = get(hObject, 'Value');

if layers == 1;

    set(handles.name_1, 'Enable', 'off')
    set(handles.thickness_1, 'Enable', 'off')
    set(handles.d_1, 'Enable', 'off')
    set(handles.k_1, 'Enable', 'off')
    set(handles.cp_1, 'Enable', 'off')

    set(handles.name_2, 'Enable', 'off')
    set(handles.thickness_2, 'Enable', 'off')
    set(handles.d_2, 'Enable', 'off')
    set(handles.k_2, 'Enable', 'off')
    set(handles.cp_2, 'Enable', 'off')

    set(handles.name_3, 'Enable', 'off')
    set(handles.thickness_3, 'Enable', 'off')
    set(handles.d_3, 'Enable', 'off')
    set(handles.k_3, 'Enable', 'off')
    set(handles.cp_3, 'Enable', 'off')

    set(handles.name_4, 'Enable', 'off')
    set(handles.thickness_4, 'Enable', 'off')
    set(handles.d_4, 'Enable', 'off')
    set(handles.k_4, 'Enable', 'off')
    set(handles.cp_4, 'Enable', 'off')

    set(handles.name_5, 'Enable', 'off')
    set(handles.thickness_5, 'Enable', 'off')
    set(handles.d_5, 'Enable', 'off')
    set(handles.k_5, 'Enable', 'off')
    set(handles.cp_5, 'Enable', 'off')

    set(handles.name_1, 'String', [])
    set(handles.thickness_1, 'String', [])
    set(handles.d_1, 'String', [])
    set(handles.k_1, 'String', [])
    set(handles.cp_1, 'String', [])

    set(handles.name_2, 'String', [])
    set(handles.thickness_2, 'String', [])
```

```
set(handles.d_2, 'String', [])
set(handles.k_2, 'String', [])
set(handles.cp_2, 'String', [])

set(handles.name_3, 'String', [])
set(handles.thickness_3, 'String', [])
set(handles.d_3, 'String', [])
set(handles.k_3, 'String', [])
set(handles.cp_3, 'String', [])

set(handles.name_4, 'String', [])
set(handles.thickness_4, 'String', [])
set(handles.d_4, 'String', [])
set(handles.k_4, 'String', [])
set(handles.cp_4, 'String', [])

set(handles.name_5, 'String', [])
set(handles.thickness_5, 'String', [])
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

set(handles.calculate, 'Enable', 'off')
set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
set(handles.material_box_3, 'Position', [51.8 3.7 0.1 7])
set(handles.material_box_4, 'Position', [51.8 3.7 0.1 7])
set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])

set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
set(handles.material_thickness_3, 'Position', [51.8 5.3 0.1 4])
set(handles.material_thickness_4, 'Position', [51.8 5.3 0.1 4])
set(handles.material_thickness_5, 'Position', [51.8 5.3 0.1 4])

set(handles.material_box_1, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_2, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_2, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))
```

```
    set(handles.material_box_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_2, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
    set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'FontSize', 30)
    set(handles.material_thickness_2, 'FontSize', 30)
    set(handles.material_thickness_3, 'FontSize', 30)
    set(handles.material_thickness_4, 'FontSize', 30)
    set(handles.material_thickness_5, 'FontSize', 30)

    set(handles.material_box_1, 'String', [])
    set(handles.material_box_2, 'String', [])
    set(handles.material_box_3, 'String', [])
    set(handles.material_box_4, 'String', [])
    set(handles.material_box_5, 'String', [])

    set(handles.material_thickness_1, 'String', [])
    set(handles.material_thickness_2, 'String', [])
    set(handles.material_thickness_3, 'String', [])
    set(handles.material_thickness_4, 'String', [])
    set(handles.material_thickness_5, 'String', [])

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [30.8 3.7 0.5 7])

elseif layers == 2;

    set(handles.name_1, 'Enable', 'on')
    set(handles.thickness_1, 'Enable', 'on')
    set(handles.d_1, 'Enable', 'on')
    set(handles.k_1, 'Enable', 'on')
    set(handles.cp_1, 'Enable', 'on')

    set(handles.name_2, 'Enable', 'off')
    set(handles.thickness_2, 'Enable', 'off')
    set(handles.d_2, 'Enable', 'off')
    set(handles.k_2, 'Enable', 'off')
    set(handles.cp_2, 'Enable', 'off')

    set(handles.name_3, 'Enable', 'off')
```

```
set(handles.thickness_3, 'Enable', 'off')
set(handles.d_3, 'Enable', 'off')
set(handles.k_3, 'Enable', 'off')
set(handles.cp_3, 'Enable', 'off')

set(handles.name_4, 'Enable', 'off')
set(handles.thickness_4, 'Enable', 'off')
set(handles.d_4, 'Enable', 'off')
set(handles.k_4, 'Enable', 'off')
set(handles.cp_4, 'Enable', 'off')

set(handles.name_5, 'Enable', 'off')
set(handles.thickness_5, 'Enable', 'off')
set(handles.d_5, 'Enable', 'off')
set(handles.k_5, 'Enable', 'off')
set(handles.cp_5, 'Enable', 'off')

set(handles.name_2, 'String', [])
set(handles.thickness_2, 'String', [])
set(handles.d_2, 'String', [])
set(handles.k_2, 'String', [])
set(handles.cp_2, 'String', [])

set(handles.name_3, 'String', [])
set(handles.thickness_3, 'String', [])
set(handles.d_3, 'String', [])
set(handles.k_3, 'String', [])
set(handles.cp_3, 'String', [])

set(handles.name_4, 'String', [])
set(handles.thickness_4, 'String', [])
set(handles.d_4, 'String', [])
set(handles.k_4, 'String', [])
set(handles.cp_4, 'String', [])

set(handles.name_5, 'String', [])
set(handles.thickness_5, 'String', [])
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

set(handles.calculate, 'Enable', 'off')
set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
set(handles.material_box_3, 'Position', [51.8 3.7 0.1 7])
set(handles.material_box_4, 'Position', [51.8 3.7 0.1 7])
set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])
```

