Development of a three-dimensional thermal analysis tool for sounding rockets

ANDRÉ RYMAN
ANDREAS WAHLBERG

**KTH Industrial Engineering
and Management**

Master of Science Thesis
Stockholm, Sweden 2014

DEVELOPMENT OF A THREE-DIMENSIONAL THERMAL ANALYSIS TOOL FOR SOUNDING ROCKETS

André Ryman

Andreas Wahlberg

Master of Science Thesis MMK 2014:24 MKN 114
KTH Industrial Engineering and Management
Machine Design
SE-100 44 STOCKHOLM

KTH Industriell teknik
och management

Examensarbete MMK 2014:24 MKN 114

Utveckling av ett tredimensionellt termisk beräkningsverktyg för sondraketer

André Ryman
Andreas Wahlberg

Godkänt 2014-06-05	Examinator Ulf Sellgren	Handledare Ulf Sellgren
	Uppdragsgivare Swedish Space Corporation	Kontaktperson Ylva Houltz

Sammanfattning

Detta examensarbete har utförts i samarbete med Swedish Space Corporation på avdelningen Science Services. SSC är ett svenskt företag verksam inom rymdtekniksektorn som erbjuder myndigheter, företag och forskarlag runt om i världen möjlighet att dra nytta av rymden. Avdelningen Science Service är ansvariga för utvecklig samt uppskjutning av sondraketer. I dagsläget finns en god kunskap hur sondraketens experimentmoduler ska konstrueras för att minimera den termiska påverkan i systemen. Dock existerar ingen beräkningsmodell för att undersöka värmeutvecklingen och temperaturer i dessa experimentmoduler, all kunskap inom detta område är baserad på tidigare erfarenheter. Syftet med detta examensarbete var att utveckla en termisk beräkningsmodell som kan användas som underlag när nya experimentmoduler konstrueras på SSC. Användningsområdet för modellen var avsett i ett tidigt skede i produktutvecklingsprocessen, innan CAD-modeller eller dylikt har framställts. Därav efterfrågades en flexibel modell där användaren kan undersöka olika typer av kompententer och konfigurationer.

Den Finita elementmetoden (FEM) har används för att skapa en termisk beräkningsmodell i MATLAB. Utvecklingen delades upp i tre steg, eller tre programversioner, vilket bidrog till att frågeställningens komplexitet reducerades. Första programversion genomfördes för att approximera värmeflöden och temperaturer i tre dimensioner med hjälp av Galerkins viktade residualmetod. I den andra programversionen implementerades den dynamiska omgivningen som uppstår under flygning. Baserat på den yttre påverkan från det dynamiska förloppet delades flygningen in i olika faser, alla med skilda randvillkor. I den slutliga programversionen implementerades intern konvektion, strålning och ett grafiskt användargränssnitt. Samtliga versioner verifierades numerisk med hjälp av COMSOL (2013) .

Resultatet från beräkningsmodellen påvisade att den interna konvektionskoefficient samt konduktiviteten hos element har stor inverkan på hur temperaturen fördelas inuti modulen. Resultaten indikerade även att den yttre miljön inte har en signifikant betydelse för dessa temperaturer. De antaganden som utförts samt förbättringsförslag avhandlades även i detta arbete.

Nyckelord: Värmeledning, tredimensionellt, Finita elementmetoden, Sondraket, Simulering

KTH Industrial Engineering
and Management

Master of Science Thesis MMK 2014:24 MKN 144

Development of a three-dimensional thermal analysis tool for sounding rockets

André Ryman
Andreas Wahlberg

Approved 2014-06-05	Examiner Ulf Sellgren	Supervisor Ulf Sellgren
	Commissioner Swedish Space Corporation	Contact person Ylva Houltz

Abstract

This thesis has been performed in collaboration with the Swedish Space Corporation at the department Science Services. SSC provides services in the areas of spacecraft subsystems, ground stations and sounding rockets to enable governments, companies and research institutes to benefit from space. Science Services are responsible for sounding rocket flight missions allowing customers to perform research in a microgravity environment. Currently, they have good knowledge how to design the sounding rockets experiment modules to minimize thermal effects within the system. However, no computational models are available to evaluate and verify the thermal heat transfer inside of the modules and as such the systems are designed primarily based on previous experience.

The main purpose of this thesis was to develop a thermal computational model, which would work as a basis for designing experiment modules. The model would be used in an early stage of the design process before CAD parts have been designed. This required a flexible model allowing the user to evaluate different types of components and configurations.

A finite element method (FEM) was used to perform heat transfer calculations in MATLAB. The development process was divided into three stages, which reduced the complexity of the problem formulation. The first version was made to approximate heat transfer solution in three-dimensions using the Galerkin's weighed residuals method. The second version was made to implement the dynamic environment occurring during flight missions. Based on the external environment, the dynamic process was divided into phases with different boundary conditions. In the final version internal convection, conductivity between air elements and a GUI was developed. The versions were verified with COMSOL (2013) and previous measured flight data.

The results from the simulations showed that the internal convection coefficient and the element's conductivity have a great impact on how the heat is distributed inside the modules. A low convection will lead to internal temperature peaks, which can cause damage to sensitive experiment equipment. Also, the results indicated that the external environment does not have a significant impact on the internal temperatures. The assumptions made and recommendations are also covered in this thesis.

Keywords: Three-dimensional heat transfer, Finite element method, Sounding rocket, Computational simulation

FOREWORD

In this section, people who have contributed to this Master thesis are acknowledged.

The work presented in this Master thesis was carried out at Swedish Space Corporation (SSC) and Machine Design track at the Royal Institute of Technology (KTH), Stockholm. We are thankful for all the help from our supervisor Professor Ulf Sellgren.

We are also extremely grateful for all the support and guidance given by the employees at SSC. Special thanks are dedicated to Ylva Houltz and Olle Janson for their commitment and contributions in this thesis.

André Ryman & Andreas Wahlberg

Stockholm, May 2014

NOMENCLATURE

In this section Notations and Abbreviations used in the Master Thesis are presented. Further descriptions of symbols are presented in the text.

Notations

Symbol	***Description***
A	Area [m^2]
c_p	Specific heat capacity [J /(°K ·kg)]
E	Young´s modulus [Pa]
F_a	Geometric factor for Earth's albedo [-]
F_{IR}	Geometric factor for incoming radiation [-]
h	Convection coefficient [W/(°K·m^2)]
I_{IR}	Earth's IR heat flux [W/m^2]
I_{sun}	Sun's heat flux [W/m^2]
k_x, k_y, k_z	Thermal conductivity (x,y-and z-direction)[W/(°K·m)]
N_{ij}	Shape function for node ij [-]
q	Heat flux [W/m^2]
r	Radius [m]
T	Temperature [°K]
t	Time [sec]
V	Volume [m^3]
α	Thermal diffusivity [m^2/s]
α_a	Albedo [-]
α_r	Linearized radiation coefficient [W/(°K·m^2)]
ε	Emissivity [-]
ρ	Density [kg/m^3]
σ	Stefan-Boltzmann's constant [W/(°K^4·m^2)]

Matrix and vector notations

$\boldsymbol{K}_{conv}$	Convection matrix [W/(°K·m^2)]
$\boldsymbol{K}_{cond}$	Conduction matrix [W/(°K·m)]
$\boldsymbol{K}_{rad}$	Linearized radiation matrix [W/°K]
$\boldsymbol{q}_{\mathrm{cond}}$	Conduction load vector [W]
$\boldsymbol{q}_{\mathrm{conv}}$	Convection load vector [W]
$\boldsymbol{q}_{ext}$	External heat load vector [W]
$\boldsymbol{q}_{int}$	Internal generated heat vector [W]
$\boldsymbol{q}_{\mathrm{rad}}$	Radiation load vector [W]

Abbreviations

CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
FEM	Finite Element Method
GUI	Graphical User Interface
PDS	Product Design Specification
SSC	Swedish Space Corporation

TABLE OF CONTENTS

1 INTRODUCTION

The first part of this thesis provides a background and description of the problem formulation. The purpose and delimitations of the work are explained together with a description of the method used to solve the task.

1.1 Background

The Swedish Space Corporation provides space services in the areas of spacecraft subsystems, ground stations and sounding rockets to enable governments, companies and research institutes to benefit from space. This thesis has been performed in collaboration with the department SSC Science Services. They are responsible for sounding rockets and balloon flights allowing customers to perform research in microgravity. By using high-altitude sounding rockets, research in a low gravity environment can be performed in a more cost-effective way and within a shorter time frame compared to manned flights. The rockets are launched and controlled from Esrange Space Center based in Kiruna. (SSC Space, 2014)

As of today, SSC has two different sounding rockets in their program, MASER and MAXUS. The MASER program has been active since 1993 and vehicles are launched every 18 months. The launch campaign usually takes place during the winter season when the lakes in the impact area around Kiruna are frozen. This reduces the risk of losing rockets when landing and eases the recovery process. The MASER sounding rocket has the capability to conduct research for 6-8 minutes within a microgravity environment, traveling with an altitude up to 300 km above sea level (SSC Space 2014). The vehicle uses a two-stage solid propellant thruster and a payload. In order to provide flexibility the sounding rocket's payload is modularized, allowing it to carry four independent experiment modules with total mass of 400 kg. (MASER User Manual, 2013)

1.2 Problem Description

The experiment equipment inside the sounding rockets is working in a unique thermal environment. The equipment inside the payload can vary significantly between different experiments. Anything from heating elements, such as furnace or electronic components, to thermally sensitive objects, such as plants or microorganisms can be included. The sounding rocket is exposed to a dynamic thermal environment with significant varying boundary conditions. During countdown prior to launch, the temperature of the external environment can vary from -30° C to 25° C depending on the season and weather. Normally, the temperature inside the rocket is held constant during this phase by external heaters or coolers connected to the vehicle through umbilical cord. During lift-off these cords are disconnected from the rocket and the ability to affect the internal temperature thereby lost. Furthermore, during ascend and descend the vehicles skin is heated by friction from the atmosphere. In microgravity, the heating and cooling is limited due to the lack of natural convection. In some cases, the modules are not pressurized leading to vacuum and a great reduction in heat transfers. During the final phases of flight, parachute landing and recovery, the experimental payload can both be cooled or heated depending on the rockets temperature and the external conditions.

The high demands from the internal temperature together with the rockets temperature and the external environment leads to a very dynamic process in terms of temperature control with a large amount of parameters needs to be taken into consideration for the model. Today, SSC has a good understanding of how to design thermal systems in sounding rockets but this is purely based on previous experience. Therefore, a thermal model is required to help the engineers in an early stage of the design process to gain a general picture of the thermal processes and to understand the needs of internal heating/cooling.

1.3 Purpose

This MSc degree project aims to develop and verify a computational model that can be used to calculate the internal temperatures of the MASER sounding rocket and subsequent the temperature of experiment modules within the payload. The model should offer a satisfying approximation of the inside temperature compared with quantitative data from previous flights. The main aspect of the program is the ability to handle different configuration of equipment and allow the user to adjust the elements of the module to match the intended design. In other words, a flexible module is required where the engineers at SSC can test different cases. Furthermore, the program should be able to simulate the dynamic environment from a MASER mission, including all conditions from pre-launch to recovery back to Esrange. A more detailed description of the deliverables can be found in the requirement specification, seen in Appendix A.

1.4 Delimitations

The model will not take into consideration the internal airflow inside the module since this would require a detailed model of the interior design and much more complex algorithms. There are several CFD-software packages on the market, which can be used in later steps to obtain a more detailed view of the heat transfer inside the modules. In order to develop an appropriate computational model, for this thesis, following assumptions have been made:

- Thermal expansions are neglected. All structures are treated as rigid bodies.
- During exit and re-entry of the atmosphere the skin of module is assumed to experience the same heat flow. The heating is therefore independent of the skin location.
- No thermal bridges between the modules and the rocket skin.
- Heat caused by friction between components inside the module is neglected since it is assumed to be small in comparison to other heat sources.
- In the case of an open module, the ambient pressure in microgravity is assumed to be zero.
- The thermal conductivity of material is assumed to be constant and not dependent on temperature.
- No atmosphere is assumed to exist above 90 km altitude.
- All gases are assumed to ideal.

1.5 Method

Solving and evaluating certain problems in scientific computing are preferable done by dividing processes into three main stages. The first stage is to define the problem and model by selecting a suitable mathematical approach. Secondly, develop an iterative computational model to numerically solve the defined problem. Finally, interpret and present the result in an adequate way. (Pentenrieder, 2005)

Thermal heat transfer simulations are often conducted by finding an approximate solution to the differential equation. This is due to their high complexity, especially in three dimensions, which makes an analytical solution difficult or nearly impossible to obtain. In most cases a numerical approach is more suitable (Bergheau & Fortunier, 2008). The finite element method (FEM) is a numerical method satisfying a suitable approximation for heat transfer problems in any dimension. Finite elements are obtained by dividing the whole domain into sub-domains and by increasing the number of elements a more accurate approximation of the solution can be obtained. In this thesis project, a FEM approach has been be selected to solve the three-dimensional heat transfer problem of simulating thermal effects in sounding rockets. All calculations have been made iteratively and the heat transfer variation is found as a function of a pre-define time step. This is a suitable approach since sounding rockets have a dynamic environment on its boundaries. The direct FEM method (Gauss method) was used to numerically compute solutions since the number of nodes in the system is relative small and does not require iterations. Despite requiring a lot of input data, the direct method is suitable for transient heat simulation and easy to implement. (Bergheau & Fortunier, 2008)

The computational FEM-model was developed with Mathwork's MATLAB R2013a (2013). The advantage of MATLAB is that it can handle large matrix systems in a time efficient way. MATLAB also contains a built-in development environment for graphical user interfaces (GUIDE), which has been used to increase the applications handiness.

An agile methodology was used to develop the computational model. During the planning phase the task was divided into three main versions (see Appendix B) to reduce the program structure's complexity. Also, stage gate meetings at SSC were conducted allowing feedback from engineers to be provided regarding the program structure to ensured that the pre-defined requirements. (Szalvay, 2004). The verification of the MATLAB computational model was made by comparing the results with data from previous flights and simulations made in COMSOL (2013). For early versions of the model the basic functions was isolated and verified, followed by verification of more complex and realistic scenarios. As a final verification, a real flight was represented in the program and the results from the simulations were compared with previous measured data.

2 FRAME OF REFERENCE

This chapter describes the SSC sounding rocket program and the thermal aspects of a flight. Based on thermal conditions the flight is divided into different phases. Thereafter the fundamentals of heat transfer are described together with the mathematical description of heat conduction in three dimensions. At last, a numerical method for solving the presented differential equation is described.

2.1 SSC Sounding Rocket Program

The MASER systems consist of two main categories, the vehicle and ground equipment. The ground equipment includes all additions services needed to perform a successful flight mission such as facilities, service systems and experiment module ground support. A schematic picture of the MASER system and related sub-categories can be seen in Figure 1.

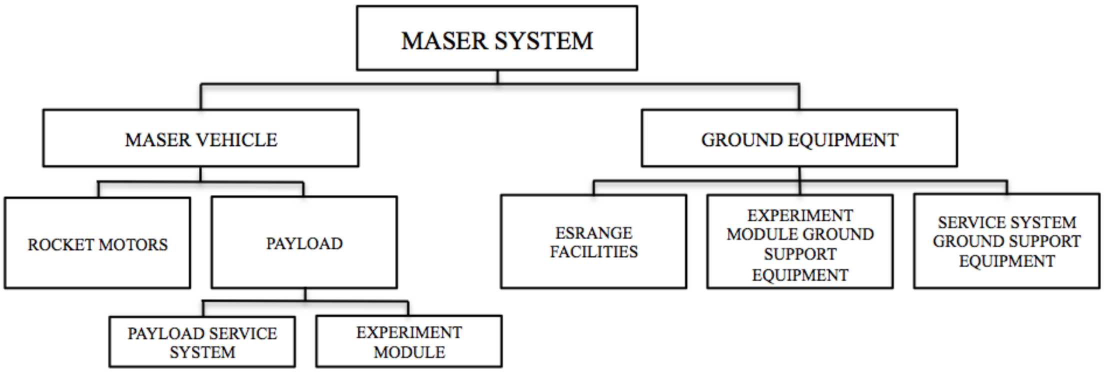
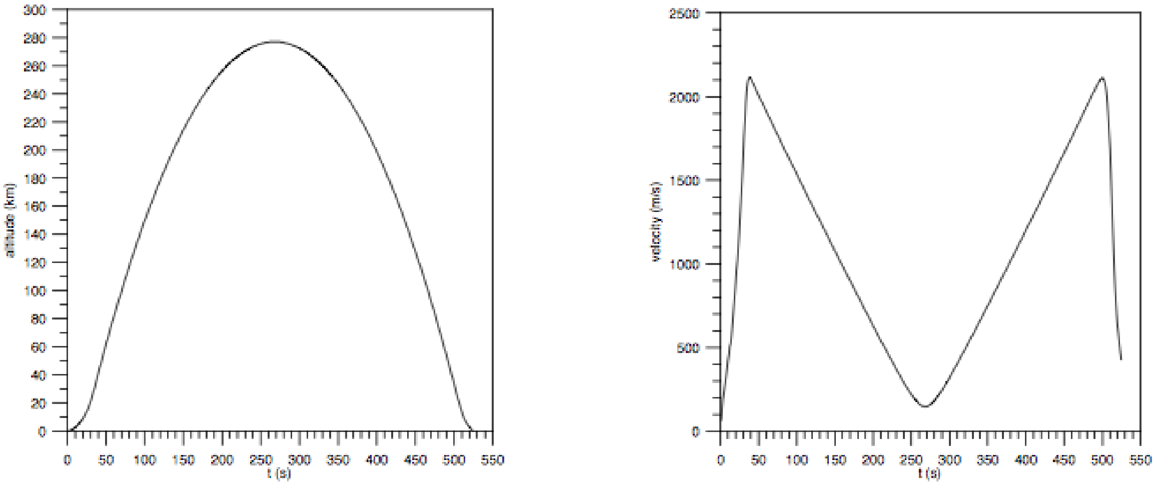


Figure 1. Schematic picture of the MASER system (MASER User Manual, 2013)

The MASER vehicle is a VSB-30 developed by the Brazilian Space Agency. It is equipped with two solid propellant motors, S30 and S31 (Dalle Lucca, 2014). The first stage is active for 15 seconds. Subsequently, the second stage is ignited and active for another 30 sec, until a microgravity environment is reached. The rocket can carry a payload with a mass up to 400 kg and has a length of 13 m (Machado & Pessoa Filho, 2007). A typical velocity and altitude map from a VSB-30 flight can be seen in Figure 2.



a) Altitude map

b) Velocity map

Figure 2. Altitude and velocity map from a VSB-30 flight (Machado & Pessoa Filho, 2007).

The rocket payload is divided into standardized sections, which enables it to carry four custom-designed experiment modules. The standardization of the payload gives the MASER vehicle a large flexibility, simple interface connections and independence between each experiment. The experiment modules are placed inside of the payload structure, as seen in Figure 3. Each module is placed on an experiment deck and connected to the payload housing using rubber dampers to protect the module from launch vibrations. Radax joints are used to connect the payloads to each other. Appendix C shows the drawings and dimensions of the payload's mechanical interface. (MASER User Manual, 2013)

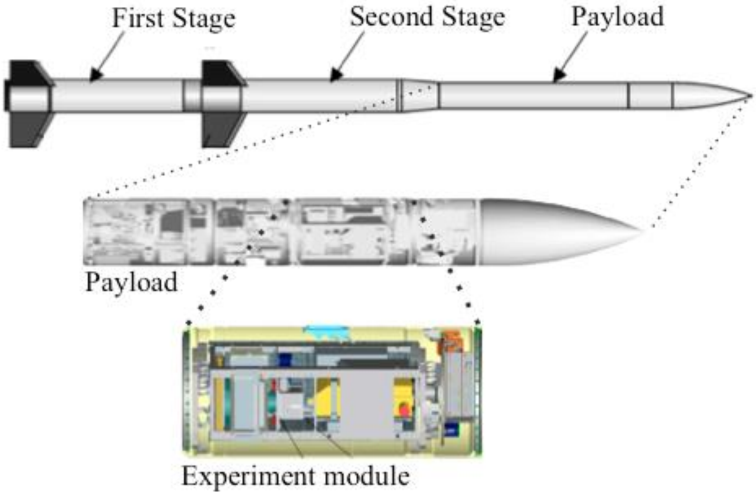


Figure 3. Schematic picture of the MASER vehicle.

2.2 MASER Flight Mission Phases

The flight mission can be divided into five phases; pre-launch, ascent, microgravity, decent and recovery. Figure 4 shows each phase, which has a different thermal effect on the vehicle since the outside environment is changing. In this section, each phase is described in detailed with focus on the thermal conditions.

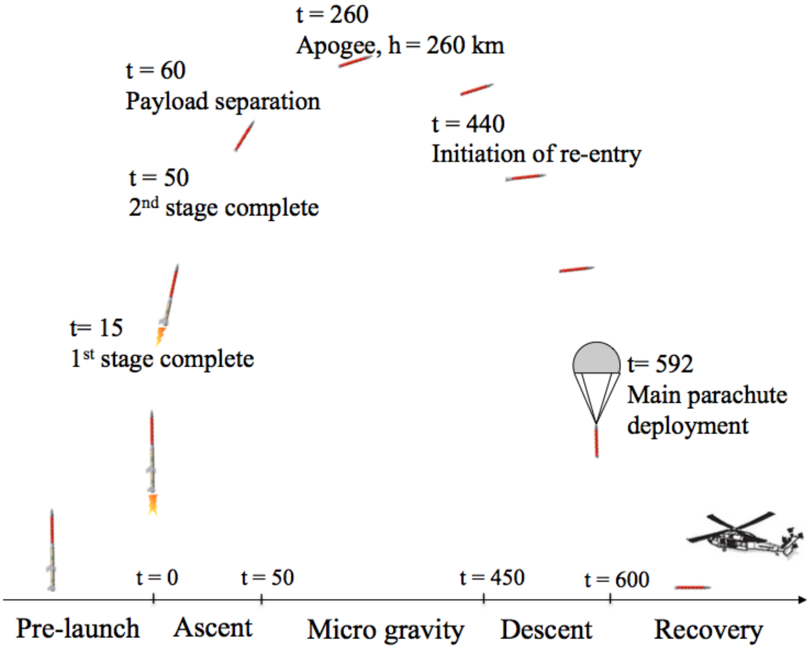


Figure 4. MASER flight phases and events.

The temperature inside and outside of the experiment module is important to consider when performing research in microgravity. The temperature is often measured at the rocket skin, close to crucial components and the actual experiments, to verify that the temperature has not influenced the results or damaged the components (MASER User Manual, 2013). However, the thermal knowledge is collected after completed mission. Figure 5 shows previous flight data of MASER 12's skin temperature. The graph clearly indicates that the maximum skin temperature is reached during atmospheric re-entry.

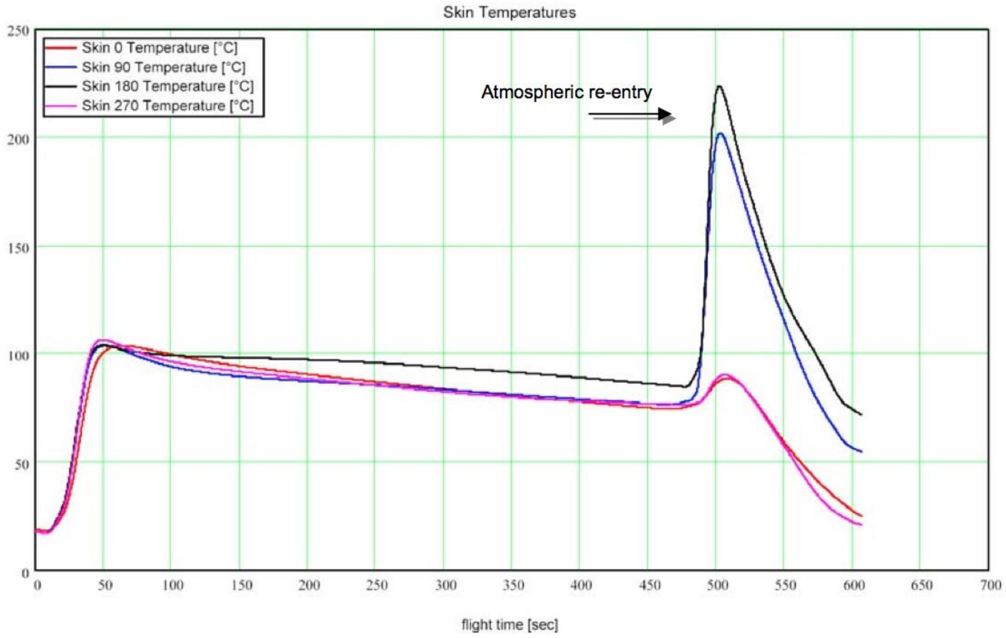


Figure 5. MASER 12 skin temperature during flight. (MASER User Manual, 2013)

2.2.1 Pre-launch

During pre-launch the MASER rocket is placed inside a ventilated launch building where the temperature is kept constant at around 20° C with the help of heaters. This is used to protect the rocket from snow, temperature changes in the environment, radiation from the sun and wind. Due to the low outside temperature the air is circulated within the launch building with fans to avoid large temperature gradients. To allow radio communication between the command center and the rocket, the building has hatches, which further increases the flow of air within the building. The launch building is not insulated since it is important to be able to ventilate the fumes that are produced during lift off. All these factors lead to a high circulation of air inside the launch building giving a high convection coefficient on the rocket skin. (Thorstenson, 2014)

2.2.2 Ascent and Descent

During ascent and descent the rocket is affected by aerodynamical heating as a result from friction with the atmosphere at high velocities. According to measurements by Mazzoni et al. (2005), the VSB-30 rocket can reach speeds up to 1800 m/s while still in an atmospheric environment. The thermal heating is most significant at the nose cone, fins and other leading edges where the vehicle collides with atmospheric particles. At these positions, the air temperature can reach over 1400 °C due to air compression caused by shocks (Mazzoni et al., 2005). The temperature of the rocket payload is however much lower and evenly distributed. Figure 6 shows the temperature distribution plotted along the side of the rocket nose cone and side of the payload. It can be noted that the temperature is constant over the whole side of the payload. The temperature itself depends on surface treatment of the rocket.

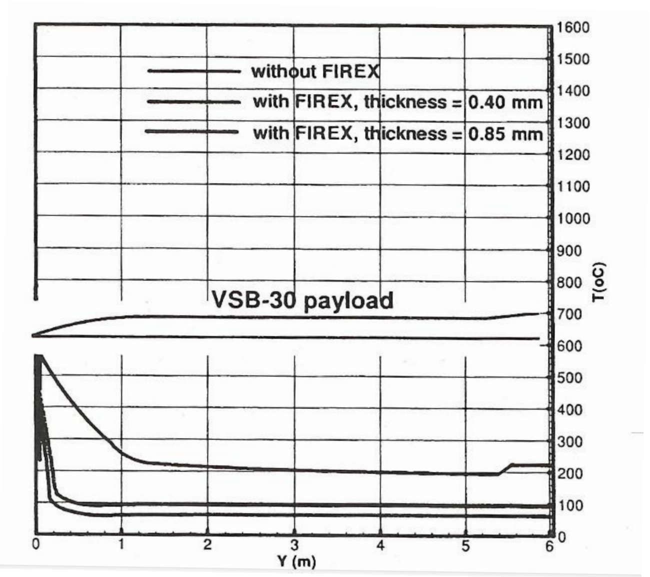


Figure 6. Temperature distribution along the side of the nose cone and payload at 525 sec (Machado & Filho, 2007)

According to Goddard Space Flight Center (2005) the high temperature on the outside surface of the rocket does not generally affect the inside components during the course of a flight. This is of course primarily depending on the interior design and placement of components and can vary widely among different configurations. Internal electrical heating caused by long standby times during pre-launch and long flight missions could be more crucial than the thermal effects caused by the environment. During the ascent most of the inside equipment is either active or on standby, waiting for the microgravity phase to start. In the pre-launch phase connected external heating or cooling systems allows to control the inside temperatures. To reduce energy consumption and unnecessary temperature rises during the decent, the experimental equipment is often turned off.

2.2.3 Microgravity

The microgravity phase starts around 60 seconds after the launch when the rocket has an altitude of approximately 100 km. The g-levels in microgravity are below 10^{-4} g and lasts in average six and a half minutes. In this phase the experiments are performed, many systems starts and usually the experiment equipment increase their power consumption (MASER User Manual, 2013).

Due to the lack of particles in space no convection can take place from the rocket skin. Instead, the cooling and heating of the rocket is purely depended on radiation. Heat arises when absorbing radiation from other sources such as the earth and sun. However, cooling occurs by emitting infrared radiation from the rocket. The absorbing radiation can be categorized into direct sunlight, sunlight reflected off the Earth (albedo) and IR from the Earth (Gilmore, 2002). The direct sunlight is the largest radiation source and due to the Earth’s elliptical orbit around the sun it is depending on season. At summer solstice the distance between the Sun and the Earth is largest resulting in minimum radiation power intensity of 1322 W/m^2. However, the maximum intensity is 1414 W/m^2, which occurs during winter solstice. In average, the power intensity value 1367 W/m^2 can be used when calculating thermal effects from the Sun. The Sun itself has an 11-year solar cycle nevertheless the radiation can be seen as constant since the total emitted radiation only varies by 1%. The reflected sunlight from an object, such as a planet, is called albedo, expressed as a fraction of the total light impacting the object. The albedo can be fairly uncertain due to the variation of planet reflectance. Generally, the albedo from the Earth increases at higher latitude because of the large reflectance from snow and ice. Continents tend to have higher albedo than ocean regions but it can also depend on the cloudiness of the region; more clouds increase the amount of reflected radiation. The albedo affecting the rocket is also depending on the angle between the rocket and incoming sunlight. A large angle from zenith (the sun is directly facing the rocket's nose cone) decreases the energy per square meter. The IR radiation is the sunlight that is not reflected by the earth initially but absorbed and then eventually emitted. Factors such as clouds and local temperatures affect the amount of IR emitted by the earth. Most IR is emitted from desert regions and tropical regions with high temperatures. In contrast to albedo the emitted IR can be seen as decreasing with increasing latitude. The different heat flux terms can be calculated accordingly to Ross (2003) as

$$q_{IR} = I_{IR} \cdot F_{IR} \cdot \varepsilon \cdot \alpha \tag{1}$$

$$q_{albedo} = I_{sun} \cdot F_a \cdot p_a \cdot \alpha \tag{2}$$

$$q_{sun} = I_{sun} \cdot \alpha \tag{3}$$

where I_{IR} denotes the earth infrared flux and I_{sun} is the heat flux from the sun. F_{IR} is a geometric factor for IR based on the average for a flat plate with the surface normal angle of 0° and 30°. The surface emissivity was denoted by ε, and the surface absorbance by α. Earth's albedo is set as p_a, F_a is a geometric factor for albedo that depends on the angle between the sun, the rocket and the centre of the earth as well as the angle between the rocket and horizon.

Another non-significant source of heating is *Charged-Particle Heating*, which occurs when the spacecraft collides with electrons or protons. The charged particle heating only affects the front of the vehicle and only to the depth of a few millimetres in the skin. For orbiting vehicles this affects the outer skin temperature and needs to be taken into consideration when designing the cooling system (Gilmore, 2002). However, due to the short flight time of MASER this source of heating is disregarded in this thesis.

The outside environment also affects the thermal conditions in a more indirect way. The lack of gravity affects the inside of the module since no natural convection can take place. For an open module there will be vacuum instead of air between the components, leading to a non-convective environment. The only mode of heat transfer in this case is radiation. According to Goddard Space Flight Center (2005) there is a risk for local hot spots with an open module due to the reduction of heat transfer. This should be taken into consideration when designing the module by protecting sensitive equipment with thermal bridges and/or insulation (Goddard Space Flight Center, 2005).

2.2.4 Recovery

The recovery phase starts when the rocket has landed. A helicopter collects and transports the sounding rocket back to Esrange. During the recovery phase, the rocket is subjected to natural convection and conduction from the surrounding snow, radiation is assumed to be minimal. Since the experiments have been conducted the experiment equipment inside the module is turned off. This includes deactivation of fans and cooling systems, which can as a result lead to local temperature rises occurring in this phase (Törnqvist, 2012).

2.3 Fundamentals of Heat Transfer

Thermodynamics and heat transfer is of essential importance in the field of engineering. Knowledge of thermal effects can be valuable information in order to optimize products or systems performance. Lewis et al. (2004) has defined heat transfer as the energy transportation between bodies due to temperature differences caused by three different modes; conduction, convection and radiation. In order to derive the equations needed to compute heat transfer solutions, fundamental knowledge within the laws of heat transfer is necessary.

2.3.1 Conduction

Conduction occurs due to temperature difference in a body and is defined as the diffusive transfer of internal energy. The diffusion differs depending on what type of medium and state of aggression. The diffusion is caused by exchange of energy between molecules or because of the motion of present free electrons (Lewis et al., 2004). This can be comprehended by imagining a closed bowl of water; the water temperature will try to reach an equilibrium temperature equal the surrounding temperature by transfer of internal energy. Mathematically, the conduction heat flux can be described with *Fourier's law*. In three-dimensional analysis, the heat flux in W/m^2 can be expressed in different direction as

$$q_x = -k_x \frac{\partial T}{\partial x} \tag{4}$$

$$q_y = -k_y \frac{\partial T}{\partial y} \tag{5}$$

$$q_z = -k_z \frac{\partial T}{\partial z} \tag{6}$$

Note that $k_{x,y,z}$, expressed in W/(m°K), is the thermal conductivity and related to type of medium and state of aggression.

2.3.2 Radiation

All bodies exchange thermal radiation in some sense when the environment temperature differs from the body temperature. The amount of emitted radiation from a surface can be expressed with *Stefan–Boltzmann's law*. However, it is also valuable to know how much energy absorbed by the body. By assuming that the absorptivity is equal to the emissivity (gray surface) the net radiation *from* a surface to its environment can be calculated as

$$q_{rad} = \varepsilon\, \sigma(T^4 - T_0^4) \tag{7}$$

and is expressed in W/m^2. The environment temperature is defined as T_0 and the Stefan-Boltzmann's constant is defined as σ equal to $5.7 \cdot 10^{-8}$ W/m^2°K^4). Note that the emissivity ε is a constant within the interval 0 to 1. For an ideal surface (a black surface) the emissivity is equal to 1. (Bergman et al., 2011) For moderate and low temperatures the radiation can be linearized, Equation (7) can be rewritten as

$$q_{rad} = \alpha_r(T_1 - T_2) \tag{8}$$

where the coefficient α_r is defined as

$$\alpha_r = 4\varepsilon_{12}\sigma T_m^3 \tag{9}$$

and T_m is the mean the temperatures of T_1 and T_2 (Siegel & Howell, 1992). The emissivity is stated as

$$\frac{1}{\varepsilon_{12}} = \frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1 \tag{10}$$

2.3.3 Convection, Free & Forced

Convective heat transfer describes the temperature change caused by the movement of a surrounding gas or liquid. There are two different types of convection, free and forced. Forced convection can be realized by imagine a hair dryer blowing on a body; the air is forced to act on the surface. In contrast to free connection, which occurs due to a density change in the surrounding medium, where often air is naturally moving (Bergman et al., 2011). *Newton's law of cooling* describes the heat flux in W/m^2 through convection as

$$q_{conv} = h\left(T_f - T_s\right) \tag{11}$$

where the surrounding temperature of a fluid or gas is defined as T_f and the surface temperatures are defined as T_s. The convection heat transfer coefficient is denoted as h in W/(°K·m^2) and is dependent on type of medium, surface geometry and fluid motion. For free convection in gases typical values for h are between 2 and 25 W/(m^2°K). For forced convection the coefficient is normally 25-250 W/(°K·m^2), depending on the velocity of the medium. (Bergman et al., 2011)

2.4 Mathematical Description

Heat conduction in a small three-dimensional body can be described by the following partial differential equation

$$-\left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z}\right) + Q = \rho c_p \frac{\partial T}{\partial t} \tag{12}$$

where q_x, q_y and q_z are the components in each direction of the heat flow through the body's surfaces, Q is the inner generated heat and expressed in W/m^3, ρ is the density of material (kg/m^3) and c is the volumetric heat capacity (J/(kg°K)). The temperature is expressed by the term T and the time is denoted by t (Nikishkov, 2010). Combining the above equation with equation (4), (5) and (6) gives the following basic heat transfer equation

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right) + Q = \rho c_p \frac{\partial T}{\partial t} \tag{13}$$

Note that in a steady state case the right side of the equation can be removed since there is no change in temperature with respect to time.

2.4.1 Boundary -and Initial Conditions

In order to solve partial differential equations appropriate boundary conditions and initial conditions are required. For the heat transfer equation above, the boundary conditions can be of two types namely Dirichelt or Neumann, these two can be used separately or combined (Blomberg, 1996). The Dirichlet's condition, also called boundary conditions of the first type, prescribes a temperature on the boundary of the body and can be stated as

$$T = T_b \; on \;\; \Gamma_T \tag{14}$$

where Γ_T is the boundary surface. Implementation of this type of boundary condition is often easy since it can be assumed to be a scalar. The Neumann boundary condition, called boundary condition of the second type, prescribes a heat flow out of the body on a given boundary and can be mathematically expressed as

$$q(t) = -k\frac{\partial T}{\partial n} = C \; on \; \Gamma_q \tag{15}$$

Here, n is the outwards normal to the surface, C is the given flux and Γ_q is the boundary. In the case of perfect insulation the Neumann boundary condition is set to zero, meaning that there is no heat flow over the boundary. Heat transfer through convection over a surface falls in this category and can be expressed with the help of Neumann formulation as (Lewis et al., 2004)

$$-k\frac{\partial T}{\partial n} = h(T_w - T_a) \tag{16}$$

To be able to implement boundary conditions of the second type it is necessary to take the possible anisotropy of the material into consideration. Equation (15) can be rewritten as

$$k_x\frac{\partial T}{\partial x}l + k_y\frac{\partial T}{\partial y}m + k_z\frac{\partial T}{\partial z}n = C \; on \; \Gamma_q \tag{17}$$

and (16) as

$$k_x\frac{\partial T}{\partial x}l + k_y\frac{\partial T}{\partial y}m + k_z\frac{\partial T}{\partial z}n = h(T_w - T_a) \tag{18}$$

where *l*, *m* and *n* are the cosine directions of the outwards surface normal.

In contrast to the heat conduction in equation (13), which contains second order derivatives and therefore requires two boundary conditions, the time term, *t* only consist of a first order derivate and therefore only require that all the temperature in the body is known at a certain time (Lewis et al., 2004). The initial temperature can be denoted as

$$T(x, y, z, t_{start}) = T_i \; over \; the \; whole \; body \tag{19}$$

It is worth to mention that in the steady state-cases the initial temperature is irrelevant for the solution. (Blomberg, 1996)

2.5 Numerical Method

The finite element method (FEM) is a numerical technique that can be used to solve a large number of engineering problems. It offers approximations with good accuracy to differential problems where the exact solutions are complex and difficult to obtain. In some cases the exact solution can be derived from the differential equation but hard to apply in reality, often due to complex geometry of a feature (Hutton, 2004). The wide range of applications together with the high accuracy of the results has made the finite element method popular. Since it has received a lot of attention, several different methods have been developed over the years. The most common methods are finite difference, finite volume and finite element method. They all have different advantages and disadvantages (Lewis et. al, 2004).

The idea behind the finite element method is that a body is considered to be continuous, i.e. it consist of an infinite number of points and thereby an infinite number of degree of freedoms. In order to solve this continuum, an infinite number of unknowns need to be determined, which is impossible (Hsu, 1986). The finite element method divides the solution regions into several smaller interconnected elements (or sub-domains). By doing so it “breaks” the original region’s continuum into elements that are connected at their edges by nodes. The continuum's infinite numbers of unknowns are reduced to a number of nodes. The nodes are points where the differential equation is calculated explicitly and the elements offer piece-wise approximation of the governing differential equation. The shape of elements can vary in order to give a good representation of the original area. Lewis et al (pp. 39, 2004) divides the finite element method into the following six steps.

1. Discretize the continuum into elements that are non-overlapping and match the region. The numbers of nodes are depending on the selected type of element.

2. Select interpolation or shape functions. The shape function is highly linked to the selected element type and represents how the variation of the region changes within the element based on the element's nodes.

3. Form element equations (Formulation). Form equations that represent the properties of each element such as load vectors and stiffness matrices.

4. Assemble the element equations to obtain a system of simultaneous equations.

5. Solve the system of equations

6. Calculate the secondary quantities, such as reaction forces or in this case, reaction heat flows.

2.5.1 Shape Functions

From the finite element method the temperatures for the body are solved explicitly at the nodes. In order to obtain an approximation of the temperature between the nodes so-called shape Functions or interpolation functions needs to be used. These functions use the temperature at the nodes to interpolate the value in any point within an element. The temperature inside a three-dimensional element can be expressed as

$$T(x, y, z) = \sum_{j=1}^{n} N_j T_j \tag{20}$$

where N_j is the shape function and T_j is the temperature associated with node i of the element. The value of n matches the number of nodes belonging to an element. The form of the shape function is dependent of the chosen element, which means that different elements have different shape functions and that shape function can be both linear and non-linear. The temperature $T(x,y,z)$ and thereby the shape functions must fulfill the following demands to be suitable for FEM (Faleskog, 2014),

- It must be continuous over element boundaries
- Completeness, meaning that the function and its derivatives up to the highest order that appear in Equation (13) are assumed to be constant and separated from zero.

The sum of all shape functions is always equal to one (=1) inside the element. When evaluating the temperature at one node the associated shape function is one (unity) and the rest of the Shape Functions are equal to zero. The shape functions can be seen as functions that decide how much influence each node should have in a certain point inside the sub-domain. When moving closer to node j, its shape function will produce a larger value increasing the effect of T_j. The shape functions are dimensionless and usually from -1 to 1 or 0 to 1, however other values can be used since it is depended on the element formulation.

Equation (20) can be rewritten as a matrix product

$$T = [N_1\ N_2 \dots N_n]\begin{Bmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{Bmatrix} = [\boldsymbol{N}]\{\boldsymbol{T}\} \tag{21}$$

The vector ***T*** in the equation is time depended but is independent of coordinates and can therefore be considered to be a constant when calculating the gradient of the temperature. The vector ***N*** on the other hand is depending on the coordinates but not on time for a rigid element. This results in that the temperature gradient can be expressed as

$$\nabla T = \begin{Bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \dots \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial x} & \dots \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial x} & \dots \end{bmatrix}\{\boldsymbol{T}\} = [\boldsymbol{B}]\{\boldsymbol{T}\} \tag{22}$$

where ***B*** is a matrix with the temperature gradient interpolations functions. (Nikishkov, 2010)

2.5.2 The Galerkin Method

With the basics of shape functions explained, the Galerkin method to approximate equation (13) can be described. Since the purpose of this thesis is not to explain the Galerkin method in detail, only the crucial steps in the approximation are explained. For a more detailed explanation of the Galerkin method the work by Nellis & Klein (2008) are recommended.

With an approximate solution for *T* applied to equation (13) the right side of the equation will be a residual separated from zero (Nellis & Klein, 2008).