```
set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
set(handles.material_thickness_3, 'Position', [51.8 5.3 0.1 4])
set(handles.material_thickness_4, 'Position', [51.8 5.3 0.1 4])
set(handles.material_thickness_5, 'Position', [51.8 5.3 0.1 4])

set(handles.material_box_1, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_2, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_2, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_box_1, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_2, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_2, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'FontSize', 30)
set(handles.material_thickness_2, 'FontSize', 30)
set(handles.material_thickness_3, 'FontSize', 30)
set(handles.material_thickness_4, 'FontSize', 30)
set(handles.material_thickness_5, 'FontSize', 30)

set(handles.material_box_1, 'String', [])
set(handles.material_box_2, 'String', [])
set(handles.material_box_3, 'String', [])
set(handles.material_box_4, 'String', [])
set(handles.material_box_5, 'String', [])

set(handles.material_thickness_1, 'String', [])
set(handles.material_thickness_2, 'String', [])
set(handles.material_thickness_3, 'String', [])
set(handles.material_thickness_4, 'String', [])
set(handles.material_thickness_5, 'String', [])

set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
```

```
    set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [30.8 3.7 0.5 7])

elseif layers == 3;

    set(handles.name_1, 'Enable', 'on')
    set(handles.thickness_1, 'Enable', 'on')
    set(handles.d_1, 'Enable', 'on')
    set(handles.k_1, 'Enable', 'on')
    set(handles.cp_1, 'Enable', 'on')

    set(handles.name_2, 'Enable', 'on')
    set(handles.thickness_2, 'Enable', 'on')
    set(handles.d_2, 'Enable', 'on')
    set(handles.k_2, 'Enable', 'on')
    set(handles.cp_2, 'Enable', 'on')

    set(handles.name_3, 'Enable', 'off')
    set(handles.thickness_3, 'Enable', 'off')
    set(handles.d_3, 'Enable', 'off')
    set(handles.k_3, 'Enable', 'off')
    set(handles.cp_3, 'Enable', 'off')

    set(handles.name_4, 'Enable', 'off')
    set(handles.thickness_4, 'Enable', 'off')
    set(handles.d_4, 'Enable', 'off')
    set(handles.k_4, 'Enable', 'off')
    set(handles.cp_4, 'Enable', 'off')

    set(handles.name_5, 'Enable', 'off')
    set(handles.thickness_5, 'Enable', 'off')
    set(handles.d_5, 'Enable', 'off')
    set(handles.k_5, 'Enable', 'off')
    set(handles.cp_5, 'Enable', 'off')

    set(handles.name_3, 'String', [])
    set(handles.thickness_3, 'String', [])
    set(handles.d_3, 'String', [])
    set(handles.k_3, 'String', [])
    set(handles.cp_3, 'String', [])

    set(handles.name_4, 'String', [])
    set(handles.thickness_4, 'String', [])
    set(handles.d_4, 'String', [])
    set(handles.k_4, 'String', [])
    set(handles.cp_4, 'String', [])

    set(handles.name_5, 'String', [])
    set(handles.thickness_5, 'String', [])
```

```
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

set(handles.calculate, 'Enable', 'on')
set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
set(handles.material_box_3, 'Position', [51.8 3.7 0.1 7])
set(handles.material_box_4, 'Position', [51.8 3.7 0.1 7])
set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])

set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
set(handles.material_thickness_3, 'Position', [51.8 5.3 0.1 4])
set(handles.material_thickness_4, 'Position', [51.8 5.3 0.1 4])
set(handles.material_thickness_5, 'Position', [51.8 5.3 0.1 4])

set(handles.material_box_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_box_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_thickness_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_box_1, 'ForegroundColor', 'black')
set(handles.material_box_2, 'ForegroundColor', 'black')
set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'ForegroundColor', 'r')
set(handles.material_thickness_2, 'ForegroundColor', 'r')
set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'FontSize', 30)
set(handles.material_thickness_2, 'FontSize', 30)
set(handles.material_thickness_3, 'FontSize', 30)
set(handles.material_thickness_4, 'FontSize', 30)
set(handles.material_thickness_5, 'FontSize', 30)
```

```
    set(handles.material_box_1, 'String', [1])
    set(handles.material_box_2, 'String', [2])
    set(handles.material_box_3, 'String', [])
    set(handles.material_box_4, 'String', [])
    set(handles.material_box_5, 'String', [])

    set(handles.material_thickness_1, 'String', [])
    set(handles.material_thickness_2, 'String', [])
    set(handles.material_thickness_3, 'String', [])
    set(handles.material_thickness_4, 'String', [])
    set(handles.material_thickness_5, 'String', [])

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.31 0.31 0.31]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [30.8 3.7 0.5 7])

elseif layers == 4;

    set(handles.name_1, 'Enable', 'on')
    set(handles.thickness_1, 'Enable', 'on')
    set(handles.d_1, 'Enable', 'on')
    set(handles.k_1, 'Enable', 'on')
    set(handles.cp_1, 'Enable', 'on')

    set(handles.name_2, 'Enable', 'on')
    set(handles.thickness_2, 'Enable', 'on')
    set(handles.d_2, 'Enable', 'on')
    set(handles.k_2, 'Enable', 'on')
    set(handles.cp_2, 'Enable', 'on')

    set(handles.name_3, 'Enable', 'on')
    set(handles.thickness_3, 'Enable', 'on')
    set(handles.d_3, 'Enable', 'on')
    set(handles.k_3, 'Enable', 'on')
    set(handles.cp_3, 'Enable', 'on')

    set(handles.name_4, 'Enable', 'off')
    set(handles.thickness_4, 'Enable', 'off')
    set(handles.d_4, 'Enable', 'off')
    set(handles.k_4, 'Enable', 'off')
    set(handles.cp_4, 'Enable', 'off')