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z\frac{\partial T}{\partial z}\right) + Q - \rho c\frac{\partial T}{\partial t} = residual \tag{23}$$

The finite element method does not aim at minimizing the residuals; instead the purpose is to minimize the weighted average residual over the whole domain. In order to do so the equation is multiplied with a weight function, *w(x,y,z)* and thereafter integrated over the domain. This gives the following expression

$$\iiint \left(w(x,y,z)\left[\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z\frac{\partial T}{\partial z}\right) + Q - \rho c\frac{\partial T}{\partial t}\right]\right) dxdydz = weighted\ \ average\, residual \tag{24}$$

The weight function, *w(x,y,z)* is assumed to be on the same form as *T(x,y,z)* in equation (20) and depending on the shape function. The weight function and its derivatives can be rewritten as

$$w(x,y,x) = \sum_{i=1}^{n} N_i\beta_i = [\boldsymbol{N}]\{\boldsymbol{\beta}\} \tag{25}$$

and

$$\nabla w(x,y,x) = \sum_{i=1}^{n} B_i \beta_i = [\boldsymbol{B}]\{\boldsymbol{\beta}\} \tag{26}$$

where β_i is a non-zero arbitrary constant. By implementing equation (25) within (24) and dividing with β_i, the following equation is obtained

$$\int_V \left(N_i \left[\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right) + Q - \rho c_p \frac{\partial T}{\partial t} \right] \right) dV = 0 \tag{27}$$

According to Lewis et al. (2004, p. 153), integration by part of the three first terms results in the equation

$$\begin{aligned} -\int_V \left[k_x \frac{\partial N_i}{\partial x}\frac{\partial T}{\partial x} + k_y \frac{\partial N_i}{\partial y}\frac{\partial T}{\partial y} + k_z \frac{\partial N_i}{\partial z}\frac{\partial T}{\partial z} + N_i Q - N_i \rho c_p \frac{\partial T}{\partial t} \right] dV \\ + \int_{\Gamma_q} k_x \frac{\partial T}{\partial x} l d\Gamma_q + \int_{\Gamma_q} k_y \frac{\partial T}{\partial y} m d\Gamma_q + \int_{\Gamma_q} k_z \frac{\partial T}{\partial z} n d\Gamma_q = 0 \end{aligned} \tag{28}$$

where *l*, *m* and *n* are the cosine directors of the outwards surface normal and Γ_q are at the surface of the domain. By replacing the three last terms in the equation with the boundary conditions from equation (15) and (16) along with rewriting the temperature on the form of equation (20) the following expression are obtained

$$\begin{aligned} &\int_V \left[k_x \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + k_y \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + k_z \frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z} \right] T_j dV \\ &+ \int_V N_i \rho c_p \frac{\partial N_j}{\partial t} T_j dV = \int_V N_i Q - \int_{\Gamma_q} N_i h(T - T_a) d\Gamma_q \\ &- \int_{\Gamma_q} N_i q d\Gamma_q \end{aligned} \tag{29}$$

This can be rewritten as the matrix product

$$[\mathbf{C}]\left\{\frac{\partial \mathbf{T}}{\partial t}\right\} + [\mathbf{K}]\{\mathbf{T}\} = \{\mathbf{q}\} \tag{30}$$

with the following left side terms

$$[\mathbf{C}] = \int_V \rho c_p [\mathbf{N}]^T [\mathbf{N}] dV \tag{31}$$

and

$$[\mathbf{K}] = \int_V [\mathbf{B}]^T [\mathbf{D}][\mathbf{B}] dV + \int_\Gamma h[\mathbf{N}]^T [\mathbf{N}] d\Gamma \tag{32}$$

where

$$[\mathbf{D}] = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \tag{33}$$

By assuming that the thermal conductivity is independent of the temperature the equation remains linear which simplifies the solution. If k_x, k_y and k_z would be functions of T, the equation would be non-linear and require an iterative solution. The vector on the right hand side of the equation contains the following terms

$$\{\mathbf{q}\} = \int_V Q[\mathbf{N}]^T dV - \int_\Gamma q[\mathbf{N}]^T d\Gamma_q + \int_\Gamma hT_a[\mathbf{N}]^T d\Gamma_q \tag{34}$$

If the domain has a boundary exposed to radiation, the following terms must be added to the equation

$$[\mathbf{K}_{rad}]\{\mathbf{T}\} = \int_\Gamma \sigma\varepsilon T^4 [\mathbf{N}]^T \, d\Gamma \tag{35}$$

$$\{\mathbf{q}_{rad}\} = \int_\Gamma \alpha q_r [\mathbf{N}]^T d\Gamma_q \tag{36}$$

By adding a radiation term the equation becomes non-linear and the equation requires an iterative solution. (Nikishkov, 2010)

2.5.3 Isoparametric Elements

Many FEM problems have complex geometries and curved boundaries. This can be challenging to represent with elements since it requires either a large number of element and/or curved elements. A common approach to these situations is to use isoparametric elements. The main advantages of isoparametric mapping are that the number of required elements as well as the complexity of the shape functions can be reduced. This is especially useful for 3D-cases where there are several variables and the reduction of these are crucial for the computing time and allocated memory. (Lewis et al., 2004)

An isoparametric element can be seen as a reference element in a reference coordinate system, often referred as local coordinate system. In this coordinate system the geometry of the element is much simpler than the original where curved lines can be represented as straight. The mapping from the global coordinate system into the local coordinates is made by a geometrical transformation based on the nodes position and the shape function of the element (Bergheau & Fortunier, 2008). A graphical representation of the geometrical transformation into isoparametric coordinates can be seen in Figure 7.

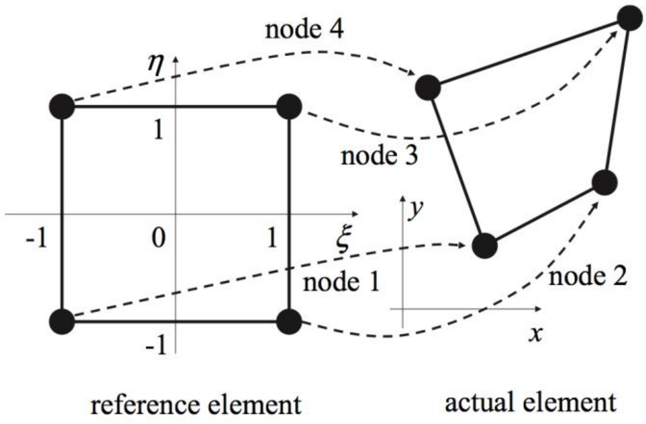


Figure 7. Transformation into linear quadrangle element (Bergheau & Fortunier, 2008, p.83).

The geometrical transformation can be expressed by using the shape functions of the elements. In two dimensions, a position in inside an element can be written as

$$x(\xi,\eta) = \sum_{i=1}^{4} N_i(\xi,\eta)x_i \tag{37}$$

$$y(\xi,\eta) = \sum_{i=1}^{4} N_i(\xi,\eta)y_i \tag{38}$$

where N_i are the shape functions of the nodes expressed in the isoparametric coordinates ξ and η. The global coordinates of node *i* are represented by x_i and y_i.

In the numerical approximation of equation (29) the derivatives of the shape function with respect to global coordinates are required to express the conduction stiffness. Applying the chain rule to a shape function on one of the local coordinates gives the following expression

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \xi} \tag{39}$$

The equation can be rewritten as a matrix product

$$\frac{\partial N_i}{\partial \xi} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = [\,\mathbf{J}\,] \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \tag{40}$$

where ***J***, which contains the first order partial derivation, is the *Jacobian matrix*. For a three-dimensional case, using equation (40), the Jacobian matrix can be written as

$$[\,\mathbf{J}\,] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial \xi} y_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial \xi} z_i \\ \sum_{i=1}^{n} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial \eta} y_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial \eta} z_i \\ \sum_{i=1}^{n} \frac{\partial N_i}{\partial \zeta} x_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial \zeta} y_i & \sum_{i=1}^{n} \frac{\partial N_i}{\partial \zeta} z_i \end{bmatrix} \tag{41}$$

Rearranging the terms and applying all the local coordinates for three dimensions, the derivatives of the shape functions with respect to the global coordinates can be formulated as

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = [\,\mathbf{J}\,]^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} \tag{42}$$

The determinant of the Jacobian plays an important role since it is used for changing variables when integrating over a volume. The transformation from global to local coordinates takes the following form for an integral over three dimensions

$$\iiint dxdydx = \iiint |\,\mathbf{J}\,| d\xi d\eta d\zeta \tag{43}$$

2.5.4 Direct Assembly Procedure

The assembly procedure is an important step in order to perform numerical FEM calculations (Liu & Quek, 2013). In this thesis a direct assemble procedure has been used. By adding up the contributed entities from connected local elements a global system can be created. This method can be realized by considering a one-dimensional rod with two connected elements, each consisting of two nodes, see Figure 8.

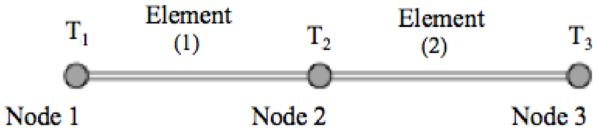


Figure 8. Rod with two elements.

Equation (30) in steady state (*dT/dt=0*) can be expanded and described in the local domain as,

$$\begin{bmatrix} K_{11}^{(e)} & K_{12}^{(e)} \\ K_{21}^{(e)} & K_{22}^{(e)} \end{bmatrix} \begin{Bmatrix} T_1^{(e)} \\ T_2^{(e)} \end{Bmatrix} = \begin{Bmatrix} q_1^{(e)} \\ q_2^{(e)} \end{Bmatrix} \tag{44}$$

where *(e)* indicates element number. The heat flow for the first element can be written as

$$q_1^{(1)} = k_{11}^{(1)} T_1 + k_{12}^{(1)} T_2 \tag{45}$$

$$q_2^{(1)} = k_{21}^{(1)} T_1 + k_{22}^{(1)} T_2 \tag{46}$$

For the second element it can be written as

$$q_1^{(2)} = k_{11}^{(2)} T_2 + k_{12}^{(2)} T_3 \tag{47}$$

$$q_2^{(2)} = k_{21}^{(2)} T_2 + k_{22}^{(2)} T_3 \tag{48}$$

By summing up all contributions from connecting local elements to a node, the equation for the assembled system can be written in global matrix form as

$$\begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & 0 \\ k_{21}^{(1)} & k_{22}^{(1)} + k_{11}^{(2)} & k_{12}^{(2)} \\ 0 & k_{21}^{(2)} & k_{22}^{(2)} \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \end{Bmatrix} = \begin{Bmatrix} q_1^{(1)} \\ q_2^{(1)} + q_1^{(2)} \\ q_2^{(2)} \end{Bmatrix} \tag{49}$$

2.5.5 Reduction Procedure

When applying boundary conductions of the first order, the system matrix equation needs to be restructured to separate known and unknown entities (Liu & Quek, 2013). A thermal boundary in FEM can be seen as a pre-described load in the system. Consider that the temperature in the first node, in Figure 8, is known and 20 °C. Equation (49) can be written as

$$\begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & 0 \\ k_{21}^{(1)} & k_{22}^{(1)} + k_{11}^{(2)} & k_{12}^{(2)} \\ 0 & k_{21}^{(2)} & k_{22}^{(2)} \end{bmatrix} \begin{Bmatrix} 20 \\ T_2 \\ T_3 \end{Bmatrix} = \begin{Bmatrix} q_1^{(1)} \\ q_2^{(1)} + q_1^{(2)} \\ q_2^{(2)} \end{Bmatrix} \tag{50}$$

To perform numerical calculations, the system needs to be restructured as seen in Figure 9.

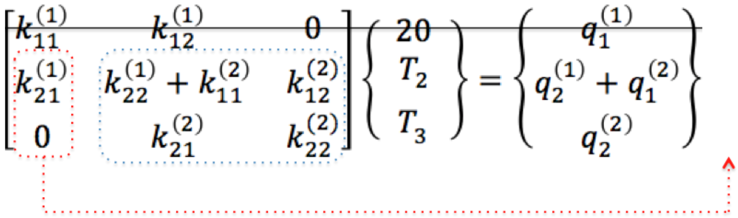


Figure 9. Matrix reduction, the blue dotted box shows the new stiffness matrix.

The reduced system can be now rewritten as

$$\begin{bmatrix} k_{22}^{(2)} + k_{11}^{(2)} & k_{12}^{(2)} \\ k_{21}^{(2)} & k_{22}^{(2)} \end{bmatrix} \begin{Bmatrix} T_2 \\ T_3 \end{Bmatrix} = \begin{Bmatrix} q_2^{(1)} + q_1^{(2)} \\ q_2^{(2)} \end{Bmatrix} - 20 \cdot \begin{Bmatrix} k_{21}^{(1)} \\ 0 \end{Bmatrix}. \tag{51}$$

2.5.6 Gauss Integration Rule

To evaluate the integrals of the element stiffness matrix, ***K***, and the load vector, ***q***, a numerical method called Gauss integration rule or Gauss quadrature is often used. This method has a high accuracy and is therefore one of the most common methods used in FEM (Nikishkov, 2010). The idea behind Gauss quadrature is that the function is evaluated in certain coordinates, so called gauss points. Subsequently, they are multiplied with a weight connected to the value of the gauss point and the number of gauss points. The different points are then summarized and from that a numerical approximation of the integral is obtained. The Gauss quadrature uses *r* weights and *r* points (*2r* constants) and can therefore solve polynomials of the *2r-1* order exactly (Nikishkov, 2010). In three dimensions the gauss quadrature is written as

$$I = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)d\xi d\eta d\zeta = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} f\left(\xi_i,\eta_j,\zeta_k\right) w_i w_j w_k \tag{52}$$

where ξ_i, η_i and ζ_i are points in the different coordinates and w_i ,w_j and w_k are weights. In this thesis, three gauss points for each direction was used leading to 27 different terms to be summed up and an ability to solve polynomials of order five in each direction. The gauss points used and their weight can be seen in Table 1.

Table 1. Gauss points and their weights.

Point, i	ξ_i	w_i
1	$-\sqrt{(3/5)}$	5/9
2	0	8/9
3	$\sqrt{(3/5)}$	5/9

2.5.7 Time Discretization

The dynamic environment of the MASER trajectory makes steady-state calculation insufficient since this neither shows when steady state is achieved nor takes the changing boundary conditions into consideration. Instead, the analysis of the heat transfer in the module must be time dependent to acquire a realistic picture of the temperatures. The finite difference method is based on the assumption that the derivate of a function can be expressed as the difference between the functions value in two different points and divided by the time step (Hutton, 2004). By setting the temperature *T* as a function of time (*t)* the finite difference method can be used to solve transient heat transfer. The temperature at *n+1* can be expressed with the help of a Taylor series as

$$T^{n+1} = T^n + \Delta t \frac{\partial T^n}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 T^n}{\partial t^2} + \cdots \tag{53}$$

Assuming that the second order term and all higher order terms are very small or equal to zero, the equation can be written as

$$\frac{\partial T^n}{\partial t} \approx \frac{T^{n+1} - T^n}{\Delta t} \tag{54}$$

and substitute into equation (30) becomes

$$[\mathbf{C}]\frac{\{\mathbf{T}^{n+1} - \mathbf{T}^n\}}{\Delta t} + [\mathbf{K}]\{\mathbf{T}\}^n = \{\mathbf{q}\}^n \tag{55}$$

With known nodal temperatures at n the equation can be solved for $n+1$ by rearranging the terms and multiplying with the inverse of the $[\mathbf{C}]$-matrix. For heat equations the Crank-Nicholson method is often used (Lewis et al., 2004). In order to apply this method to equation (55) the parameter is θ introduced as

$$T^{n+\theta} = \theta T^{n+1} + (1-\theta)T^{n} \tag{56}$$

By implement $T^{n+\theta}$ into equation(55), the equation can be expressed as

$$[\mathbf{C}]\frac{\{\mathbf{T}^{n+1} - \mathbf{T}^{n}\}}{\Delta t} + [\mathbf{K}]\{\mathbf{T}\}^{n+\theta} = \{\mathbf{q}\}^{n+\theta} \tag{57}$$

and rewritten as

$$\begin{aligned}([\mathbf{C}] + \theta\Delta t[\mathbf{K}])\{\mathbf{T}\}^{n+1} \\ = ([\mathbf{C}] - (1-\theta)\Delta t[\mathbf{K}])\{\mathbf{T}\}^{n} + \Delta t(\theta\{\mathbf{q}\}^{n+1} - (1-\theta)\{\mathbf{q}\}^{n})\end{aligned} \tag{58}$$

In the Crank-Nicholson method the parameter θ is set to 0.5 giving the following equation to be solved (Lewis et al., 2004)

$$([\mathbf{C}] + 0.5\Delta t[\mathbf{K}])\{\mathbf{T}\}^{n+1} = ([\mathbf{C}] - 0.5\Delta t[\mathbf{K}])\{\mathbf{T}\}^{n} + \Delta t\{\mathbf{q}\} \tag{59}$$

The nodal temperatures can now be solved for the time $n+1$ from the equation above by rearranging the terms. However, it requires initial nodal temperatures and that the load vector $\boldsymbol{q}$ is known at the time step n.

2.5.8 Stability

The Crank-Nicolson scheme is a semi-implicit method meaning that it is a marginally stable scheme (Lewis et al., 2004). At larger time steps the Crank-Nicolson still converges but with an oscillating character. According to Holman (2010) the relation between time step and increments in space can be described as

$$\frac{(\Delta x)^2}{\alpha\Delta\tau} = \begin{cases} M \geq 2 & for\ one - dimensional\ system \\ M \geq 4 & for\ two - dimensional\ system \end{cases} \tag{60}$$

where Δt is the time step and Δx is the increment in space. The equation is based on the assumption that the partial derivatives can be approximated as in equation (54), the temperature in one node depends on the temperatures in the adjoining nodes and the increments in space are equal in all directions. The coefficient α is the thermal diffusivity, which can be expressed as

$$\alpha = \frac{k}{\rho c_p} \tag{61}$$

it is a relation between the material's ability to conduct and store thermal energy. Equation (60) shows that a small time step is required for a small increment in space in order to fulfill the requirement $M \geq 2$. If this condition is not achieved, the system will break the second law of thermodynamics (the entropy of an isolated system can never decrease). (Holman, 2010)

3 DEVELOPMENT AND VERIFICATION

In this chapter the working process is described. The chapter is divided into four sections representing each version of the developed program and a simulation section. In the first section, the implementation of the Finite Element Method to a simple geometry is presented. Subsequently, the generation of elements and how the dynamic environment was implemented to the model is presented. In the third part, additional functions such as internal radiation and required external power are described. Each section ends with a verification of the implemented functions. Finally, simulations were performed.

3.1 Version I

The purpose of the first version was to implement the element matrices and perform element assembly properly. The model for the first version consisted of three 27-node elements, all with the same attributes, see Figure 10. By keeping the model relatively small the verification and troubleshooting process was simplified. A more detailed version of the first model with node numbering can be seen in Appendix C.

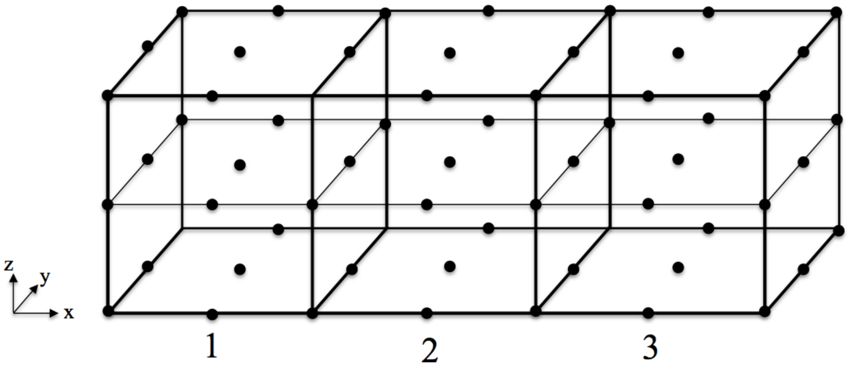


Figure 10. Program version I and its elements.

The program structure of the first version can be seen in Figure 11. In the first stage, the system coordinates and nodes are generated thus allowing elements to be created. Each element goes through a calculation-script where conductivity, radiation and convection are created. The global stiffness matrices and load vectors are thereafter assembled before the calculations are done, solving the temperature for each time step. Thereafter the post processing takes place, where reaction heat fluxes are determined in prescribed nodes and temperatures for the nodes are plotted.

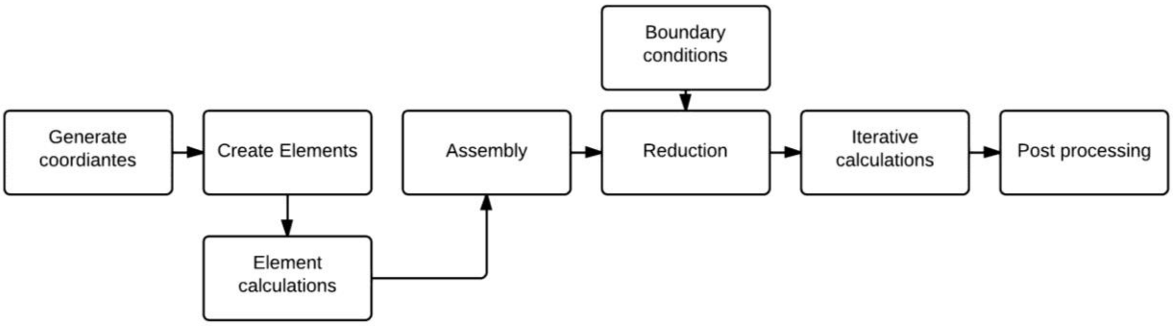


Figure 11. Flow chart for version I.

3.1.1 Elements

The shape of the elements where chosen to be hexahedral or “brick” elements and not tetrahedral, which can be seen in most commercial FEM-programs. Brick elements are often used when the mesh is made by hand while the tetrahedral dominate in 3D since not all geometries can be build up by bricks. The element type that has the best accuracy is related to the geometry of the domain and the type of performed analysis (Cifuentes & Kalbag, 1992). The used type of element can be seen in Figure 12 together with its local node numbering. The used shape functions were based on the work by Delmas (2013) and can be seen in Appendix E.

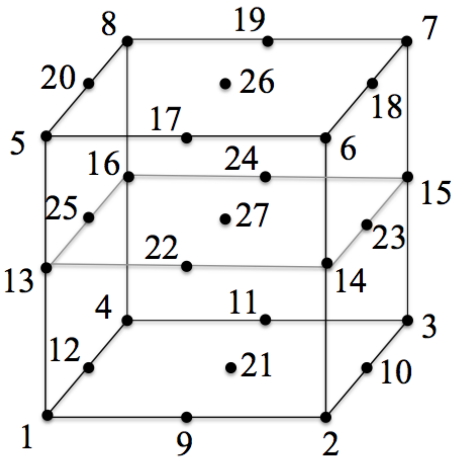


Figure 12. Element with local node numbering.

The conduction stiffness and internal generated heat was calculated for each element by first mapping the element into an isoparametric element with a Jacobian matrix as described in *Chapter 2.5.3*. Thereafter, equation (32) and (34) were numerically integrated using Gauss quadrature with three points in each direction. Element attributes such as conductivity, density and specific heat were stored in vectors during the generation of elements. The stored values position represented the global element number and was selected when working with a specific element.

The minimum size of an element was calculated from equation (60) as

$$\Delta x = \sqrt{\alpha \Delta t M} \tag{62}$$

Using a thermal diffusivity for aluminum of 8,418·10^{-5} m^2/s, a time step of one (=1) second and *M*-value of 8 (=2^3) the minimum element size was calculated to 25 mm. The time step and minimum element size was set as a default for later versions.