    set(handles.name_5, 'Enable', 'off')
    set(handles.thickness_5, 'Enable', 'off')
    set(handles.d_5, 'Enable', 'off')
    set(handles.k_5, 'Enable', 'off')
    set(handles.cp_5, 'Enable', 'off')

    set(handles.name_4, 'String', [])
```

```
set(handles.thickness_4, 'String', [])
set(handles.d_4, 'String', [])
set(handles.k_4, 'String', [])
set(handles.cp_4, 'String', [])

set(handles.name_5, 'String', [])
set(handles.thickness_5, 'String', [])
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

set(handles.calculate, 'Enable', 'on')
set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

set(handles.material_box_1, 'Position', [9.8 3.7 14 7])
set(handles.material_box_2, 'Position', [23.8 3.7 14 7])
set(handles.material_box_3, 'Position', [37.8 3.7 14 7])
set(handles.material_box_4, 'Position', [51.8 3.7 0.1 7])
set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])

set(handles.material_thickness_1, 'Position', [9.8 5.9 14 2.7])
set(handles.material_thickness_2, 'Position', [23.8 5.9 14 2.7])
set(handles.material_thickness_3, 'Position', [37.8 5.9 14 2.7])
set(handles.material_thickness_4, 'Position', [51.8 5.9 0.1 2.7])
set(handles.material_thickness_5, 'Position', [51.8 5.9 0.1 2.7])

set(handles.material_box_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_box_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_box_3, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_thickness_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_thickness_3, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_box_1, 'ForegroundColor', 'black')
set(handles.material_box_2, 'ForegroundColor', 'black')
set(handles.material_box_3, 'ForegroundColor', 'black')
set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'ForegroundColor', 'r')
set(handles.material_thickness_2, 'ForegroundColor', 'r')
set(handles.material_thickness_3, 'ForegroundColor', 'r')
set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))
```

```
    set(handles.material_thickness_1, 'FontSize', 20)
    set(handles.material_thickness_2, 'FontSize', 20)
    set(handles.material_thickness_3, 'FontSize', 20)
    set(handles.material_thickness_4, 'FontSize', 20)
    set(handles.material_thickness_5, 'FontSize', 20)

    set(handles.material_box_1, 'String', [1])
    set(handles.material_box_2, 'String', [2])
    set(handles.material_box_3, 'String', [3])
    set(handles.material_box_4, 'String', [])
    set(handles.material_box_5, 'String', [])

    set(handles.material_thickness_1, 'String', [])
    set(handles.material_thickness_2, 'String', [])
    set(handles.material_thickness_3, 'String', [])
    set(handles.material_thickness_4, 'String', [])
    set(handles.material_thickness_5, 'String', [])

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.31 0.31 0.31]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [37.8 3.7 0.5 7])

elseif layers == 5;

    set(handles.name_1, 'Enable', 'on')
    set(handles.thickness_1, 'Enable', 'on')
    set(handles.d_1, 'Enable', 'on')
    set(handles.k_1, 'Enable', 'on')
    set(handles.cp_1, 'Enable', 'on')

    set(handles.name_2, 'Enable', 'on')
    set(handles.thickness_2, 'Enable', 'on')
    set(handles.d_2, 'Enable', 'on')
    set(handles.k_2, 'Enable', 'on')
    set(handles.cp_2, 'Enable', 'on')

    set(handles.name_3, 'Enable', 'on')
    set(handles.thickness_3, 'Enable', 'on')
    set(handles.d_3, 'Enable', 'on')
    set(handles.k_3, 'Enable', 'on')
    set(handles.cp_3, 'Enable', 'on')

    set(handles.name_4, 'Enable', 'on')
    set(handles.thickness_4, 'Enable', 'on')
    set(handles.d_4, 'Enable', 'on')
    set(handles.k_4, 'Enable', 'on')
    set(handles.cp_4, 'Enable', 'on')
```

```
set(handles.name_5, 'Enable', 'off')
set(handles.thickness_5, 'Enable', 'off')
set(handles.d_5, 'Enable', 'off')
set(handles.k_5, 'Enable', 'off')
set(handles.cp_5, 'Enable', 'off')

set(handles.name_5, 'String', [])
set(handles.thickness_5, 'String', [])
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

set(handles.calculate, 'Enable', 'on')
set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

set(handles.material_box_1, 'Position', [9.8 3.7 10.5 7])
set(handles.material_box_2, 'Position', [20.3 3.7 10.5 7])
set(handles.material_box_3, 'Position', [30.8 3.7 10.5 7])
set(handles.material_box_4, 'Position', [41.3 3.7 10.5 7])
set(handles.material_box_5, 'Position', [51.8 3.7 0.1 7])

set(handles.material_thickness_1, 'Position', [9.8 6.2 10.5 2])
set(handles.material_thickness_2, 'Position', [20.3 6.2 10.5 2])
set(handles.material_thickness_3, 'Position', [30.8 6.2 10.5 2])
set(handles.material_thickness_4, 'Position', [41.4 6.2 10.5 2])
set(handles.material_thickness_5, 'Position', [51.8 6.2 0.1 2])

set(handles.material_box_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_box_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_box_3, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_box_4, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_thickness_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_thickness_3, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_thickness_4, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_box_1, 'ForegroundColor', 'black')
set(handles.material_box_2, 'ForegroundColor', 'black')
set(handles.material_box_3, 'ForegroundColor', 'black')
set(handles.material_box_4, 'ForegroundColor', 'black')
set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'ForegroundColor', 'r')
set(handles.material_thickness_2, 'ForegroundColor', 'r')
set(handles.material_thickness_3, 'ForegroundColor', 'r')
set(handles.material_thickness_4, 'ForegroundColor', 'r')
```

```
    set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

    set(handles.material_thickness_1, 'FontSize', 15)
    set(handles.material_thickness_2, 'FontSize', 15)
    set(handles.material_thickness_3, 'FontSize', 15)
    set(handles.material_thickness_4, 'FontSize', 15)
    set(handles.material_thickness_5, 'FontSize', 15)

    set(handles.material_box_1, 'String', [1])
    set(handles.material_box_2, 'String', [2])
    set(handles.material_box_3, 'String', [3])
    set(handles.material_box_4, 'String', [4])
    set(handles.material_box_5, 'String', [])

    set(handles.material_thickness_1, 'String', [])
    set(handles.material_thickness_2, 'String', [])
    set(handles.material_thickness_3, 'String', [])
    set(handles.material_thickness_4, 'String', [])
    set(handles.material_thickness_5, 'String', [])

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.31 0.31 0.31]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [41.3 3.7 0.5 7])

elseif layers == 6;

    set(handles.name_1, 'Enable', 'on')
    set(handles.thickness_1, 'Enable', 'on')
    set(handles.d_1, 'Enable', 'on')
    set(handles.k_1, 'Enable', 'on')
    set(handles.cp_1, 'Enable', 'on')

    set(handles.name_2, 'Enable', 'on')
    set(handles.thickness_2, 'Enable', 'on')
    set(handles.d_2, 'Enable', 'on')
    set(handles.k_2, 'Enable', 'on')
    set(handles.cp_2, 'Enable', 'on')

    set(handles.name_3, 'Enable', 'on')
    set(handles.thickness_3, 'Enable', 'on')
    set(handles.d_3, 'Enable', 'on')
    set(handles.k_3, 'Enable', 'on')
    set(handles.cp_3, 'Enable', 'on')

    set(handles.name_4, 'Enable', 'on')
    set(handles.thickness_4, 'Enable', 'on')
    set(handles.d_4, 'Enable', 'on')
    set(handles.k_4, 'Enable', 'on')
    set(handles.cp_4, 'Enable', 'on')
```

```
set(handles.name_5, 'Enable', 'on')
set(handles.thickness_5, 'Enable', 'on')
set(handles.d_5, 'Enable', 'on')
set(handles.k_5, 'Enable', 'on')
set(handles.cp_5, 'Enable', 'on')

set(handles.calculate, 'Enable', 'on')
set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

set(handles.material_box_1, 'Position', [9.8 3.7 8.4 7])
set(handles.material_box_2, 'Position', [18.2 3.7 8.4 7])
set(handles.material_box_3, 'Position', [26.6 3.7 8.4 7])
set(handles.material_box_4, 'Position', [35 3.7 8.4 7])
set(handles.material_box_5, 'Position', [43.4 3.7 8.4 7])

set(handles.material_thickness_1, 'Position', [9.8 6.4 8.4 1.7])
set(handles.material_thickness_2, 'Position', [18.2 6.4 8.4 1.7])
set(handles.material_thickness_3, 'Position', [26.6 6.4 8.4 1.7])
set(handles.material_thickness_4, 'Position', [35 6.4 8.4 1.7])
set(handles.material_thickness_5, 'Position', [43.4 6.4 8.4 1.7])

set(handles.material_box_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_box_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_box_3, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_box_4, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_box_5, 'BackgroundColor', ([0.31 0.31 0.31]))

set(handles.material_thickness_1, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_thickness_2, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_thickness_3, 'BackgroundColor', ([0.31 0.31 0.31]))
set(handles.material_thickness_4, 'BackgroundColor', ([0.5 0.5 0.5]))
set(handles.material_thickness_5, 'BackgroundColor', ([0.31 0.31 0.31]))

set(handles.material_box_1, 'ForegroundColor', 'black')
set(handles.material_box_2, 'ForegroundColor', 'black')
set(handles.material_box_3, 'ForegroundColor', 'black')
set(handles.material_box_4, 'ForegroundColor', 'black')
set(handles.material_box_5, 'ForegroundColor', 'black')

set(handles.material_thickness_1, 'ForegroundColor', 'r')
set(handles.material_thickness_2, 'ForegroundColor', 'r')
set(handles.material_thickness_3, 'ForegroundColor', 'r')
set(handles.material_thickness_4, 'ForegroundColor', 'r')
set(handles.material_thickness_5, 'ForegroundColor', 'r')

set(handles.material_thickness_1, 'FontSize', 12)
set(handles.material_thickness_2, 'FontSize', 12)
set(handles.material_thickness_3, 'FontSize', 12)
```

```
    set(handles.material_thickness_4, 'FontSize', 12)
    set(handles.material_thickness_5, 'FontSize', 12)

    set(handles.material_box_1, 'String', [1])
    set(handles.material_box_2, 'String', [2])
    set(handles.material_box_3, 'String', [3])
    set(handles.material_box_4, 'String', [4])
    set(handles.material_box_5, 'String', [5])

    set(handles.material_thickness_1, 'String', [])
    set(handles.material_thickness_2, 'String', [])
    set(handles.material_thickness_3, 'String', [])
    set(handles.material_thickness_4, 'String', [])
    set(handles.material_thickness_5, 'String', [])

    set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'BackgroundColor', ([0.31 0.31 0.31]))
    set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
    set(handles.plot_line, 'Position', [43.4 3.7 0.5 7])

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MATERIAL FUNCTIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Material 1
function name_1_Callback(hObject, eventdata, handles)
% -
% -

function thickness_1_Callback(hObject, eventdata, handles)
% -
% -

function d_1_Callback(hObject, eventdata, handles)
% -
% -

function k_1_Callback(hObject, eventdata, handles)
% -
% -

function cp_1_Callback(hObject, eventdata, handles)
% -
% -
```

```
% --- Material 2
function name_2_Callback(hObject, eventdata, handles)
% -
% -

function thickness_2_Callback(hObject, eventdata, handles)
% -
% -

function d_2_Callback(hObject, eventdata, handles)
% -
% -

function k_2_Callback(hObject, eventdata, handles)
% -
% -

function cp_2_Callback(hObject, eventdata, handles)
% -
% -


% --- Material 3
function name_3_Callback(hObject, eventdata, handles)
% -
% -

function thickness_3_Callback(hObject, eventdata, handles)
% -
% -

function d_3_Callback(hObject, eventdata, handles)
% -
% -

function k_3_Callback(hObject, eventdata, handles)
% -
% -

function cp_3_Callback(hObject, eventdata, handles)
% -
% -


% --- Material 4
function name_4_Callback(hObject, eventdata, handles)
% -
% -
```