3.1.2 Local and Global Coordinates

The next step after specifying the element shape was to determine which and how the generated global nodes represented each element. This was solved by creating a logical element matrix (***E***), each row represented an element and each column represented a node in local coordinates. The matrix was the link between local coordinates, elements and global coordinates and was therefore a critical part of the program. For a 27-node element the element matrix had the following form

$$\begin{array}{cccccc} & Local\ node\ 1 & Local\ node\ 2 & \dots & Local\ node\ 27 & \boldsymbol{Element\ \#} \\ & n_{gi} & n_{gj} & \cdots & n_{gk} & 1 \\ [\mathbf{E}] = & \vdots & \vdots & \vdots & \vdots & \vdots \\ & n_{gl} & n_{gi} & \cdots & n_{gj} & n \end{array} \tag{63}$$

where n_{gj} represented the global number of node i. By knowing the global node number at a specific place in the matrix the local node number and element could be extracted. This made it possible to connect element properties, such as conductivity and density, to nodes and link nodes to surfaces. The element matrix also constituted the base for assembling the contribution from each to the global stiffness matrices and transforming local load vectors into global.

For *version I* the element-matrix was partly done by hand since the system only consisted of three elements and it would have been unnecessary to develop an algorithm for automatic generation. This was changed in later versions when the geometry was determined.

3.1.3 Implementation of Boundary Conditions

To implement boundary conditions of the first type (prescribed T_j), no additional calculations were required. The solution was simply obtained by reducing the rows related to the prescribed temperature, as descried in *Chapter 2.5.5*. Boundary conditions of the second order on the other hand, such as convection or prescribed heat flow, required more complex calculations since they needed to be integrated over the surface as

$$\mathbf{K}_{conv} = \int_{\Gamma} h[\mathbf{N}]^T[\mathbf{N}]d\Gamma \tag{64}$$

and

$$\mathbf{q}_{conv} = \int_{\Gamma} hT_a[\mathbf{N}]^T d\Gamma_q \tag{65}$$

In the same way as for the conduction stiffness and internal generated heat, the element was first mapped into an isoparametric element before the integration. The numbers of gauss-points were reduced from 27 (=3^3) to 9 by fixating one of the coordinates to -1 or 1 depending on which side the convection occurs. Since equation (64) and (65) were evaluated over a surface and not a volume, the Jacobian had to be recalculated in 2D. The determinate of the *Jacobian* was found by taking the cross product of the partial derivatives of the surface and thereafter sums the terms as

$$\left|\mathbf{J}_{surf}\right| = \sqrt{x^2 + y^2 + z^2} \tag{66}$$

From this, the convection stiffness and load could be determined for each element. To determine which element was exposed to convection, a logical convection matrix was created. Each element was represented with a row and the different element surfaces were expressed as columns. A one (=1) in a specific location in the logical convection matrix indicated that a surface was exposed to convection. The logical convection matrix is showed below and can be read as *element 1* has convection on its north side.

$$\mathbf{CONV} = \begin{array}{cc} \begin{array}{cccccc} North & South & East & West & Top & Bottom \end{array} & \boldsymbol{Element}\ \# \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} & \downarrow \end{array} \tag{67}$$

This method was used in the later versions of the program for both the internal convection and the environmental radiation in microgravity. By using the element density vector and logical operators the generation of ***CONV*** was made automatic in *version II*. Notice that in equation (65) the surrounding temperature T_a is not dependent on the inside temperature nor time. Therefore, the equation is only applicable when the convection occurs on the outside of the module and not between the different elements on the inside. How this was solved is presented in *version II*.

3.1.4 Verification Version I

An important part of the work associated with *version I* was to verify that the implemented FEM formulation gave an accurate picture of the heat transfer and temperatures. Therefore, three scenarios were set up and simulated; one for convection, one for radiation and one for the internal generated heat. By simulating each separately the accuracy could clearly be identified. Each scenario was run with different coefficients and temperatures to assure the reliability of the model. In the report, only one case for each scenario is presented as verification in order to save space. Subsequently, each scenario was set up in COMSOL to verify the MATLAB generated results. All the simulations were made with the geometry of *Version I*. For the MATLAB simulation three elements were used, see Figure 10. In COMSOL the number of elements was iterated until a converged solution was found.

Convection Verification

In order to verify the convection the following case was defined:

- One of the short sides of the model was prescribed with a temperature, T_1, of 50 °C.
- The ambient temperature, T_a was set to 30 °C.
- The convection coefficient h, was set to 40 W/(m^2 °K·).
- Initial temperature, T_{start} was set to 10 °C for all nodes.
- The total size of the brick was set to 0,6x0,2x0,2 m.
- The brick material was set to aluminum, with material properties from the COMSOL library.

Figure 13, shows a schematic view of the input system to the simulated model in COMSOL.

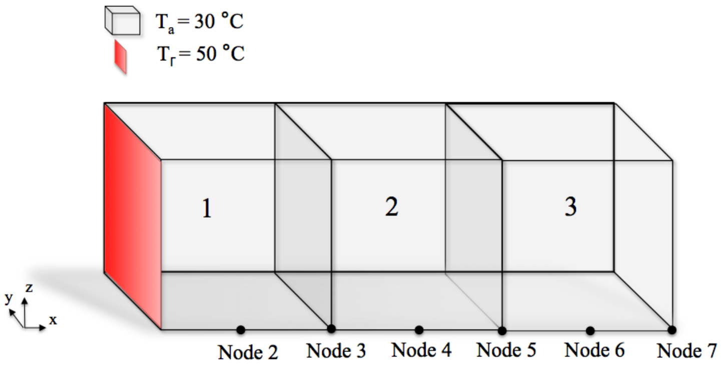


Figure 13. Input values and boundaries for the simulations.

Results from the simulation in MATLAB can be seen in Figure 14. Node 2 had the highest temperature of 47 °C, 17°C above the ambient temperature of 30 °C. This was logical since it was the closest node to the prescribed temperature boundary and thereby the node that was most affected by it. As seen in Figure 14, the effect from the prescribed boundary temperature decreased with increased node number i.e. the nodes with the greatest distance from the prescribed temperature was least affected by it.

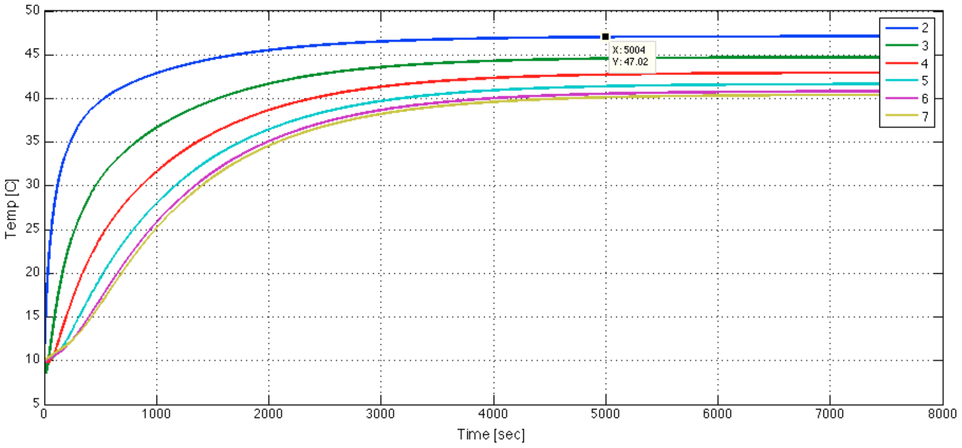


Figure 14. Node temperatures. h=40 W/(m^2 °K). (MATLAB)

The temperature distribution along the brick's length, after 5000 seconds, was visualized in MATLAB and can be seen in Figure 15

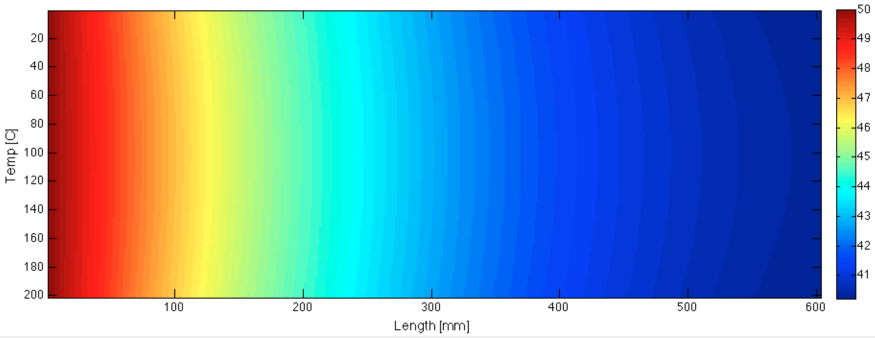


Figure 15. Temperature distribution along the brick's long side. (MATLAB)

The same case was conducted in COMSOL and the node temperatures were plotted as function of time, see Figure 16. The results from COMSOL correspond with the curves from MATLAB, thereby verifying the convection function of the computational model.

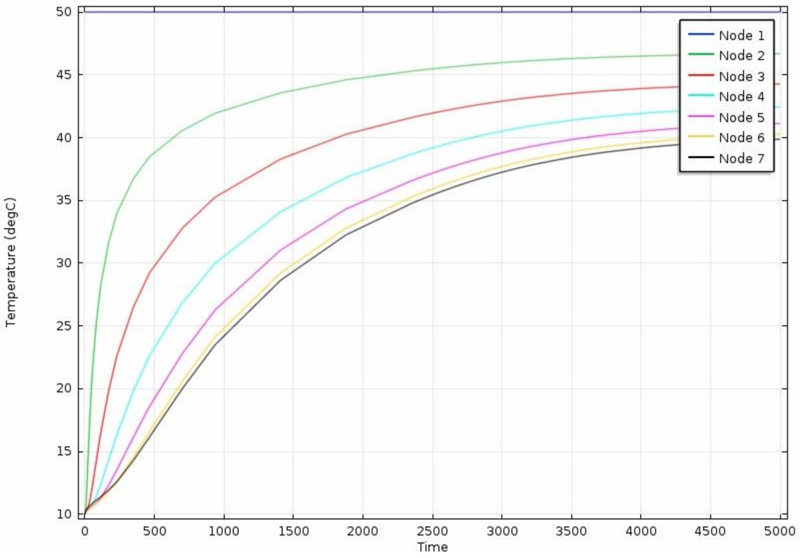


Figure 16. Node temperatures. h=40 W/m^2. (COMSOL)

The temperature distribution after 5000 seconds was plotted in COMSOL and can be seen in Figure 17. The MATLAB and COMSOL simulation demonstrates equal results, which verifies that MATLAB heat transfer calculations are correct.

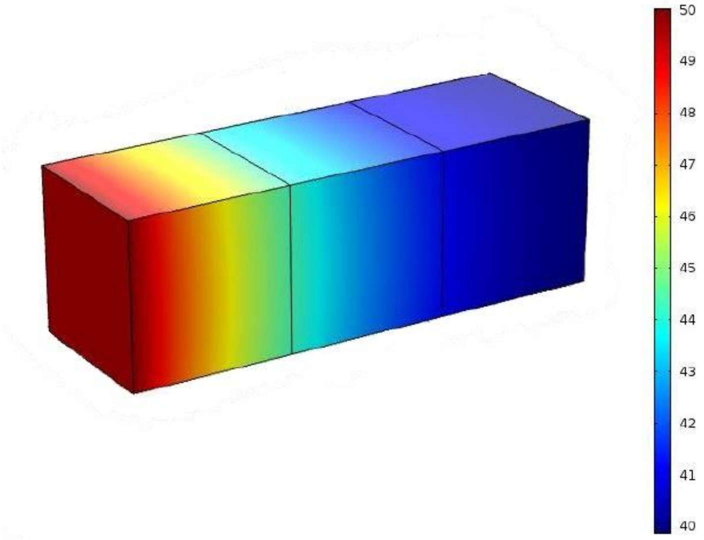


Figure 17. Temperature distribution of the brick. (COMSOL)

A summary of the simulations from *Version I* and COMSOL can be seen in Table 2. The average error for each node temperature was calculated to below 0.6 %.

Table 2. Node temperatures in MATALB and COMSOL at t =3000 s, h= 40 W/m^2

Temperatures	MATLAB	COMSOL	Δ	% Δ
Node 2	46,3 °C	46,1 °C	0,2 °C	0,43
Node 3	43,5 °C	43,2 °C	0,3 °C	0,69
Node 4	40,9 °C	40,6 °C	0,3 °C	0,73
Node 5	39,6 °C	39,4 °C	0,2 °C	0,51
Node 6	39,1 °C	38,9 °C	0,2 °C	0,52
Node 7	38,1 ° C	37,9 °C	0,2 °C	0,57

Radiation Verification

The brick dimensions in the radiation model were of the same size as in the convection case. The following parameters were used in MATLAB and COMSOL.

- Radiation was applied to all surfaces except the left temperature boundary which was fixed at 50 °C
- The ambient temperature, T_a was set to 30 °C
- Initial temperature, T_{start} was set to 10 °C for all nodes.
- The convection coefficient, *h*, was set to 0 W/(m^2 °K·).
- The emissivity and absorbance of the model was set to 0,5.
- The emitted radiation from the environment was set to 1368 W/m^2 (emitted energy from the Sun according to Gilmore (2002)).

Results from the MATLAB simulation can be seen in Figure 18. Each curve represents a node in the brick. Due to the large heat flux, the node temperatures are all above the ambient temperature and most nodes reached a steady state temperature between 60 °C and 70 °C. As in the convection case, node 2 was influenced the most by the boundary conditions and had thereby a lower temperature, closer to 50 °C. It can be seen that all the nodes reached the same temperature (51°C) at *t* equal to 2000 s.

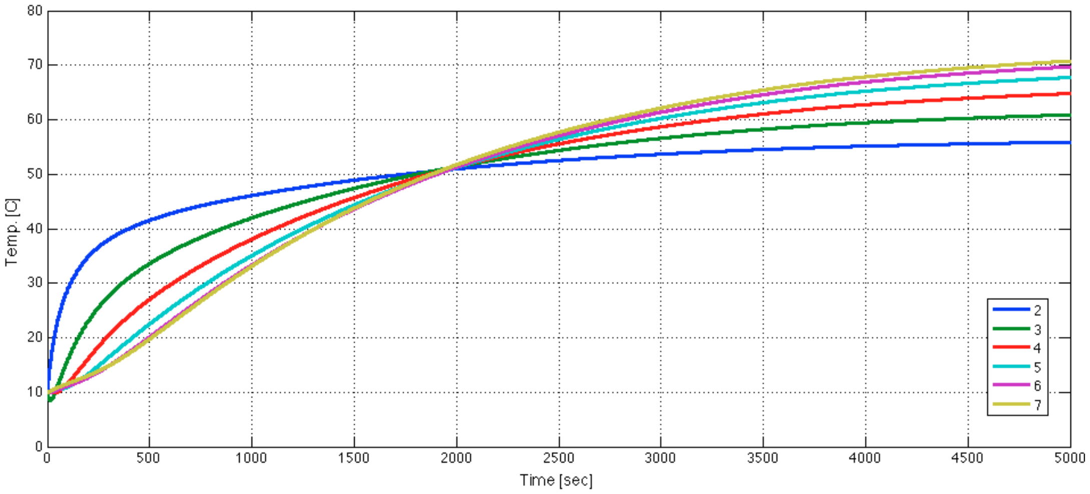


Figure 18. Transient heating from radiation. q_r= 1368 W/m^2. (MATLAB)

Figure 19 shows the temperature distribution from the MATLAB simulations after 5000 second.

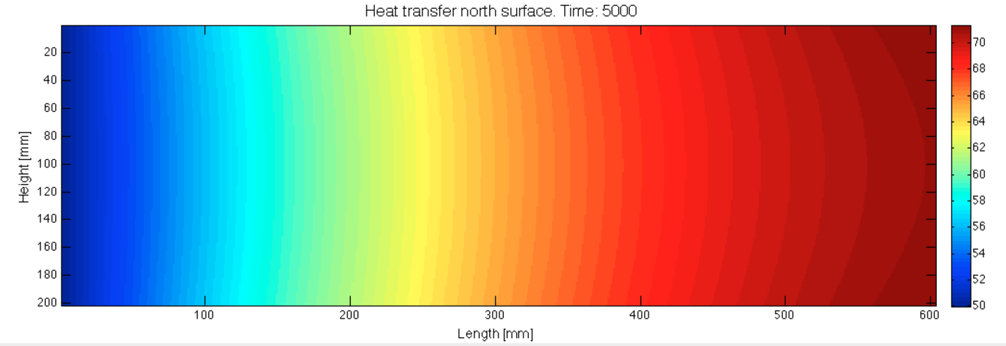


Figure 19. Temperature distribution of the brick's long side. (MATLAB)

The temperature curves from COMSOL, see Figure 20, show a good agreement with the ones from MATLAB. The steady state temperatures were almost the same in both programs. The time when the all nodes had equal temperature was however slightly delayed and occurred at t equal to 2100 s. The temperature at this time was 51 °C for all nodes.

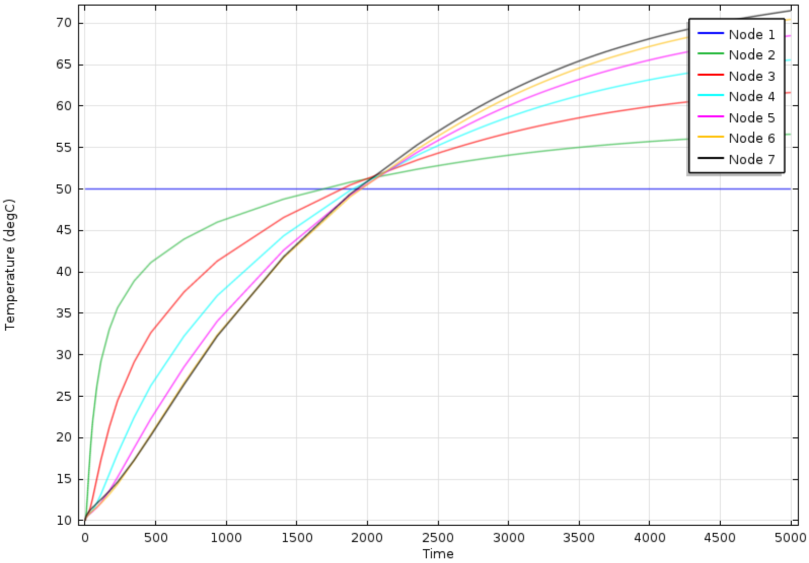


Figure 20. Transient heating from radiation. q_r= 1368 W/m^2. (COMSOL)

The temperature distribution after 5000 seconds from the simulation in COMSOL can be seen in Figure 21.

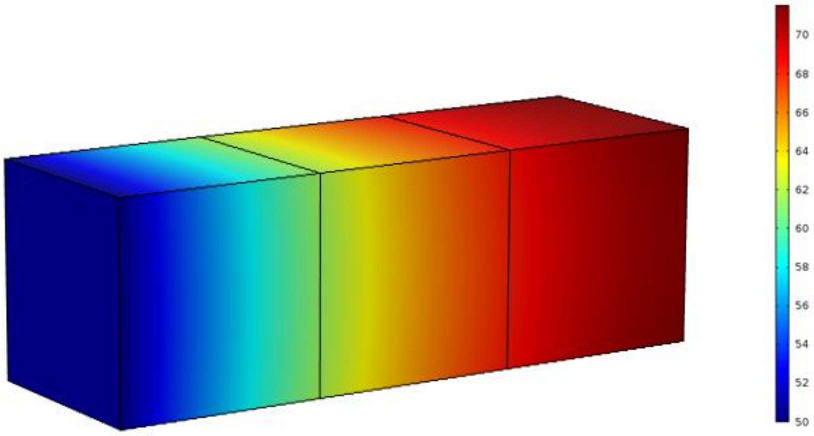


Figure 21. Temperature of the brick. q_r = 1368 W/m^2. (COMSOL)

The similar results from the simulations indicated that the method for calculating radiation in the computational model was correct. Table 3 shows a summary of the node temperatures both in MATLAB and COMSOL. The average error for the node temperatures was 0,25 %.

Table 3. Node temperatures from MATLAB compared with COMSOL t=5000s.

Temperatures	MATLAB	COMSOL	Δ	% Δ
Node 2	56,5 °C	56,6 °C	-0,1 °C	-0,18
Node 3	61,4 °C	61,6 °C	-0,2 °C	-0,31
Node 4	65,4 °C	65,6 °C	-0,2 °C	-0,29
Node 5	68,3 °C	68,5 °C	-0,2 °C	-0,28
Node 6	70,3 ° C	70,5 °C	-0,2 °C	-0,41
Node 7	71,4	71,5 °C	-0,1 °C	-0,25

Internal Generated Heat Source

The same brick configuration was used for the internal heating verifications. The following data was applied on the brick,

- The internal generated heat was set to 120 W for the middle element.
- The ambient temperature, T_a was set to 10 °C
- Initial temperature, T_{start} was set to 10 °C for all nodes.
- The convection coefficient, h, was set to 40 W/(m^2°K·).
- No radiation was assumed.
- No prescribed temperatures on boundaries

The node temperatures from the transient MATLAB simulation can be seen in Figure 22. All nodes show a similar increase in temperature. This is explained by the high thermal conductivity of aluminum. For materials with lower conductivity the node temperature was more “scattered”. In this case, nodes belonging to *element 2* had a higher temperature.

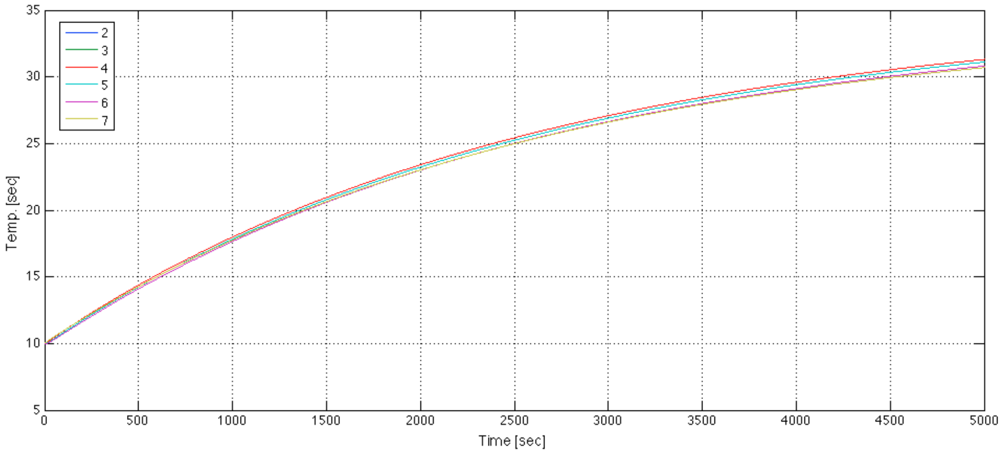


Figure 22. Node temperatures from internal generated heat. (MATLAB)

The temperature distribution from the MATLAB simulations can be seen in Figure 23

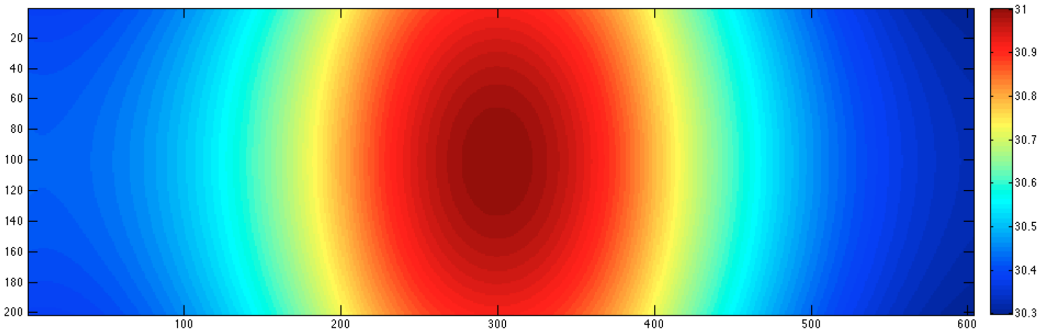


Figure 23. Temperature distribution from the internal heat simulation. (MATLAB)

Node temperatures from the simulation in COMSOL are shown in Figure 24.