```
function thickness_4_Callback(hObject, eventdata, handles)
% -
% -

function d_4_Callback(hObject, eventdata, handles)
% -
% -

function k_4_Callback(hObject, eventdata, handles)
% -
% -

function cp_4_Callback(hObject, eventdata, handles)
% -
% -


% --- Material 5
function name_5_Callback(hObject, eventdata, handles)
% -
% -

function thickness_5_Callback(hObject, eventdata, handles)
% -
% -

function d_5_Callback(hObject, eventdata, handles)
% -
% -

function k_5_Callback(hObject, eventdata, handles)
% -
% -

function cp_5_Callback(hObject, eventdata, handles)
% -
% -


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OUTPUT FUNCTIONS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in calculate.
function calculate_Callback(hObject, eventdata, handles)

% Calculationz
```

```
global out_temp in_temp1 curve

cla(handles.temp_plot,'reset');
set(handles.temp_plot,'XColor', 'White');
set(handles.temp_plot,'YColor', 'White');
set(handles.temp_plot,'XTick', []);
set(handles.temp_plot,'YTick', []);

set(handles.material_thickness_1, 'String', [])
set(handles.material_thickness_2, 'String', [])

set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))

two_layer_check = get(handles.two_layer_enable, 'Value');
multi_layer_check = get(handles.multi_layer_enable, 'Value');

oo = length(out_temp);
ii = length(in_temp1);

if oo > ii
   d = oo - ii;
   in_temp = ones(oo,1);
   in_temp(1:1:ii) = in_temp1;
   in_temp(ii+1:1:oo) = in_temp1(length(in_temp1));
elseif oo < ii
    in_temp = ones(oo,1);
    in_temp(1:1:oo) = in_temp1(1:1:oo);
else

   in_temp = in_temp1;

end


if two_layer_check > multi_layer_check                    % Two Layer Optimizer
Method

   set(handles.figure1, 'pointer', 'watch')
   drawnow;

   % Input Data

   L = str2num(get(handles.thickness, 'String'))/1000;          % Overall Thickness [m]
   nnod = 200;                                          % Number of Nodes
   dx = L/nnod;                                         % Step Size in X Dir [m]
   Tmax = str2num(get(handles.max_temp, 'String'));             % Max Temp PUR-
Plastic Transition
```

```
   xStep = [0:(L/(nnod+1)):L];
   TA = out_temp;                                    % Temperature on west side between
0 and time_End
   TB = in_temp;                                    % Temperature on east side between
0 and time_End
   Tinitial = linspace(TA(1),TB(1),nnod+2);                  % Initial temperature in
nodes [degree C]

   time_End = length(TA);                              % Total time measured in minutes
   noT = length(TA);                                % Number of iteration times
   dt = 1;                                       % Step time [s]

   % Material Data

   k = [str2num(get(handles.k_1, 'String')) str2num(get(handles.k_2, 'String'))];      % thermal
conductivity [W/m.K]
   rho = [str2num(get(handles.d_1, 'String')) str2num(get(handles.d_2, 'String'))];    % density
plastic  [kg/m^3]
   Cp = [str2num(get(handles.cp_1, 'String')) str2num(get(handles.cp_2, 'String'))];   %
specific heat for 1)PUR (1400-1500)    2)PA66GF30 [J/m^3.K]]

   rhoC = rho.*Cp; % (density)/specific heat rho/Cp) [J/m^3.K]

   % Boundary condition
   optim=zeros(nnod+1,1);
   for m=0:1:nnod

   W = ones(1,nnod);
   W(1,m+1:nnod)=2;

   A = zeros(noT+1,nnod+2);

   A(1,:) = Tinitial;       % initial temperature

   A(:,nnod+2) = TB(1,:);   % east temperature BC
   A(:,1) = TA(1,:);        % constant temp on west side of TA degrees
   %creating Su
   Su = zeros(noT,nnod);
   for v = 1:1:noT
   Su(v,nnod) = 2*k(W(nnod))*TB(v)/dx;
   Su(v,1) = 2*k(W(nnod))*TA(v)/dx;
   end
   Sp = zeros(1,nnod);
   Sp(1,nnod) = -2*k(W(nnod))/dx;
   Sp(1,1) = -2*k(W(nnod))/dx;

   %create vectors
   apo = zeros(1,nnod);
   aw = zeros(1,nnod);
   ae = zeros(1,nnod);
```

```
ap = zeros(1,nnod);
tpo = zeros(1,nnod);
for i=1:1:nnod;

apo(i) = rho(W(i)).*Cp(W(i))*(dx/dt);
aw(i) = k(W(i))/dx;
ae(i) = k(W(i))/dx;

ap(i)=aw(W(i))+ae(W(i))+apo(W(i))-Sp(W(i));
tpo(i) = rho(W(i)).*Cp(W(i))*dx/dt;

end
ae(1,nnod) = 0;
aw(1,1) = 0;

T = zeros(noT,nnod);

M = zeros(nnod,nnod);

for i = 1:1:nnod
        M(i,i)=aw(i)+ae(i)+apo(i)-Sp(i);
end
for j=2:1:nnod
   M(j-1,j) = -k(W(j-1))/dx;
   M(j,j-1) = -k(W(j))/dx;
end
M(nnod,nnod) = aw(nnod)+ae(nnod)+apo(nnod)-Sp(nnod);

%
T=A(:,2:nnod+1);

for j=1:1:noT;
   T(j+1,:)=M\((T(j,:).*tpo(1,:))+Su(j,:))';
end
TT = zeros(noT+1,nnod+2);
for i=2:1:noT+1
TT(i,2:nnod+1)=T(i-1,:);
end
TT(1,:) = Tinitial;

for i =2:1:noT+1
TT(i,nnod+2) = TB(i-1,:); %TB input
TT(i,1) = TA(i-1,:);   %TA input
end

% Plot

optim(m+1,1) = max(TT(:,m+1));
```

```
    end

    Opt_thickness = abs(optim-Tmax);
    [idx idx] =min(Opt_thickness);

    %

    axes (handles.temp_plot);
    time_min = 0:(dt):time_End-dt;
    plot(time_min,TA(:,:),'color','r')
    hold on
    plot(time_min,TB(:,:),'color','b')
    hold on
    plot(time_min,TT(2:1:noT+1,idx),'color','g')
    hold on
    plot([0 length(TA)],[Tmax Tmax],'black--')

    curve = TT(2:1:noT+1,idx);

    xlabel('Time [s]')
    ylabel('Temperature [C]')

    axis([0 time_End 40 max(TA)+10])
    legend('Outer Temp','Inner Temp','Optimized Temp','Max Temp','Location','northwest')
    hold on

    L1 = (L/nnod*(idx-1));
    L2 = (L-L1);

    set(handles.material_thickness_1, 'String', L1*1000)
    set(handles.material_thickness_2, 'String', L2*1000)

    set(handles.red_line, 'BackgroundColor', 'red')
    set(handles.plot_line, 'BackgroundColor', 'green')
    set(handles.blue_line, 'BackgroundColor', 'blue')

    set(handles.figure1, 'pointer', 'arrow')
    set(handles.save_curve_button, 'Enable', 'on')
    set(handles.save_figure_button, 'Enable', 'on')

elseif multi_layer_check > two_layer_check                    % Multi Layer Calculate
Method

    set(handles.material_thickness_1, 'String', [])
    set(handles.material_thickness_2, 'String', [])
    set(handles.material_thickness_3, 'String', [])
    set(handles.material_thickness_4, 'String', [])
    set(handles.material_thickness_5, 'String', [])
    set(handles.total_thickness_disp, 'String', [])
    set(handles.max_temp_disp, 'String', [])
```

```
    set(handles.figure1, 'pointer', 'watch')
    drawnow;

    % Number of layers
    NumberOfLayers = (get(handles.material_layer_popup, 'Value'))-1;       % Number of
Layers                          % Number of material used through the thickness of the
product

    % Lengths
    l1 = str2num(get(handles.thickness_1, 'String'));                  % Length Layer 1 [mm]
    l2 = str2num(get(handles.thickness_2, 'String'));                  % Length Layer 2 [mm]
    l3 = str2num(get(handles.thickness_3, 'String'));                  % Length Layer 3 [mm]
    l4 = str2num(get(handles.thickness_4, 'String'));                  % Length Layer 4 [mm]
    l5 = str2num(get(handles.thickness_5, 'String'));                  % Length Layer 5 [mm]

    l=[l1 l2 l3 l4 l5].*0.001;                                  % Translation of length from [mm] to
[m] into a vector
    L = zeros(1,NumberOfLayers);
    L = l(1,1:NumberOfLayers);                                        % vector with each used material
thickness in meter
    nnod = 500;                                                % number of nodes. Will adjust +-1 to fit
node distribution to materials
    dx = sum(l(1:NumberOfLayers))/nnod;

    % Percent Node Distribution
    node_L = zeros(1,NumberOfLayers);

    for i=1:NumberOfLayers
        node_L(1,i) = round(nnod*(L(i)/(sum(L))));
    end

    while nnod~=sum(node_L)
        nnod=sum(node_L);
    end

    % K, Thermal Conductivity [W/m.K]

    k1 = str2num(get(handles.k_1, 'String'));                      % Thermal Conductivity
Material 1
    k2 = str2num(get(handles.k_2, 'String'));                      % Thermal Conductivity
Material 2
    k3 = str2num(get(handles.k_3, 'String'));                      % Thermal Conductivity
Material 3
    k4 = str2num(get(handles.k_4, 'String'));                      % Thermal Conductivity
Material 4
    k5 = str2num(get(handles.k_5, 'String'));                      % Thermal Conductivity
Material 5

    kk = [k1 k2 k3 k4 k5];
```

```
k = zeros(1,NumberOfLayers);
k = kk(1,1:NumberOfLayers);

% Rho, Density [kg/m3]

rho1 = str2num(get(handles.d_1, 'String'));          % Density Material 1 [kg/m^3]
rho2 = str2num(get(handles.d_2, 'String'));          % Density Material 1 [kg/m^3]
rho3 = str2num(get(handles.d_3, 'String'));          % Density Material 1 [kg/m^3]
rho4 = str2num(get(handles.d_4, 'String'));          % Density Material 1 [kg/m^3]
rho5 = str2num(get(handles.d_5, 'String'));          % Density Material 1 [kg/m^3]

RHO = [rho1 rho2 rho3 rho4 rho5];
rho = zeros(1,NumberOfLayers);
rho = RHO(1,1:NumberOfLayers);

% Cp, Specific Heat [J/kg.K]

Cp1 = str2num(get(handles.cp_1, 'String'));          % Specific Heat Material 1
Cp2 = str2num(get(handles.cp_2, 'String'));          % Specific Heat Material 2
Cp3 = str2num(get(handles.cp_3, 'String'));          % Specific Heat Material 3
Cp4 = str2num(get(handles.cp_4, 'String'));          % Specific Heat Material 4
Cp5 = str2num(get(handles.cp_5, 'String'));          % Specific Heat Material 5

CP = [Cp1 Cp2 Cp3 Cp4 Cp5];
Cp = zeros(1,NumberOfLayers);
Cp = CP(1,1:NumberOfLayers);

TA = out_temp;                                  % Temperature on west side between 0
and time_End
TB = in_temp;                                   % Temperature on east side between 0
and time_End