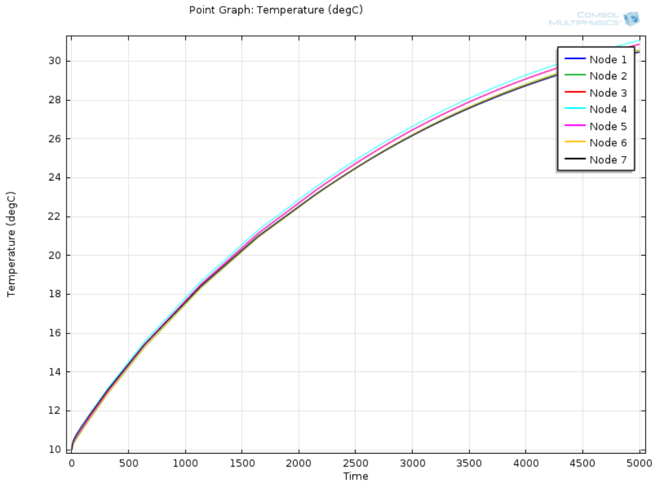


Figure 24. Node temperatures from internal generated heat. (COMSOL)

The temperature distribution from the COMSOL simulations can be seen in Figure 25.

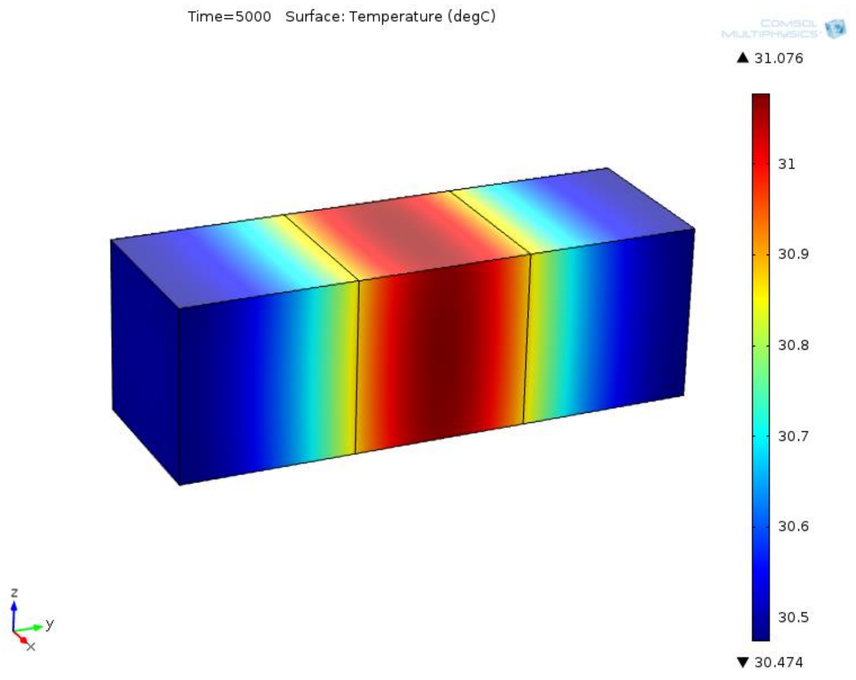


Figure 25. Temperature distribution from the internal heat simulation. (COMSOL)

The node temperatures from the MATLAB and COMSOL simulations were uniformed and showed good agreement over the whole simulation time of 5000 s. However, the distribution of heat looks slightly different. This was because of the finer mesh of the COMSOL model in comparison to the MATLAB, which only contained 3 elements. The oval shape in the developed program occurred due to the shape functions used for calculations and visualization in MATLAB. The middle node of the internal heat element (node number 27) is designed to have a larger influence of heat distribution. Table 3 shows the node temperatures from the simulations at the time 5000 seconds. The average error for the node temperatures was 0,65 %.

Table 4. Brick temperature after 5000 seconds.

Temperatures	MATLAB	COMSOL	Δ	% Δ
Node 2	30,7 °C	30,4 °C	0,3	0,98
Node 3	30,8 °C	30,6 °C	0,3	0,65
Node 4	31,2 °C	30,8 °C	0,4	1,28
Node 5	30,7 °C	30,6 °C	0,1	0,32
Node 6	30,7 °C	30,4 °C	0,3	0,98
Node 7	30,6 °C	30,3 °C	0,2	0,90

3.2 Version II

The second version used the previous verified FEM-equations in a more dynamical environment and with a realistic geometry. In contrast to the implementation of *Version I,* the second version demanded more unique solutions adapted for the MASER experiment modules.

3.2.1 Algorithm for Mesh Generation

In order to allow the user to adjust the number of elements of the mesh and thereby the accuracy of the simulation and the process time, the program had to be able to generate the elements by itself. As mentioned before, the mesh generation divides the experiment module domain into user-defined subdomains. The subdomain is calculated from the user-input values of height, diameter, numbers of sections, numbers of radius and numbers of segments, as seen in Figure 26. The number of sections represents the fragmentation in φ-direction, the number of radius represents the arc length of sectors and the numbers of segments represents the different levels height.

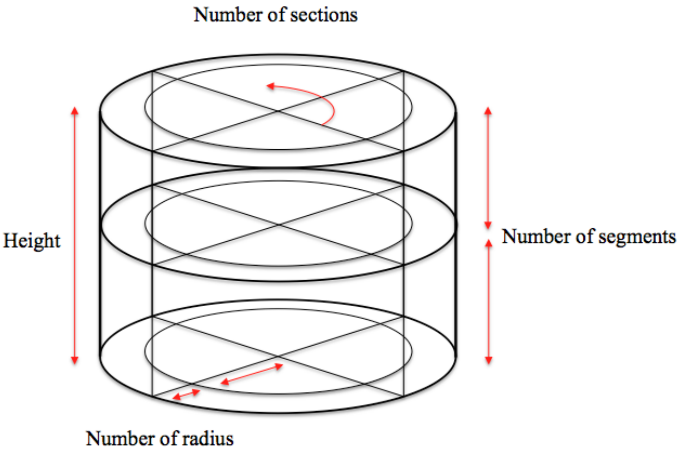


Figure 26. Element generation parameters.

By observing the global node numbering of the first and second generated element a pattern was found. This was used to create of an algorithm for automatic mesh generation i.e. automatic construction of the element matrix. The coordinate system from the mesh generation was modified so that the user can adjust the segment heights, enabling embodiment of, for example, the experiment deck and other thin horizontal sheet metals inside of the module. Figure 27 shows an experiment module with eight sections, two radius and five segments generated in MATLAB. The section height has been adjusted to replicate a thin sheet metal in the middle of the module.

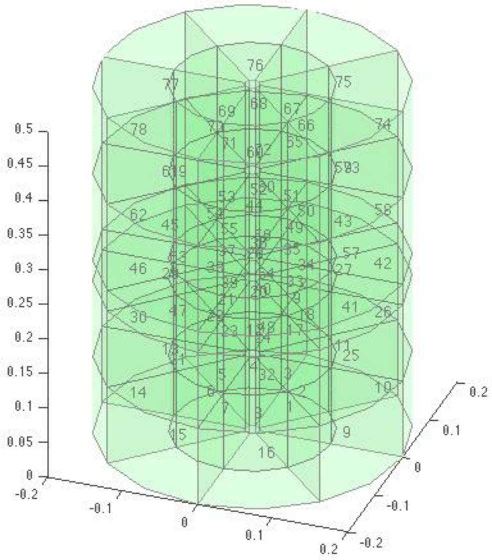


Figure 27. A model of a module divided into elements.

The skin of the module was incorporated into the model by automatically adding three more radii, which represented an air film of 10 mm, insulation with the thickness of 5 mm and an aluminum skin of 10 mm on the outside of the module, see Figure 28. The dimensions were based on measurements of existing modules. The thermal conductivity of the insulation was set to 0.54 W/m (Niklasson & Rössner, 2011).

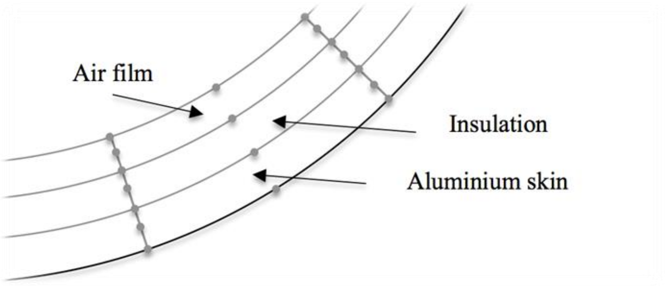


Figure 28. Different elements of the module skin.

3.2.2 Internal Convection

Besides ambient convection on the module's skin convection also takes place in between the components inside the payload. To determine the internal convection another approach other than the outside convection on the rocket skin was required. The inside air temperature was dependent on the internal power consumption of the different components and the temperature distribution in solid elements. Instead of having the convection as a load vector, it was created as a matrix where the columns represented different temperatures. The convection load matrix was expressed as

$$\mathbf{Q}_{i.conv} = \begin{pmatrix} T_{a1} & T_{a2} & \cdots & T_{an} \\ \{\mathbf{q}_{i.conv.1}\} & \{\mathbf{q}_{i.conv.2}\} & \cdots & \{\mathbf{q}_{i.conv.n}\} \end{pmatrix} \tag{68}$$

where $\{\boldsymbol{q}_{i.conv.n}\}$ were the different load vectors corresponding to the temperatures in the first row. During the simulation the program calculates an average temperature of the inside air for each time-step. This temperature was matched against the convection-load matrix finding and selecting the correct load vector. By creating the convection load matrix before the simulation, the program only has to find the right column instead of calculating a new load at each iteration, which reduces the simulation time. A high number of columns was desirable in $\boldsymbol{Q}_{i,conv}$ since this reduced the error related to a large difference between the inside air temperature and the column in. For example, assume that the internal air temperature was 10 °C but the closest column in $\boldsymbol{Q}_{i,conv}$ was evaluated at the temperature 15 °C. This would give an error of 5 °C in the load vector causing the simulation to oscillate. The disadvantage of having a large convection load matrix was that it increased the computational time of the program. By using the logical convection matrix the internal convection could be implemented on the correct nodes.

3.2.3 Time Dependent Boundary Conditions

The dynamic boundary conditions for each phase were implemented sequentially. The implementation was based on the pre study presented in the *Chapter 2. Frame of Reference*.

Pre-Launch

The pre-launch was similar to the model *Version I*, i.e. the environmental was held constant and the module experienced convection to the environment. This meant that no additional or modified boundary conditions had to be created for this phase.

Ascent & Descent

As concluded in previous chapters, both the ascent and descent were two challenging phases from a thermodynamic perspective. Due to the time frame, it was decided to use measured temperatures from previous flights as input data. By collecting measured data from MASER 8 to 12 missions (those used a VSB-30 vehicle) an average skin temperature for each time step was estimated. Based on data from Figure 6, the module placement within the payload section was neglected and all modules were assumed to experience the same aerodynamic heating. The high conductivity of the skin led to the assumption that each node on the skin experienced the same temperature. The skin temperature was implemented in the model as a boundary condition of the first order, i.e. a prescribed temperature.

Microgravity

In the micro gravity phase it was assumed that 180° of the module's surface were exposed to radiation from the sun and the other side was affected by the earth's IR and albedo, see Figure 29. It was assumed that the side of the module facing the sun was perpendicular to the incoming radiation. The angle between the module and the Earth was calculated using the shape factors from Ross (2003). The change in albedo related to the orientation of the rocket during a flight was neglected due to the short time span of the project.

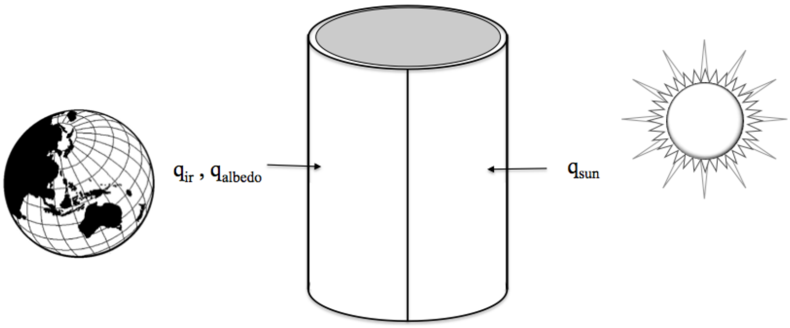


Figure 29. Environmental radiation on the module.

The absorbed heat flux for each element was calculated as

$$\{\mathbf{q}_{rad}\} = \int_{\Gamma} q_r [\mathbf{N}]^T d\Gamma_q \tag{69}$$

where q_r was the sum of earth albedo, q_{albedo} and IR flux, q_{IR} for the side facing the Earth and heat flux from the sun, q_{sun} for the other side. To apply the correct flux to the right element and correct surface the same method as for convection was used i.e. a logical radiation matrix was constructed for the module.

The emitted heat flux from the rocket was dependent on the temperature of the surface raised by four (see equation(7)), which can be problematic since the temperature is not linear. This was solved in a similar way to the internal convection by creating an emitted radiation load matrix where each column represented a load vector at a certain temperature. The average skin temperature was calculated in each iteration and matched against this matrix. In this phase there were no prescribed temperatures, instead it was only second order boundary conditions.

Recovery

During recovery the impact area and season determines the outside boundary for the payload. The snow has a direct cooling effect on the payload although it is isolating the rocket from the environment. This case was implemented in the MATLAB program by a recovery-convection input variable.

3.2.4 Verification of Version II

The algorithm for generating the circular mesh and internal convection was verified using COMSOL. When setting up a CFD-model in COMSOL, the mesh size and geometry plays and important role in order to find a good solution (Freitag & Olliver-Goch, 2008). Therefore, different mesh sizes were investigated until the simulated results converged.

A 2D-axisymetric model was developed, see Figure 30. A volume force simulating the natural internal convection was defined as the air density times the gravity coefficient.

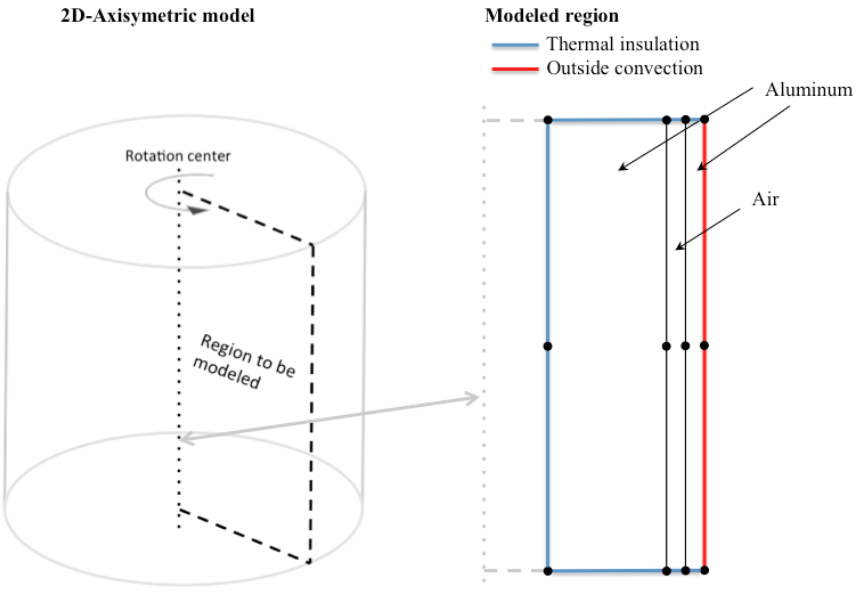


Figure 30. 2D-axisymetric schematic model input to COMSOL.

The input parameters to MATLAB and COMSOL for the verification of internal convection were,

- The internal convection coefficient, h_i was set to 10 W/(°K·m^2)
- The external convection coefficient, h_o was set to 100 W/(°K·m^2)
- The thermal conductivity of air was set to 0.5 W/(°K·m) to decrease the simulation time. In future versions, the conductivity of air was set to 0.0226 W/(°K·m).
- The ambient temperature, T_a was set to 30 °C
- Initial temperature, T_{start} was set to 10 °C for all nodes

Both simulations showed similar behavior. They both had a skin temperature around of 18 °C and an inside temperature of 10 °C. In Figure 31 the results from the MATLAB and COMSOL model are displayed.

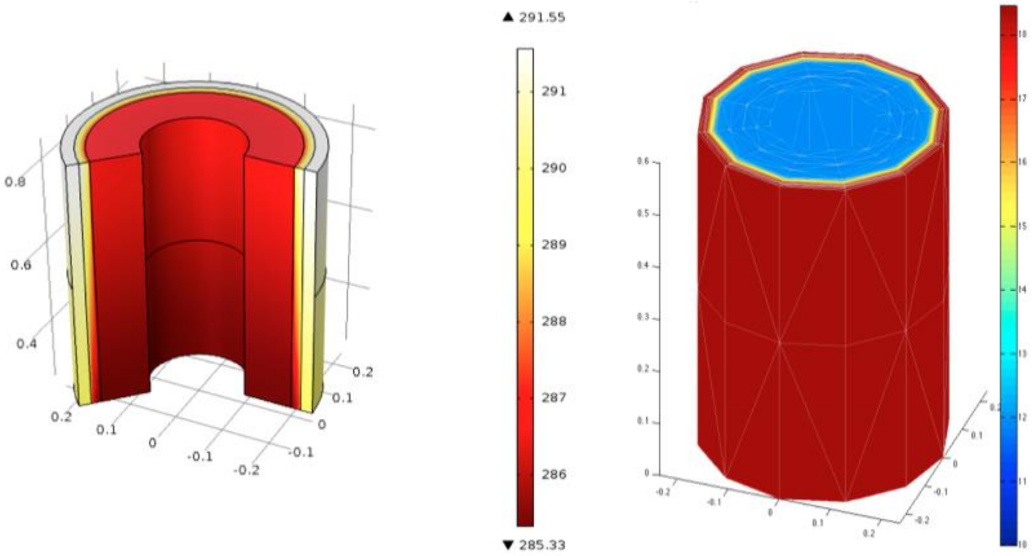


Figure 31. Data representation of Version II from COMSOL and MATLAB at 2000 seconds.

The temperature curves from the MATLAB simulation are presented in Figure 32. The nodes closest to the skin showed the fastest temperature increase.

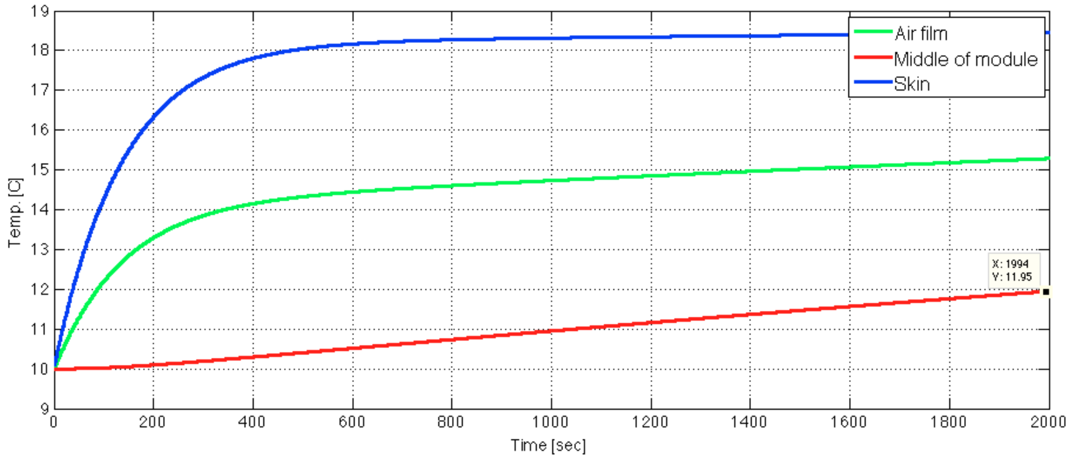


Figure 32. MATLAB simulation of internal natural convection.

Figure 33 shows the temperatures in the module from the COMSOL simulations.

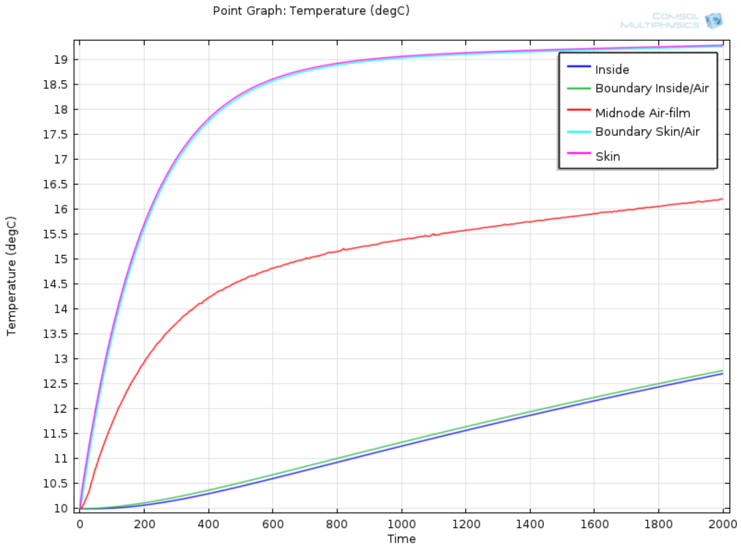


Figure 33. COMSOL simulation of internal convection.

The MATLAB model results correspond with the CFD-model results generated by COMSOL. Detailed temperature data after 2000 seconds can be seen in Table 4. The average error in the node temperature was 5,8 %. It is worth mentioning that the results from MATLAB are highly dependent on the number of columns in the internal convection- matrix. An increased number of columns would have increased the accuracy of the internal convection but was deemed unnecessary since the difference between the programs were small.

Table 5. Simulation results of internal convection after 2000 s.

Temperatures	MATLAB	COMSOL	Δ	% Δ
Skin	18,5 °C	19,2 °C	0,7 °C	3,78
Air-film	15,3 °C	16,2 °C	1,2 °C	5,83
Inside/Middle of module	12 °C	12,7 °C	0,7 °C	5,82

3.3 Version III

Version III was the final program version. In this version much effort was placed on designing a user-friendly GUI where several parameters could be controlled. Additional functions such as external power and internal radiation was added.

3.3.1 External Power

During pre-launch the rocket is kept constant using external active temperature control. To get a rough estimation of how much external power that is needed to keep the rocket at a constant temperature a thermal equilibrium was set up and investigated, based on equation (30). By re-arranging the terms in this equation the external power was expressed as

$$\{\mathbf{q}_{ext}\} = ([\mathbf{K}_{cond}] + [\mathbf{K}_{conv}] + [\mathbf{K}_{rad}])\{\mathbf{T}\} - \{\mathbf{q}_{conv}\} - \{\mathbf{q}_{int}\} - \{\mathbf{q}_{rad}\} \qquad (70)$$

where the different *q-vectors* are the loads from convection, internal generated heat and the radiation. The *K-matrices* are the global “stiffness” against the different modes of heat transfer. The vector $\boldsymbol{T}$ contains the temperature for the prescribed nodes. Note that the whole system is reduced to contain only the rows for the prescribed temperatures where the reaction heat flow takes place. From this equation the heat flow can be calculated for each time step and node. By summarizing the heat flow of the nodes and thereafter taking a mean value of this sum, for each time steps, the average required external power was estimated. The external power had an initial spike where the values were unrealistic large before converging towards a stable value. To avoid this from affecting the results the first values of the external power was not included in the calculations.

The external power was implemented in the program by letting the user first choose which element that was going to have a prescribed temperature and thereafter letting the user choose which sides were affected. This was judged as the best way to model the geometry of the interface between the module and external heater.

3.3.2 Internal Radiation

For a pressurized module, the surfaces are enclosed by gas and the contribution from radiation can be combined with the convection coefficient to obtain an approximation. When the convection is forced (h= 25-250 W/m^2°K), the surfaces have low emissivity and the temperatures are moderate, radiation is often not taken into consideration due to its small contribution. (Cengel et al., 2008)

For a non-pressurized module in micro-gravity, radiation is the only heat transfer occurring between components and can have a significant effect on the temperatures, especially if there are large temperature differences between the components (Cengel et al., 2008). The absorbed and emitted heat is highly depended on the shape and surface of the element, which makes it challenging to simulate accurately without knowing the real geometry. Besides the characteristics of the components, their placement in the module also affects the heat flow from radiation. Since the module is a closed volume it is also very likely that emitted internal radiation will reflect on the interior, hitting components that are not in the line of sight of the source. This behavior can be time consuming to study and implement and was therefore not taken into considerations. Instead, the following approach was used to approximate the internal radiation. First of all, the elements normal were calculated from in each local node, see Figure 34. The normal vector was used to take a “step” in the global coordinate system to create a second point. A line was defined between these two points.

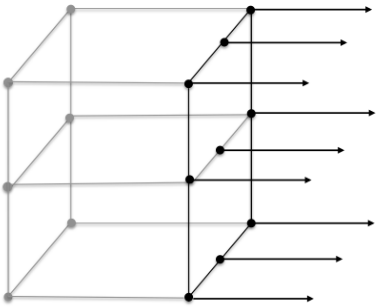


Figure 34. Normal vector of the element surface in each node.

Potential radiation nodes on solid element's boundaries were found using the convection-matrix created during the mesh generation. The shortest distance from solid element's nodes and the drawn line was calculated as

$$d = \frac{|(\mathbf{Q_2} - \mathbf{Q_1})\mathbf{x}(\mathbf{P} - \mathbf{Q_1})|}{|(\mathbf{Q_2} - \mathbf{Q_1})|} \tag{71}$$

where $\boldsymbol{Q_1}$ was the node coordinates expressed as (x_1, y_1, z_1), $\boldsymbol{Q_2}$ was the second point's coordinates and $\boldsymbol{P}$ was the coordinates of the potential radiation nodes. By finding the node with the shortest distance, d to the drawn line, the nodes that was exposed to radiation from the element was found. This is illustrated in Figure 35 where *element 1* emits radiation from one side to *element 2* and *element 3*. The nodes that are closest to the drawn line are highlighted in red.

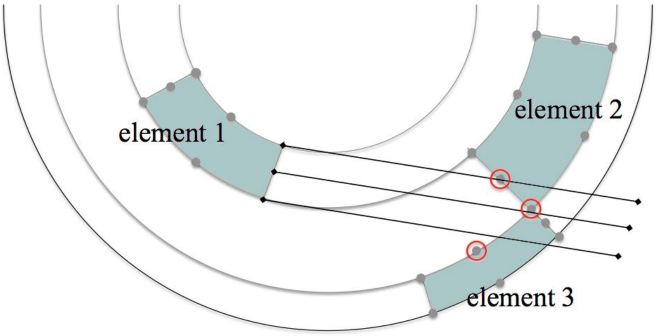


Figure 35. Element 1 emits radiation from one side to element 2 and element 3. Filled areas represent solid elements.

By knowing which nodes that were interacting and exchanging energy, the radiation could be linearized (see *Chapter 2.3.2*) and a radiation stiffness matrix could be constructed. In accordance with Holman (2010) the heat flow was seen as a "current" in a network of nodes as see Figure 36.

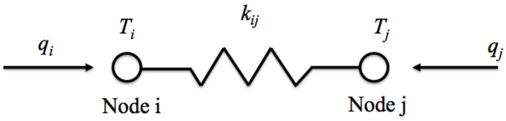


Figure 36. Radiation network between nodes

A resistance, k_{ij} in Figure 36, was set up between the nodes that were connected through radiation. For the whole module, the heat flow related to radiation could be written as the following matrix product

$$\{\mathbf{q}_{rad}\} = [\mathbf{K}_{rad}]\{\mathbf{T}\} \tag{72}$$

where $\boldsymbol{K_{rad}}$ contains the radiation resistance/stiffness between nodes. Based on the equation (8) the stiffness between two temperatures could be expressed as

$$k_{ij} = (\frac{1}{\mathcal{E}_i} + \frac{1}{\mathcal{E}_j} - 1)^{-1} \cdot T_m^3 \cdot \sigma \tag{73}$$

where T_m is the arithmetic mean values of T_i and T_j . By using this method to approximate the radiation no respect is given to reflection inside the module, instead all radiation is assumed absorbed by the first node it hits.

3.3.4 Graphical User Interface

The graphical user interface was developed using MATLAB built-in script GUIDE. The GUI eases the input definition of necessary vectors and matrices. Figure 37 illustrates a schematic view of the developed GUI and how the information flows from the *Main menu* to the data visualization in the *Results* window. In Figure 37, boxes with bolded black borders emblem a program window with belonging input scripts.

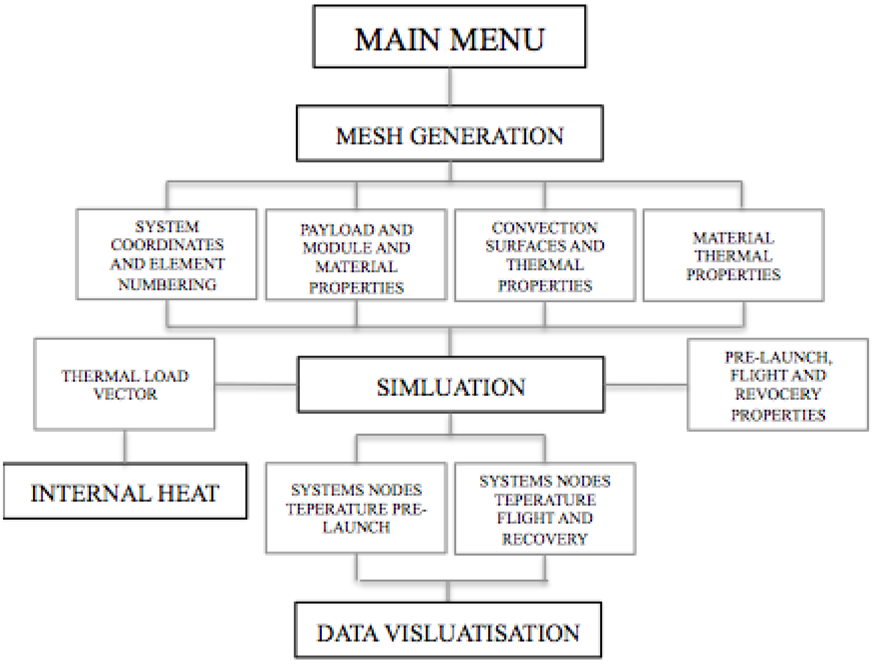


Figure 37. Schematic view of the GUI structure.

3.3.5 Verification of Version III

The internal radiation was verified by comparing the results from the model with results from equation (7) for two cylindrical surfaces. The area for the outer surface was calculated with following parameters,

- Radius (r_1) equal 220 mm.
- Radius (r_2) equal 210 mm.
- Length (L) equal 1200 mm.

Since the difference between the radiuses was assumed to be small the geometrical factor, F_{12} was set to 1. The used number of elements in the model was set to 12 with four elements on the inside and outside together with four elements representing the air film.

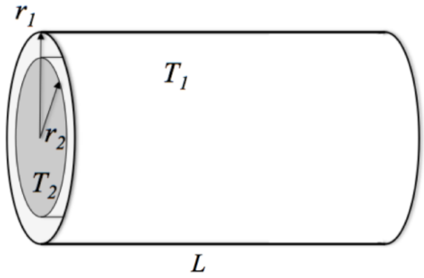


Figure 38. Radiation between two cylindrical surfaces.

To determine the total heat flow from the MATLAB model the nodes on the outside of the skin were summed up. The difference between the two methods can be seen in Table 6. Since the difference was below 5 % for all temperatures it was concluded that the numerical method in the model had acceptable accuracy. With an increased amount of elements it was expected that the error would decrease, however this was not investigated further.

Table 6. Comparison between the analytical calculation and MATLAB simulations.

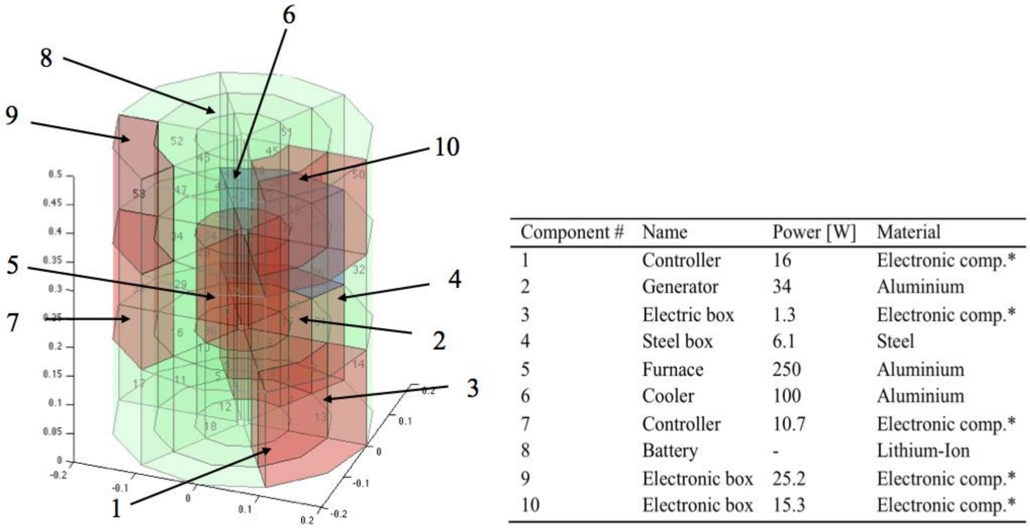
Temperature	Analytical Q_{12} [W]	Numerical Q_{12} [W]	Δ [W]	% Δ
T_1=20°C, T_2=10°C	89,9	93,4	3,5	(3,9 %)
T_1=50°C, T_2=0°C	501,3	494,5	-6,4	(-1,3 %)
T_1=85°C, T_2=0°C	1022,5	998,6	-24,0	(-2,3 %)

3.4 Simulation Configurations

Simulations were conducted in order to compare results with measured data and find important parameters for the internal temperature distribution.

3.4.1 Parameter observation

To analyze which parameters that have a large effect on the temperature distribution in the module, a test module was constructed. It contained cooling elements, heat generating components and structural elements see Figure 39. Note that the structural elements are not displayed in Figure 39, only the internal energy sources are shown.



Component #	Name	Power [W]	Material
1	Controller	16	Electronic comp.*
2	Generator	34	Aluminium
3	Electric box	1.3	Electronic comp.*
4	Steel box	6.1	Steel
5	Furnace	250	Aluminium
6	Cooler	100	Aluminium
7	Controller	10.7	Electronic comp.*
8	Battery	-	Lithium-Ion
9	Electronic box	25.2	Electronic comp.*
10	Electronic box	15.3	Electronic comp.*

Figure 39. Test module.

The behavior of the module was studied by adjusting the convection coefficient and conductivity of the inner frame. The different cases that were tested are showed in Table 7. The inner frame was constructed of aluminum in all cases except copper in *Case 4* and carbon fiber with reinforce epoxy in *Case 7*. Subsequently, all cases were simulated for 3000 sec imitating a long pre-launch phase followed by a nominal flight phase and a 100 sec long recovery-phase. The pre-launch environment as well as the initial temperature was set to 20° C. The convection coefficient between the skin of the module and the environment was set to 50 W/(m^2 °K) for both pre-launch and recovery. In the recovery phase the environment temperature was set to 10° C. The module was set to constant atmospheric pressure during the simulation for the convection properties.

Table 7. Properties of the tested cases.

	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
h [W/(m^2 °K)]	**10**	**40**	**100**	10	10	10	10
k_{frame} [W/(m^2 °K)]	170	170	170	**400**	**220**	**80**	**24**

3.4.2 Simulation of the Payload Module XRMON-GF

A simulation of the module XRMON-GF was set up to compare the result with real flight data. The module was embodiment in the MATLAB program with eight sections, three radii and eight segments, a total of 192 elements. The module with its components can be seen in Figure 40.

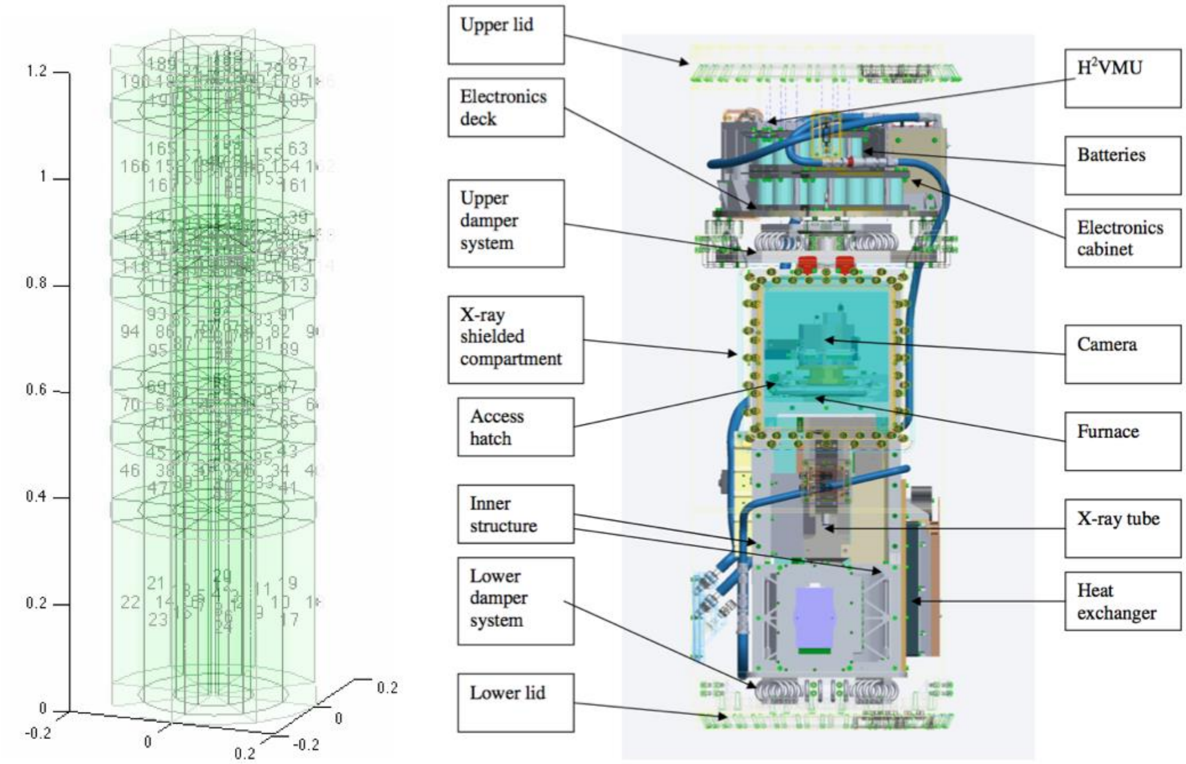


Figure 40. a) The module and its elements in the program. b) Placement of components in MASER payload.

The properties and placement of the components where based on the work by Niklasson & Rössner (2011) and Törnqvist (2011). The simulation was run with a 20 000 sec long prelaunch sequence to reach a steady state. The following input parameters were used,

- An environment temperature of 20 °C
- An internal convection coefficient of 15 W/(m^2 °K)

- An external convection coefficient of 30 W/(m^2 °K)
- A recovery temperature of 5 °C

Module components are presented in Table 8 together with their power.

Table 8. Input form the simulation - Component properties.

	k [W/m^2°K]	ρ [kg/m^3]	c_p [J/(kg K°)]	ε	Consumed Power [W]
Electronic cabinet		Aluminum		0.5	25,2
H^2VMU		Aluminum		0.5	15,3
Batteries	2.33	3000	677	0.6	-
X-ray generator		Steel		0.5	34
Furnace		Aluminum		0.8	250
Camera		Aluminum		0.8	26.8
X-ray tube		Steel		0.6	27.2
DC/DC converter		3000	1000	0.6	10.7
Heat exchanger		Aluminum		0.7	T=20°
µC ServoPW board	25	1000	1000	0.5	3.5
µC Thermo board	25	1000	1000	0.5	1.3
Gauge		Aluminum		0.6	13,5
X-ray controller	100	1000	1000	0.7	42.5

4 RESULTS

The results in this chapter are divided into three parts. In the first, the developed program is presented together with the requirement specification and the interface of the program. Thereafter, the results from the simulations with the test module are shown. In the end of the chapter the result from the simulation of the module XRMON-GF are presented and compared with previous flight data.

4.1 The computational model

The developed computational model is a result from the needs defined in the *Chapter 1.2. Problem Description*. In this section, the model's outcome in terms of interface and function are compared to the requirement specification. A schematic picture of the structure of the program can be seen in Appendix F.

4.1.1 Validation of model

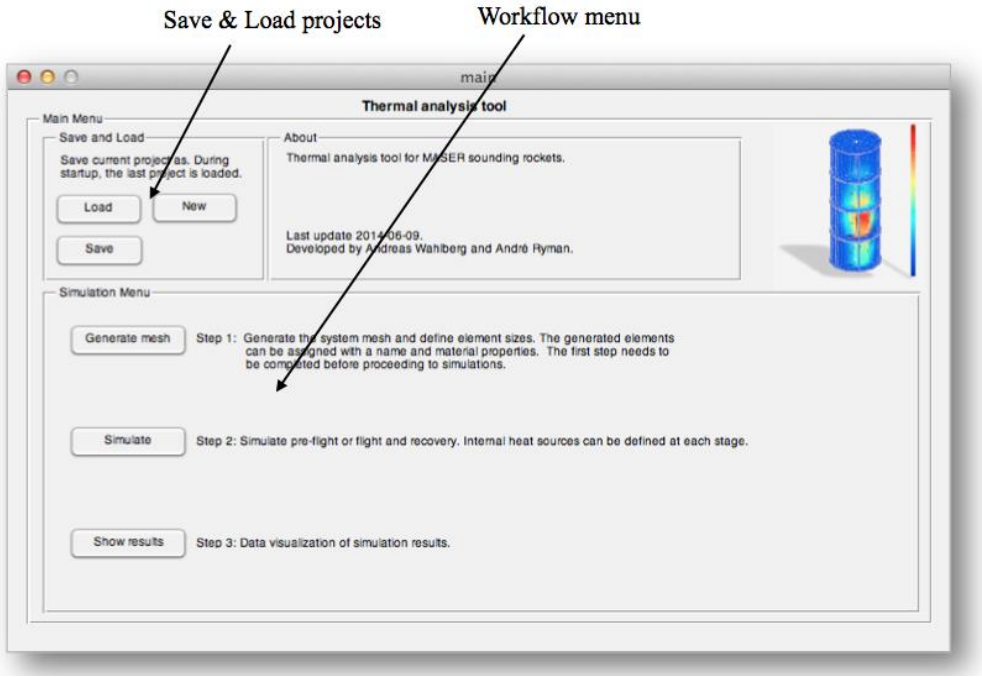
Since a significant part of this MSc thesis project was to develop a flexible computational model for thermal calculation, the program itself is the most important result. In Table 9 the central program capabilities are compared with the requirement specification. The program satisfies most of the initial requirements.

Table 9. Important requirements and status.

Index	Requirement	Achieved	MATLAB model comments
C-1	Pre-launch phase	✓	The pre-launched phase can be simulated in the MATLAB model. The GUI enables the user to define environmental temperature, environment convection, start temperature inside module, simulation time, internal heat sources and prescribed temperature at element boundaries. All requirements from the PDS have been fulfilled.
C-2	Ascending phase	✓	The GUI enables the user to define the internal heat during the ascending phase were the surrounding temperature on the payload boundary is based on previous SSC flights.
C-4	Microgravity phase	✓	During the microgravity stage, the MATLAB model takes into consideration radiation from the sun and reflections from earth. The internal pressure is assumed to be zero for open experiment modules and 101 kPa for pressurized modules. Internal heat sources can be adjusted by the user.
C-5	Descending phase	✓	The MATLAB model is prescribing temperatures at the payloads boundary. Many of the experiment equipment are turned off in this stage, which can be simulated, by turning off the internal heating sources.
C-6	Recovery phase	✓	The GUI enables the user to adjust the recovery time, surrounding temperature and convection coefficient. Also, the internal heat source can be defined.
D-1	User interface	✓	The key benefit with the mesh generation function is that the size of elements can be varied and material properties can be defined to represent specific components. Also, the width and radius of the payload can be adjusted which enables sub-system simulations. All user interface requirement has been fulfilled.