Tinitial = linspace(TA(1),TB(1),nnod+2);            % Initial Temp

time_End = length(TA);                              % Total time measured in seconds
noT = length(TA);                                   % Number of iteration times in sec
dt = 1;                                         % Step Time [s]

% Create W vector controlling each node data

W = ones(1,nnod);

for  i = 2:1:NumberOfLayers-1
   W(1,(sum(node_L(1:i-1))+1):(sum(node_L(1:i+1)))) = i;
end

W(sum(node_L(1:(NumberOfLayers-1)))+1:nnod) = NumberOfLayers;

A = zeros(noT+1,nnod+2);
A(1,:) = Tinitial;                                  % Initial Temperature
```

```
   A(:,nnod+2) = TB(1,:);                    % East Temperature BC
   A(:,1) = TA(1,:);                         % Constant temp on west side of TA
degrees

   % Creating Su
   Su = zeros(noT,nnod);

   for v = 1:1:noT
      Su(v,nnod) = (2*k(W(nnod))*TB(v)/dx);
      Su(v,1) = (2*k(W(1))*TA(v)/dx);
   end

   Sp = zeros(1,nnod);
   Sp(1,nnod) = -2*k(W(nnod))/dx;
   Sp(1,1) = -2*k(W(1))/dx;

   % Create Vectors
   apo = zeros(1,nnod);
   aw = zeros(1,nnod);
   ae = zeros(1,nnod);

   ap = zeros(1,nnod);
   tpo = zeros(1,nnod);

   for i=1:1:nnod;
      apo(i) = rho(W(i)).*Cp(W(i))*(dx/dt);
      aw(i) = k(W(i))/dx;
      ae(i) = k(W(i))/dx;

      ap(i)=aw(W(i))+ae(W(i))+apo(W(i))-Sp(W(i));
      tpo(i) = rho(W(i)).*Cp(W(i))*(dx/dt);
   end

   ae(1,nnod) = 0;
   aw(1,1) = 0;

   % M Matrix
   T = zeros(noT,nnod);
   M = zeros(nnod,nnod);

   for i = 1:1:nnod
      M(i,i)=aw(i)+ae(i)+apo(i)-Sp(i);
   end

   for j=2:1:nnod
      M(j-1,j) = -k(W(j-1))/dx;
      M(j,j-1) = -k(W(j))/dx;
   end

   T=A(:,2:nnod+1);
```

```
for j=1:1:noT;
   T(j+1,:)=M\((T(j,:).*tpo(1,:))+Su(j,:))';
end

TT = zeros(noT+1,nnod+2);

for i=2:1:noT+1
   TT(i,2:nnod+1)=T(i-1,:);
end

TT(1,:) = Tinitial;

for i =2:1:noT+1
   TT(i,nnod+2) = TB(i-1,:);                          % TB Input
   TT(i,1) = TA(i-1,:);                               % TA Input
end

time_min = 0:(dt):time_End-dt;

% Temperature at surface for protected material
Tmax2 = max(T(2:1:noT+1,sum(node_L(1:(NumberOfLayers-1)))));


% Plot 2
axes (handles.temp_plot);
plot(time_min,TA,'r');
hold on
plot(time_min,TB,'b');
hold on
plot(time_min,T(2:1:noT+1,sum(node_L(1:(NumberOfLayers-1)))),'g')
hold on
plot([0 length(TA)],[Tmax2 Tmax2],'black--')

curve = T(2:1:noT+1,sum(node_L(1:(NumberOfLayers-1))));

axis([0 length(TA) 40 max(TA)+10]);
legend('Temp Outside','Temp Inside','Temp PUR', 'Max Temp', 'Location', 'northwest')
xlabel('Time [s]')
ylabel('Temperature [C]')

set(handles.material_thickness_1, 'String', [get(handles.thickness_1, 'String')])
set(handles.material_thickness_2, 'String', [get(handles.thickness_2, 'String')])
set(handles.material_thickness_3, 'String', [get(handles.thickness_3, 'String')])
set(handles.material_thickness_4, 'String', [get(handles.thickness_4, 'String')])
set(handles.material_thickness_5, 'String', [get(handles.thickness_5, 'String')])

set(handles.red_line, 'BackgroundColor', 'red')
set(handles.plot_line, 'BackgroundColor', 'green')
set(handles.blue_line, 'BackgroundColor', 'blue')
```

```
    set(handles.total_thickness_disp, 'String', sum(L)*1000)
    set(handles.max_temp_disp, 'String', Tmax2)

    set(handles.figure1, 'pointer', 'arrow')
    set(handles.save_curve_button, 'Enable', 'on')
    set(handles.save_figure_button, 'Enable', 'on')

end

% --- Executes on button press in reset.
function reset_Callback(hObject, eventdata, handles)

% Resets all values to 0

set(handles.figure1, 'pointer', 'arrow')

% Input Data
global out_temp in_temp1

out_temp = [];
in_temp1 = [];
set(handles.in_data_button,'Enable','off')
set(handles.two_layer_enable,'Enable','off')
set(handles.two_layer_enable,'Value', 0)
set(handles.multi_layer_enable,'Enable','off')
set(handles.multi_layer_enable,'Value', 0)

set(handles.max_temp,'Enable','off')
set(handles.max_temp,'String', [])
set(handles.thickness,'Enable','off')
set(handles.thickness,'String', [])

set(handles.material_layer_popup,'Enable','off')
set(handles.material_layer_popup,'Value', 1)