4.1.2 Graphical User Interface

The GUI and the user input adjustable parameters are presented in this section together with the output visualization. The main menu of the program can be seen in Figure 41. It contains paths to the three different simulations steps; *Generate mesh*, *Simulate* and *Show results*. The main menu also offers the ability to save and load projects.



Save current project as. During startup, the last project is loaded.

Thermal analysis tool for MASER sounding rockets.

Last update 2014-06-09.
Developed by Andreas Wahlberg and André Ryman.

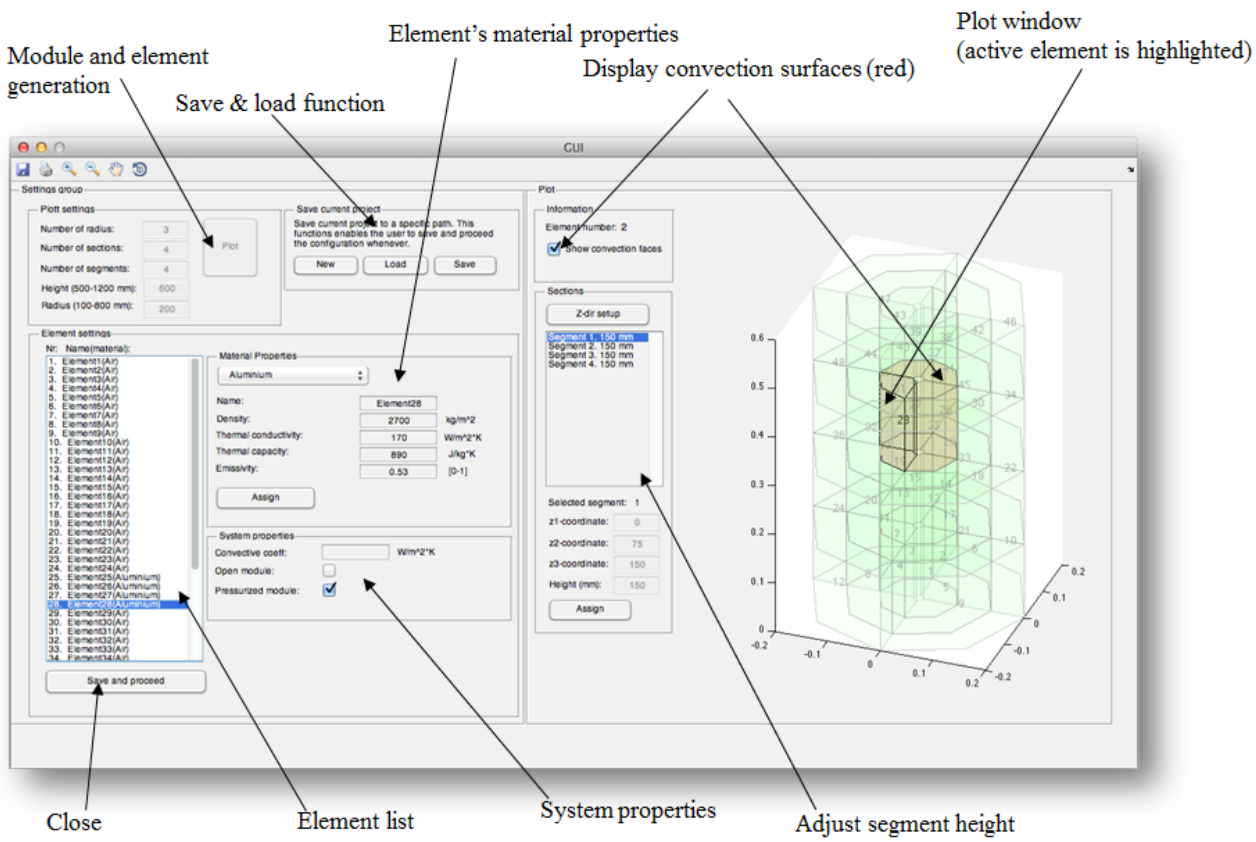
Step 1: Generate the system mesh and define element sizes. The generated elements can be assigned with a name and material properties. The first step needs to be completed before proceeding to simulations.

Step 2: Simulate pre-flight or flight and recovery. Internal heat sources can be defined at each stage.

Step 3: Data visualization of simulation results.

Figure 41. Main menu.

In the *Generate mesh* window, see Figure 42, the mesh is created by defining number of radius, sections, and segments. The height of each segment can be adjusted to image thin components. Element properties such as thermal conductivity, density and emissivity can be defined for each element. System attributes such as open or pressurized module together with internal convection coefficient are defined in this step. The window is saved and closed when pressing *Save and Proceed*.



Settings group

Plott settings

Number of radius: 3
Number of sections: 4
Number of segments: 4
Height (500-1200 mm): 600
Radius (100-800 mm): 200

Save current project

Save current project to a specific path. This functions enables the user to save and proceed the configuration whenever.

Element settings

Nr: Name(material):

1. Element1(Air)
2. Element2(Air)
3. Element3(Air)
4. Element4(Air)
5. Element5(Air)
6. Element6(Air)
7. Element7(Air)
8. Element8(Air)
9. Element9(Air)
10. Element10(Air)
11. Element11(Air)
12. Element12(Air)
13. Element13(Air)
14. Element14(Air)
15. Element15(Air)
16. Element16(Air)
17. Element17(Air)
18. Element18(Air)
19. Element19(Air)
20. Element20(Air)
21. Element21(Air)
22. Element22(Air)
23. Element23(Air)
24. Element24(Air)
25. Element25(Aluminium)
26. Element26(Aluminium)
27. Element27(Aluminium)
28. Element28(Aluminium)
29. Element29(Air)
30. Element30(Air)
31. Element31(Air)
32. Element32(Air)
33. Element33(Air)
34. Element34(Air)

Material Properties

Aluminium

Name: Element28
Density: 2700 kg/m^2
Thermal conductivity: 170 W/m^2*K
Thermal capacity: 890 J/kg*K
Emissivity: 0.53 [0-1]

System properties

Convective coeff: W/m^2*K
Open module: ☐
Pressurized module: ☑

Information

Element number: 2
☑ Show convection faces

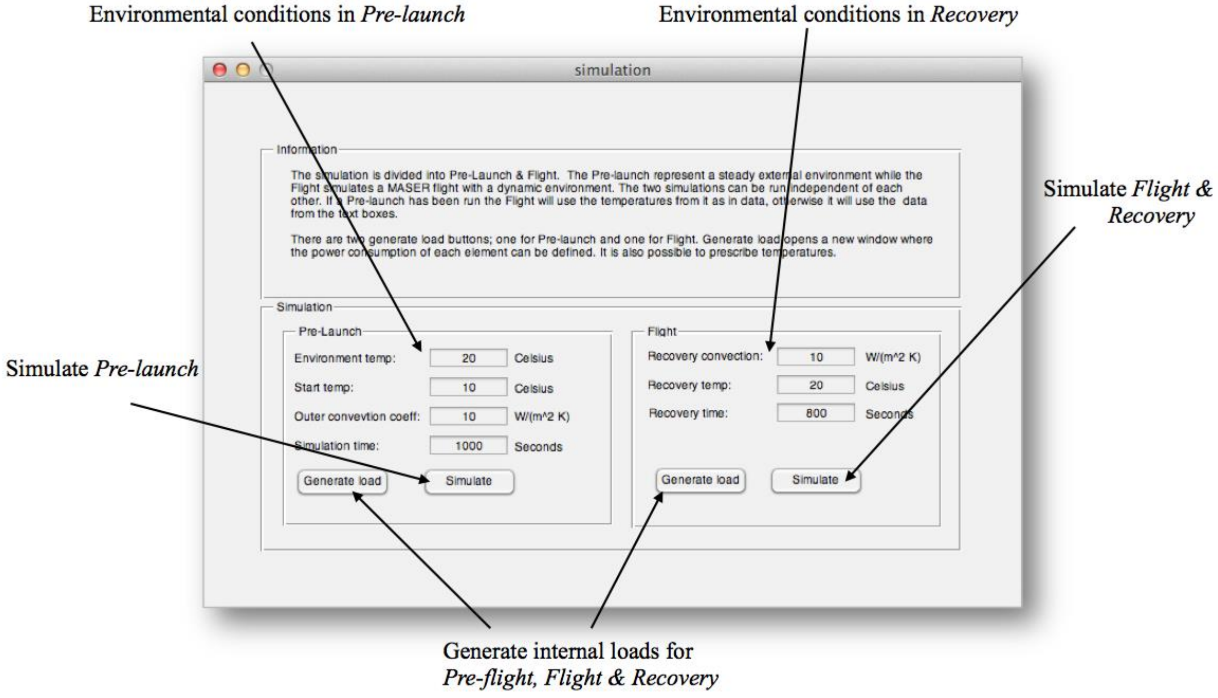
Sections

Segment 1. 150 mm
Segment 2. 150 mm
Segment 3. 150 mm
Segment 4. 150 mm

Selected segment: 1
z1-coordinate: 0
z2-coordinate: 75
z3-coordinate: 150
Height (mm): 150

Figure 42. Generate mesh window.

The second step of the process is to perform the Simulation setup, see Figure 43. Parameters such as initial temperature, environmental temperatures, outer convection coefficients and simulation time for the phases, pre-launch and recovery can be assigned here. By choosing *Generate load* a new window is opened, see Figure 44. The simulations start when pressing *Simulate* and after completion the output data is automatically saved.



Information

The simulation is divided into Pre-Launch & Flight. The Pre-launch represent a steady external environment while the Flight simulates a MASER flight with a dynamic environment. The two simulations can be run independent of each other. If a Pre-launch has been run the Flight will use the temperatures from it as in data, otherwise it will use the data from the text boxes.

There are two generate load buttons; one for Pre-launch and one for Flight. Generate load opens a new window where the power consumption of each element can be defined. It is also possible to prescribe temperatures.

Simulation

Pre-Launch

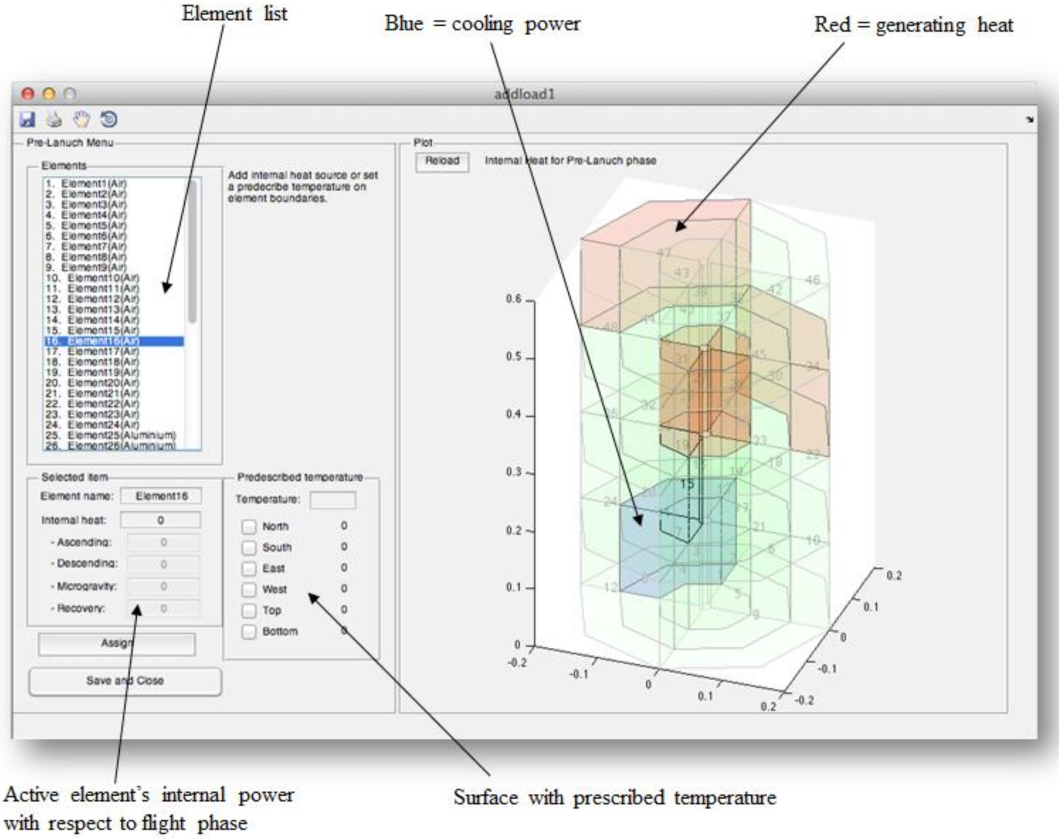
Environment temp: 20 Celsius
Start temp: 10 Celsius
Outer convevtion coeff: 10 W/(m^2 K)
Simulation time: 1000 Seconds

Flight

Recovery convection: 10 W/(m^2 K)
Recovery temp: 20 Celsius
Recovery time: 800 Seconds

Figure 43. Simulation window.

In Figure 44, the *Generate load*-window is shown where the internal power consumption for each element can be assigned. The user can assign different values for power consumption based the phase in order to get a realistic behavior of components that are being turned on and off. It is also possible to prescribe temperatures on elements surfaces. Elements that generate heat are marked with red and cooling elements are marked with blue.



Element list

Blue = cooling power

Red = generating heat

Active element's internal power with respect to flight phase

Surface with prescribed temperature

addload1

Pre-Lanuch Menu

Elements

1. Element1(Air)
2. Element2(Air)
3. Element3(Air)
4. Element4(Air)
5. Element5(Air)
6. Element6(Air)
7. Element7(Air)
8. Element8(Air)
9. Element9(Air)
10. Element10(Air)
11. Element11(Air)
12. Element12(Air)
13. Element13(Air)
14. Element14(Air)
15. Element15(Air)
16. Element16(Air)
17. Element17(Air)
18. Element18(Air)
19. Element19(Air)
20. Element20(Air)
21. Element21(Air)
22. Element22(Air)
23. Element23(Air)
24. Element24(Air)
25. Element25(Aluminium)
26. Element26(Aluminium)

Add internal heat source or set a predecribe temperature on element boundaries.

Selected item

Element name: Element16

Internal heat: 0

- Ascending: 0
- Descending: 0
- Microgravity: 0
- Recovery: 0

Predescribed temperature

Temperature:

☐ North 0
☐ South 0
☐ East 0
☐ West 0
☐ Top 0
☐ Bottom 0

Figure 44. Generate load window.

The final step of the process is to visualize the output data from the simulations, which is done by choosing *Show results* in the main menu. This opens a new window, which can be seen in Figure 45. The temperature distribution in the model is displayed by interpolation of the temperature in nodes. The user can choose which sides to display and the opacity of the surfaces to get a good picture of the temperatures in the model. It is also possible for the user to plot temperature curves for every node as a function of the simulation time. In order to visualize the temperature development in the module as a function of time there is an animation-button where the user can choose the animation time-step. The output data can be saved as a txt-file enabling further data evaluation in other software.

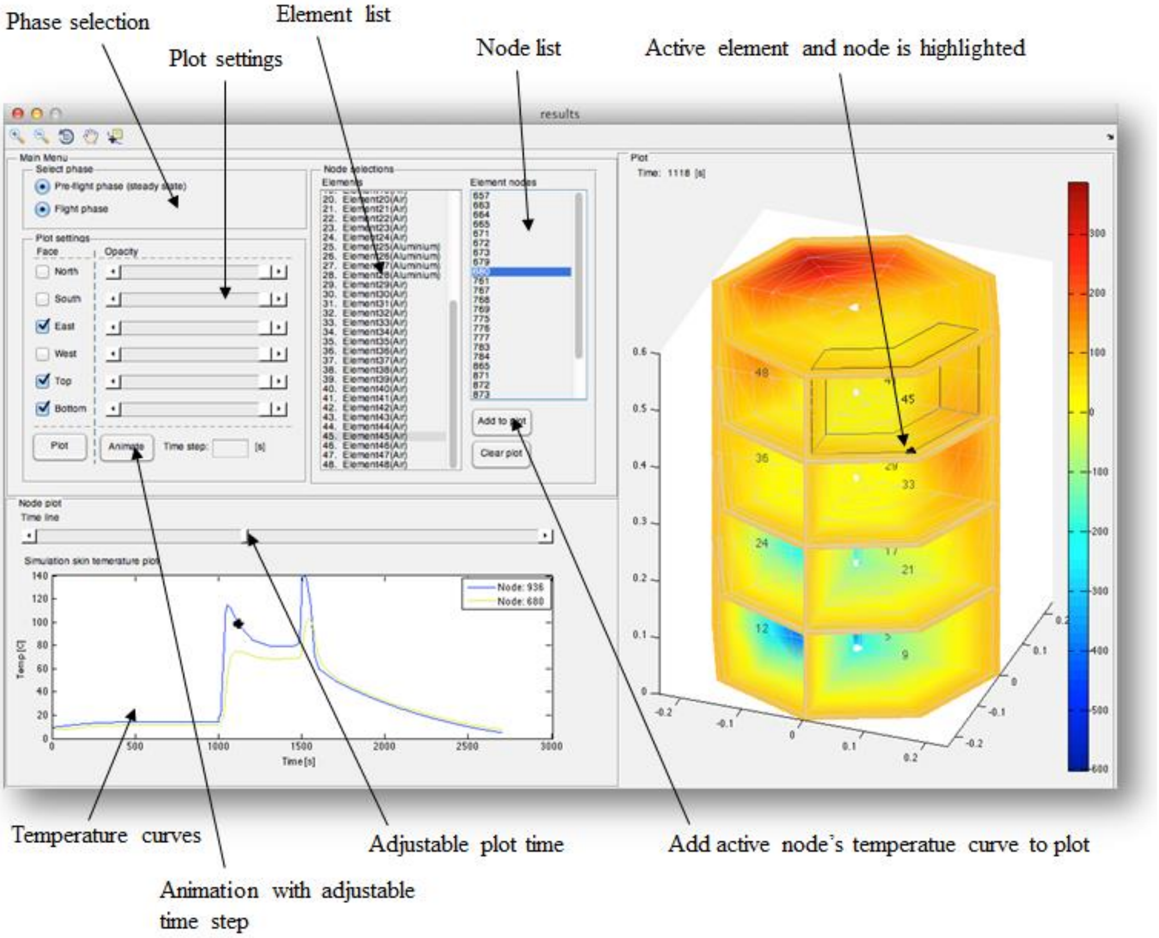
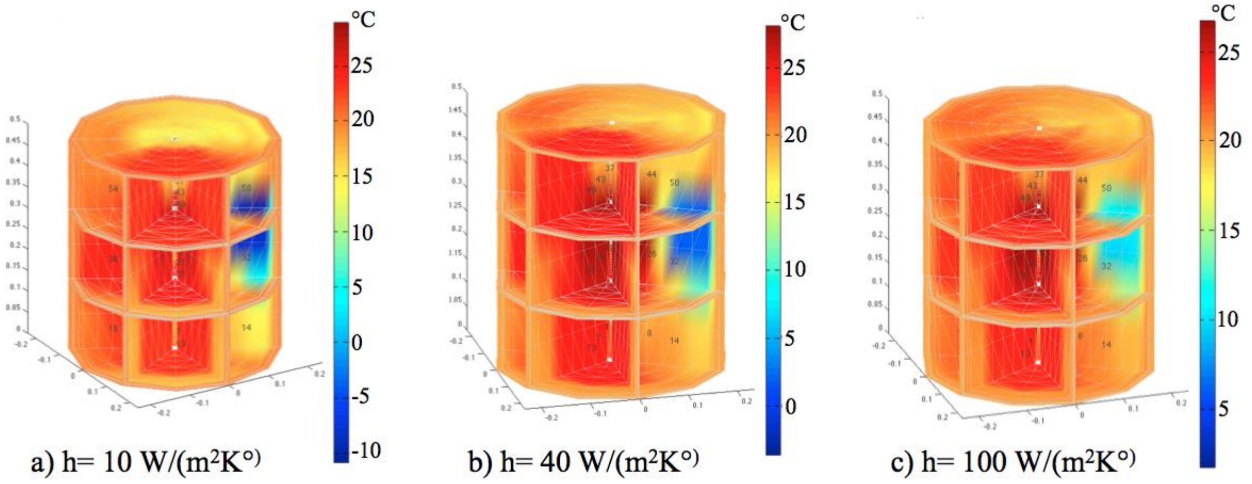


Figure 45. Results window.

4.2 Results from test module

The results from the simulation with a test module, described in *Chapter 3.5.1* are presented in this part. In the first three cases the convection coefficient of air was adjusted. The results can be seen in Figure 46. As expected, an increased convection coefficient led to a more even temperature distribution in the module. When simulating the lowest convection coefficient, the frame indicates a higher temperature compared to the other cases. By increasing the convection, heat is transferred through the air instead of through the material by conduction. This leads to higher average temperature in the elements filled with air.



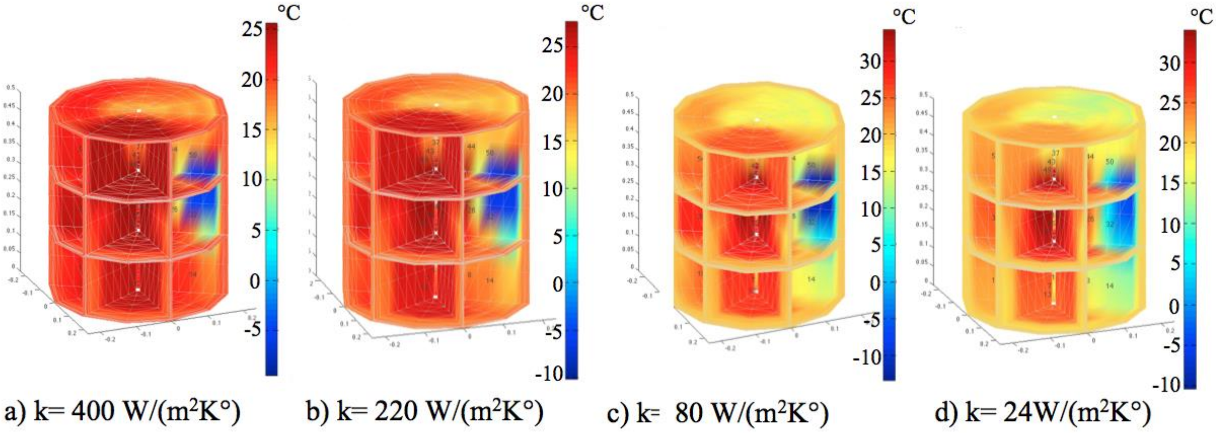
a) h= 10 W/(m²K°)

b) h= 40 W/(m²K°)

c) h= 100 W/(m²K°)

Figure 46. Results from Cases 1-3.