% Material Data
set(handles.name_1, 'Enable', 'off')
set(handles.thickness_1, 'Enable', 'off')
set(handles.d_1, 'Enable', 'off')
set(handles.k_1, 'Enable', 'off')
set(handles.cp_1, 'Enable', 'off')

set(handles.name_2, 'Enable', 'off')
set(handles.thickness_2, 'Enable', 'off')
set(handles.d_2, 'Enable', 'off')
set(handles.k_2, 'Enable', 'off')
set(handles.cp_2, 'Enable', 'off')

set(handles.name_3, 'Enable', 'off')
```

```
set(handles.thickness_3, 'Enable', 'off')
set(handles.d_3, 'Enable', 'off')
set(handles.k_3, 'Enable', 'off')
set(handles.cp_3, 'Enable', 'off')

set(handles.name_4, 'Enable', 'off')
set(handles.thickness_4, 'Enable', 'off')
set(handles.d_4, 'Enable', 'off')
set(handles.k_4, 'Enable', 'off')
set(handles.cp_4, 'Enable', 'off')

set(handles.name_5, 'Enable', 'off')
set(handles.thickness_5, 'Enable', 'off')
set(handles.d_5, 'Enable', 'off')
set(handles.k_5, 'Enable', 'off')
set(handles.cp_5, 'Enable', 'off')

set(handles.name_1, 'String', [])
set(handles.thickness_1, 'String', [])
set(handles.d_1, 'String', [])
set(handles.k_1, 'String', [])
set(handles.cp_1, 'String', [])

set(handles.name_2, 'String', [])
set(handles.thickness_2, 'String', [])
set(handles.d_2, 'String', [])
set(handles.k_2, 'String', [])
set(handles.cp_2, 'String', [])

set(handles.name_3, 'String', [])
set(handles.thickness_3, 'String', [])
set(handles.d_3, 'String', [])
set(handles.k_3, 'String', [])
set(handles.cp_3, 'String', [])

set(handles.name_4, 'String', [])
set(handles.thickness_4, 'String', [])
set(handles.d_4, 'String', [])
set(handles.k_4, 'String', [])
set(handles.cp_4, 'String', [])

set(handles.name_5, 'String', [])
set(handles.thickness_5, 'String', [])
set(handles.d_5, 'String', [])
set(handles.k_5, 'String', [])
set(handles.cp_5, 'String', [])

% Output Data
set(handles.reset,'Enable','off')
set(handles.calculate, 'Enable', 'off')
```

```
cla(handles.temp_plot,'reset');
set(handles.temp_plot,'XColor', 'White');
set(handles.temp_plot,'YColor', 'White');
set(handles.temp_plot,'XTick', []);
set(handles.temp_plot,'YTick', []);

set(handles.material_box_1, 'Position', [9.8 3.7 21 7])
set(handles.material_box_2, 'Position', [30.8 3.7 21 7])
set(handles.material_box_3, 'Position', [49.6 3.7 0.1 7])
set(handles.material_box_4, 'Position', [49.6 3.7 0.1 7])
set(handles.material_box_5, 'Position', [49.6 3.7 0.1 7])

set(handles.material_thickness_1, 'Position', [9.8 5.3 21 4])
set(handles.material_thickness_2, 'Position', [30.8 5.3 21 4])
set(handles.material_thickness_3, 'Position', [49.6 5.3 0.1 4])
set(handles.material_thickness_4, 'Position', [49.6 5.3 0.1 4])
set(handles.material_thickness_5, 'Position', [49.6 5.3 0.1 4])

set(handles.material_box_1, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_2, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_2, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_3, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'BackgroundColor', ([0.94 0.94 0.94]))

set(handles.material_box_1, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_2, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_box_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_2, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_3, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_4, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.material_thickness_5, 'ForegroundColor', ([0.94 0.94 0.94]))

set(handles.material_thickness_1, 'FontSize', 30)
set(handles.material_thickness_2, 'FontSize', 30)
set(handles.material_thickness_3, 'FontSize', 30)
set(handles.material_thickness_4, 'FontSize', 30)
set(handles.material_thickness_5, 'FontSize', 30)

set(handles.material_box_1, 'String', [])
```

```
set(handles.material_box_2, 'String', [])
set(handles.material_box_3, 'String', [])
set(handles.material_box_4, 'String', [])
set(handles.material_box_5, 'String', [])

set(handles.material_thickness_1, 'String', [])
set(handles.material_thickness_2, 'String', [])
set(handles.material_thickness_3, 'String', [])
set(handles.material_thickness_4, 'String', [])
set(handles.material_thickness_5, 'String', [])

set(handles.total_thickness_text, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.max_temp_text2, 'ForegroundColor', ([0.94 0.94 0.94]))
set(handles.total_thickness_disp, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.max_temp_disp, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.total_thickness_disp, 'String', [])
set(handles.max_temp_disp, 'String', [])

set(handles.red_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.plot_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.blue_line, 'BackgroundColor', ([0.94 0.94 0.94]))
set(handles.plot_line, 'Position', [51.8 3.7 0.5 7])

set(handles.save_curve_button, 'Enable', 'off')
set(handles.save_figure_button, 'Enable', 'off')

% --- Executes on button press in save_curve_button.
function save_curve_button_Callback(hObject, eventdata, handles)

global curve

[filename3, pathname3] = uiputfile('*.txt');
path_file = fullfile(pathname3,filename3);
a = fopen(path_file,'wt');
fprintf(a,'%6.3f \n',curve);
fclose('all');


% --- Executes on button press in save_figure_button.
function save_figure_button_Callback(hObject, eventdata, handles)

[userFilename, userFolder] = uiputfile({'*.bmp';'*.jpg';'*.png'},'Save as');
fullFileName = fullfile(userFolder, userFilename);

F = getframe(gcf);
Image = frame2im(F);
imwrite(Image, fullFileName );
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CREATE FUNCTIONS - DO NOT EDIT!
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes during object creation, after setting all properties.
function thickness_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function max_temp_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function material_layer_popup_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function name_1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function thickness_1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function d_1_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function k_1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function cp_1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function name_2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function thickness_2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function d_2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function k_2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function cp_2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function name_3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function thickness_3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function d_3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function k_3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function cp_3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes during object creation, after setting all properties.
function name_4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function thickness_4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function d_4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function k_4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function cp_4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function name_5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
```

```
function thickness_5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function d_5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function k_5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function cp_5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end


% --------------------------------------------------------------------
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```