An increased conductivity in the frame also results in a more evenly distributed temperature. In the cases with a high frame-conductivity (k=400W/(m^2°K)) the temperature distribution between solid elements became more uniform compared to the case with high convection. This was logical since heat was transferred through solids instead of air. However, in a low conductivity value-case of the frame, as seen in Figure 47 d), the temperature in both the solids and air elements were much lower compared to other cases. This was most likely related to the fact that heat transfer through convection was reduced due to a smaller area where the surface temperature was separated from the air temperature.



a) k= 400 W/(m^{2}K°) b) k= 220 W/(m^{2}K°) c) k= 80 W/(m^{2}K°) d) k= 24W/(m^{2}K°)

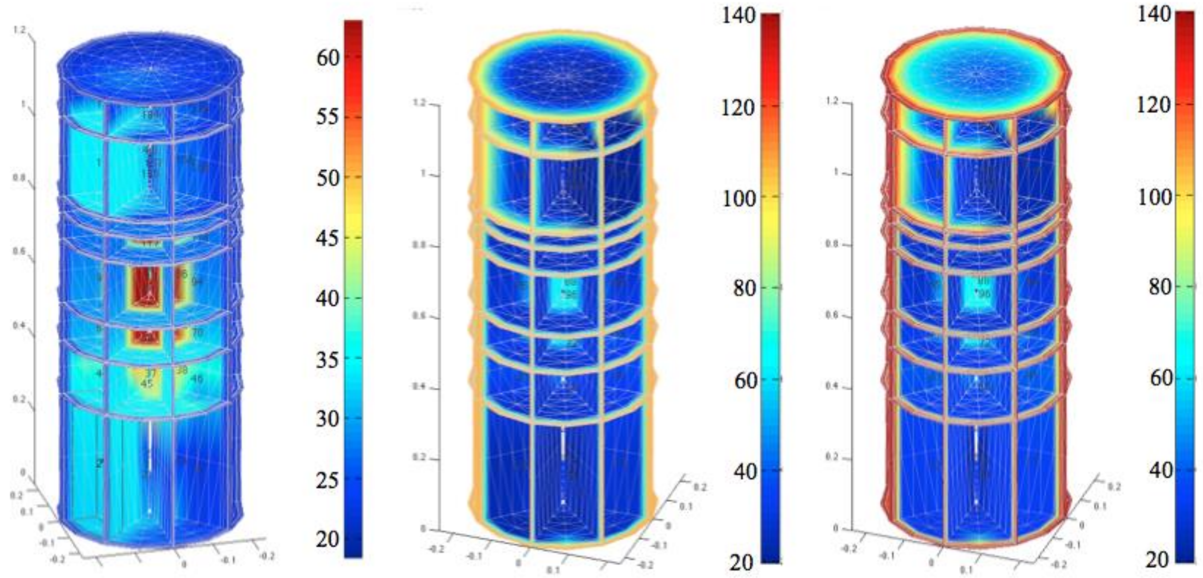
Figure 47. Results from Cases 4-7.

The results from the different cases indicate that the thermodynamic characteristics of a module are highly dependent on interior design and material. By choosing an appropriate material with corresponding properties, the balancing between the internal convection, radiation and the active cooling system can be optimized without increasing the power requirement or implementing more cooling elements. It is by combining all modes of heat transfer in a good way and avoiding sub- optimization for just one type of heat transfer, that a better module can be designed. It is recommended that this should be taken into consideration when designing the next generations of modules in order to make them more energy efficient and thermodynamically stable.

From the test module-cases it also became clear that the environment conditions have limited effect on a pressurized module. The internal generated heat has larger influence on the temperature than the aerodynamic heating during exit and re-entry of the atmosphere. The insulation and air film between the skin and the inside of the module together with the relative short flight time might be a reasonable explanation to this effect. All the results from the tested cases above were intuitive and did not generate any unexpected results. They are however a good indication that the program works as desired and verifies the correctness of the program in some aspects.

4.3 Results from XRMON-GF Simulation

In Figure 48, the results from the XRMON-GF simulation can be seen. The frame to the left (*a*) displays the temperatures at lift-off. The red colored parts in the middle are the furnace, camera and x-ray tube. The light blue area up to the left is the electronic cabinet on the battery deck. The middle frame (*b*) is when the 2nd engine is burned out at ascent. Here, it can be seen that the friction from the atmosphere has increased the temperature of the structure and skin. The frame on the right (*c*) is from the end of the descent phase when the skin of the module has it maximum temperature of around 140 °C. Note that the temperature scale is different in *a)* compared to *b)* and *c)*.



a) t=0 b) t=50 s c) t=500 s

Figure 48. Frames from the simulation of the module XRMON-GF.

The temperature in crucial components and locations as a function of time has been plotted and can be seen in Figure 49. Components located close to the furnace, generator and x-ray tube show a slightly higher temperature than the components on the battery deck, which is natural since these are close to the largest heat sources in the module. The average temperature for the whole module is between 20°C and 30°C during pre-launch. However, there is a small increase in temperature taking place during this phase, which is related to the internal convection coefficient. The air is not circulated enough for the external cooling elements to keep a constant temperature. The average power for the external element was calculated to be 180 W. However, the calculations were made without taking the power required to circulate the fluid in the cooling loop into consideration.

The plot in Figure 49 shows that there was an increase in temperatures for all components after launch. When disconnecting the umbilical cords the external temperature control is thereby removed and a temperature increase is an expected behaviour. The increase in temperature seems to level out after 500 sec, meaning that the largest increase in temperature after launch takes places in the microgravity phase. It is first when the rocket re-enters the atmosphere that the temperature trends towards its maximum.

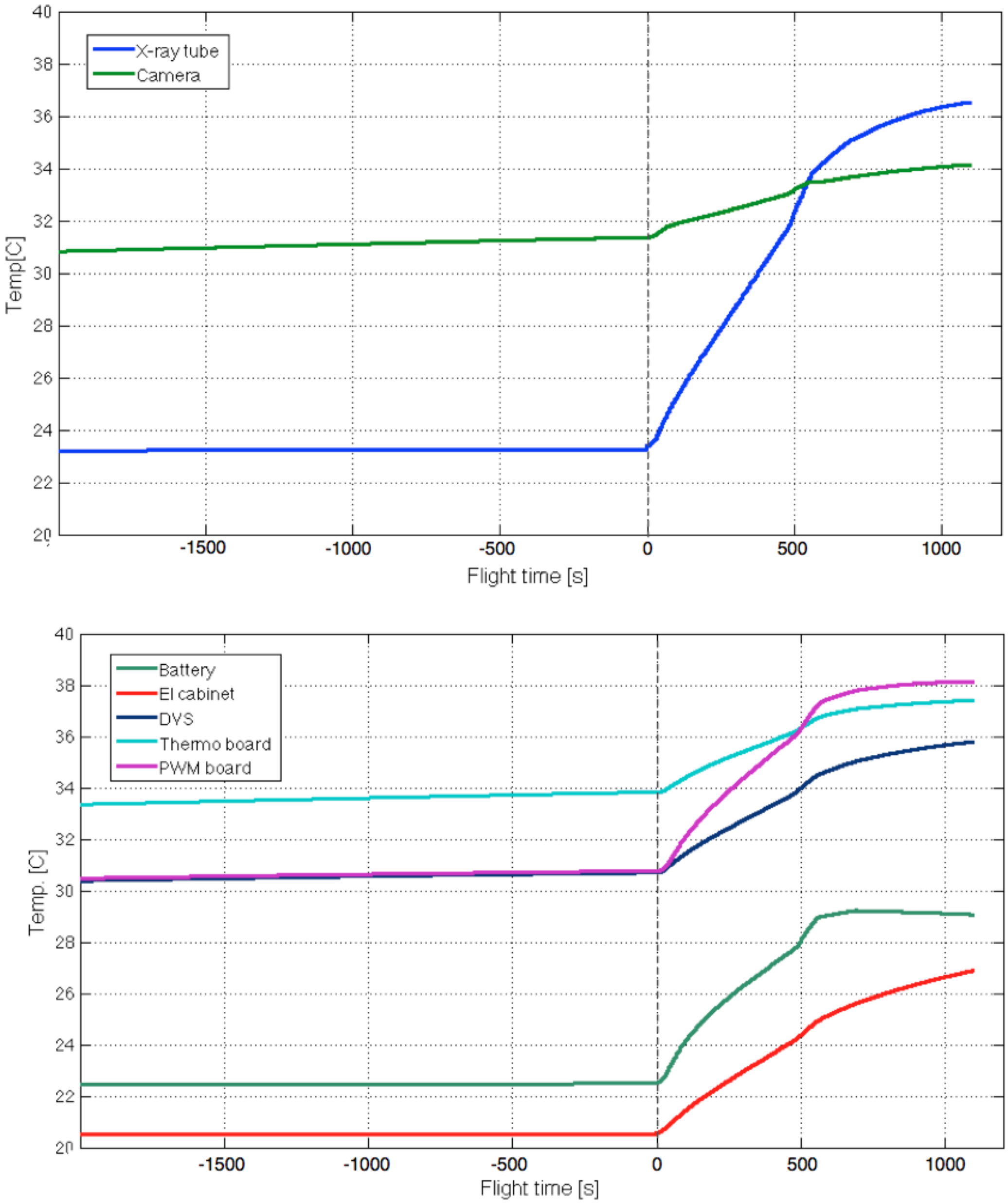


Figure 49. Temperature curves from the simulation.

In Figure 50, the measured temperatures from MASER 12 are shown. The measured curves correlate with the simulated results, both showing a temperature increase after launch due to no external temperature control. With future investigation it can be seen that the largest temperature raise in the simulated model, as well as in the measured data, occurs at the location of x-ray tube and camera.

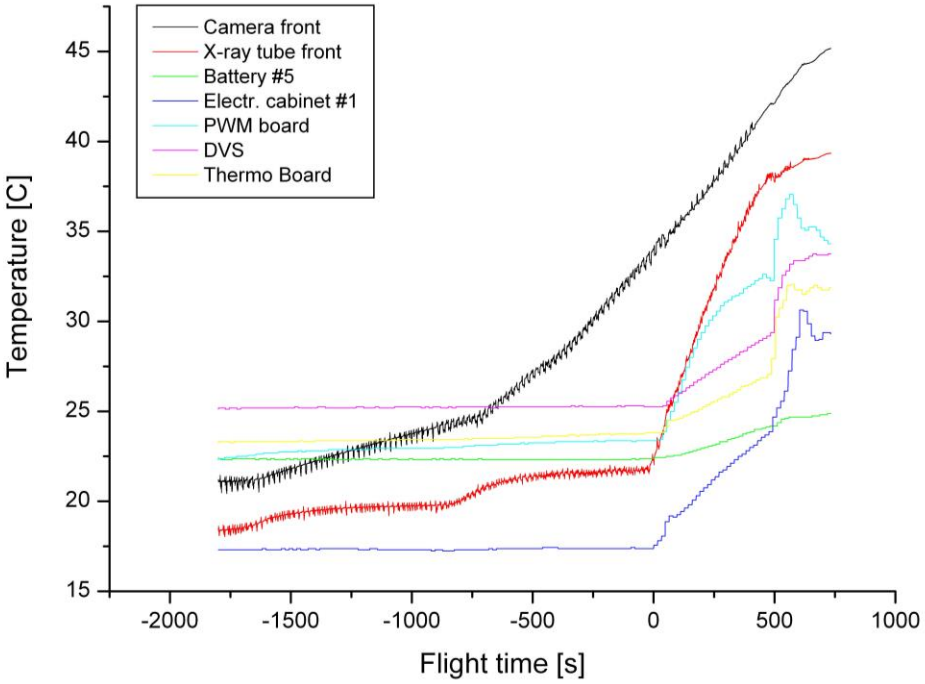


Figure 50.Measured temperatures from MASER 12.

The average temperature is higher in the simulated data compared to the measured during pre-launch but lower during flight. Another difference between the measurements and simulation can be seen in the Camera. In Figure 50, the temperature rise for the camera before launch is larger than in the simulations, see Figure 49. This can be explained by the countdown procedure for MASER 12. At *t* equal to -900 the second pre-heating of the furnace (located just beneath the camera) is started. The simulation does not take this into account but acts like the furnace is at a constant power during the entire pre-launch.

5 DISCUSSION AND CONCLUSIONS

In this chapter the results of the MSc project are discussed as well as assumptions, solving techniques and alternative methods. The conclusions from the MSc project are presented in the end of the chapter.

The computational model developed in MATLAB enables the user to easily investigate different design of a MASER payload from a thermal perspective. The output results from a completed simulation are proven to give a relatively good approximation to measured data if accurate input parameters are defined. In this section, further discussions are made regarding weaknesses, uncertainties and important aspects to take into consideration when performing simulations.

When performing the simulation of the XRMON-GF module, uncertainty in the input data might have influenced the results and the reliability of the model. First of all, the power of each component in the module was based on previous calculations from SSC and the correctness of their calculations and assumptions has not been studied in detail. Secondly, the computational model does not take air outlets from components and electrical equipment into consideration. The internal heating is assumed to be evenly distributed over the element have its maximum in the middle without any temperature peaks. It was not the purpose of the model to take such details into consideration but it should be pointed out that these features have a large impact on the results. Thirdly, there were also a few uncertainties in the material properties for some components. They were assumed to be made of one type of material and it is likely that these parts could work as thermal bridges transferring heat in an undesirable ways.

The approach for solving the internal radiation has been based on the assumption that all incoming radiation is absorbed by the elements and all wavelengths have the same impact. Due to the short time frame of this project, this was a necessary assumption in order to implement the internal radiation. Including angles, reflectivity and wavelength related behavior in the calculations would probably increase the accuracy but could take time to implement. Other essential assumptions were made concerning the skin temperature measurements. The measurements were performed by SSC between the payloads skin and the insulation layer. This could mean that the real temperatures on the outside of the module were higher than the ones used. To increase the precision, the prescribed temperature could be turned into heat flows from the outside. This was judged as not critical for the accuracy of the module since according to Goddard Space Flight Center (2005) the outside temperature does not have a considerable affect the inside. Besides that, the potential error would not disappear by changing the boundary conditions to a type of second order, since the equation would still be built on the same temperatures.

A reform that should be evaluated in detail is the effect of air rushing into an open module when re-entering the atmosphere. It is most likely that both the air temperature and velocity affects the ambient thermal conditions as it enters the inside of the module. The effect could be either cooling due to the low temperature of the air in the outer atmosphere or heating if high temperatures occur due to friction between skin and the surrounding air. Since there was no data on this phenomenon or that the effect from it was verified, it was left out from the simulations. The program can however be adjusted for this effect by just adding a load vector or increasing the air temperature during the first seconds of re-entry.

As verification, real physical tests would have been more desirable than the used method with COMSOL. In these programs there is always a chance that the mesh, loads or boundary conditions are incorrect or that the results are misleading due to the structure of the program (e.g. infinite stresses at complex geometry in stress analysis) Physical test have not been performed due to the short time frame but is highly recommended for further evaluation and verification of the computational model.

5.2 Conclusions

Following conclusions was drawn from the performed MSc thesis.

- The main objective of developing a flexible program for calculating temperatures inside a MASER module was reached. The final version of the program satisfied the requirement specification.
- The chosen method, FEM was an appropriate method for approximating the temperature in the module. For more detailed analyses it is however required to take fluid dynamics into consideration.
- Input parameters such as material properties, airflow and power are of great importance for the result. In order to produce reliable result the user need to have good knowledge about both the material and the internal convection.
- The complexity of most components makes them difficult to simulate in a large system with high accuracy.
- It is challenging to simulate radiation in an enclosed volume with several different components that each consists of different materials.
- From the performed simulations it was found that the external environment does not have significant effect on the inside temperatures for a pressurized module. The number of simulations was however too few for a conclusive statement.
- MATLAB proved to be a suitable software for developing a FEM-program due to its ability to handle large matrix systems in a time efficient way.

6 RECOMMENDATIONS AND FUTURE WORK

In this chapter, recommendations and future work are presented.

6.1 Recommendations

Since temperature is highly related to the pressure it is recommended to expand the model to take air pressure into consideration. This is especially interesting for an open module which experiences a pressure drop as they are being launched into space. The decrease in pressure would affect the inside temperature. Another motivation for implementing pressure in the program is that many components used for research and measurements are sensitive to changes in pressure.

Many components contain more than one material and have a much more complex geometry than what is possible for the developed program to handle. A solution to this problem would be to divide elements into several smaller elements, so called sub-meshing. A sub-meshing would allow the program to handle more complex geometry, handle internal generated heat in more realistic way and combine material properties in a more advanced way. It is believed that implementing this mesh type would increase the accuracy of the model.

6.2 Future Work

Following future work is recommended.

- Conduct physical tests to verify the computational model.
- Implement fluid dynamics into the module to simulate the airflow.
- A function for calculation the convection coefficient would make the program user-friendlier. By linking the convection coefficient to the power, number and size of fans it is possible to obtain an approximation of the value. The user would then only need to specify the components placement and power.
- Implement a function to enable dynamic material properties, their properties changes in accordance with temperature differences.
- Develop the internal radiation so it takes wavelength and angle of incidence into consideration. As of now it is assumed that the closest object in the line of sight for the emitting surface absorbs all radiation. In reality only a part is absorbed and the rest is reflected on the surface. This is especially important for simulations of non-pressurized modules when the radiation is the only mode of heat transfer in microgravity.
- Examine how the re-filling of air that occurs for an open module during re-entry affects the temperature.
- Implement different internal convection coefficient for different phases. Since many cooling fans are connected to the experiment equipment they are shut off after the microgravity phase. This has a significant reduction on internal air circulation and should be taken into consideration

7 REFERENCES

Bergheau, J.-M.,, Fortunier, R., *Finite Element Simulation of Heat Transfer,* 1st edition, John Wiley & Sons, New Jersey, 2008.

Bergman, T., L., Lavine, A., S., Incropera, F., P., & Devitt, D., P., *Fundamentals of Heat and Mass Transfer.* 7th edition, John Wiley & Sons, New Jersey, 2011

Blomberg, T., Heat conduction in two and three dimensions: *Computer Modeling of Building Physics Application,* Ph.D thesis, Department of Building Physics, Lund University, 1996.

Cengel, Y., A., *Fundamentals of Thermal-Fluid Science,* 3rd edition, McGraw-Hill, New York, 2008

Cifuentes, A., O., Kalbag, A., *A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysi*s, Finite Elements in Analysis and Design, Vol 12 (3-4), 1992, pp 313-318.

COMSOL, *Version 4.3b,* COMSOL Multiphysics, 2013

Dalle Lucca, E., V., *The Brazilian Sounding Rocket VSB-30: Meeting the Brazilian space Program and COPOUS objective*, Coupus, Vienna, 10-21, February, 2014.

Delmas, J., *Shape functions and points of integration of the Résumé*, Manuel de reference, Révision :4261, 2013.

Faleskog, J., *SE1025, FEM för ingejörstillämpningar*. Lecture notes distributed in SE1025 Spring 2014, Royal Institute of Technology, Stockholm.

Freitag, L., A., Olliver-Goch, C., The Effects of Mesh Quality on Solution Efficiency, Department of Mechanical Engineering – University of British Columbia, Vancouver, 2008

Gilmore, D., G., *Spacecraft Thermal Control Handbook, Volume 1: Fundamental Technologies*, 2nd edition, The Aerospace Press, California, 2002.

Goddard Space Flight Center, *Sounding rocket program handbook,* Sounding Rocket Program office, Wallop Flight Facility, Virginia, 2005.

Holman, J., P., *Heat Transfer,* 10th edition, McGraw-Hill, New York, 2010

Hutton, D., V., *Fundamentals of Finite Element Analysis*, McGraw-Hill, New York, 2004

Lewis R.W., Nithirasau P,, Seetharamu K.N., *Fundamentals of the Finite Element Method for Heat and Fluid Flow,* John Wiley & Sons, Ltd, West Sussex 2004.

Liu, G., R., Quek, S., S., The Finite Element Method, A Practical Couse, 2nd edition, Butterworth-Heinemann, Oxford, 2013

Machado, H., A., Pessoa Filho, J., B., *Aerodynamic heating at hypersonic speeds*, 19th International congress of Mechanical Engineering, Brasilia, November 5-9, 2007.

MATLAB, Version 2013a, Mathwork's

Mazzoni, J.,A., Pessoa Filho, J., B., Machado, H., A., *Aerodynamic heating on VSB-30 sounding rocket*, 18th International congress of Mechanical Engineering, Ouro Preto, November 6-11, 2005

Nellis, G., Klein, S., *The Galerkin weighted residual method,* chap. 2, Cambridge, 2008.

Nikishkov, G., *Programming Finite Elements in Java* ^{TM}, Springer-Verlag, London, 2010.

Niklasson, A., Rössner, J., Report B850: Thermal Simulation of Xrmon gf module, ÅF-technology AB, Stockholm 2011.

Pentenrieder, B., *Finite Element Solutionsof Heat Conduction Problems in Complicated 3D Geometries Using the Multigrid Method*, Ph.D thesis, Technische Universitat Munchen Fakultat fur Mathematik, 2005.

Ross Jr., R., G., Appendix B: The Geometric Factors for Earth Heating Calculations, Donabedian, M., *Spacecraft Thermal Control Handbook, Volume 2: Cryogenics*, 2nd edition, Aerospace Press, California, 2003.

Siegel, R.; J.R. Howell, *Thermal radiation heat transfer*, Hemisphere Publishing Corporation, Washington DC, 1992

SSC Space, *Science Services*, Available from: <http://www.sscspace.com>. [28 January 2014].

Swedish Space Corporation – Science Services, *MASER User Manual, Version: 2H*, Swedish Space Corporation, Stockholm, 2013.

Szalvay, V., *An Introduction to Agile Software Development*, Danube Technologies, Bellevue, Inc. 2004

Thorstenson, J., *Interview with Jimmy Thorstenson*, 2014-04-05, Solna

Törnqvist, M., *MASER, Mechanics Architecture Document (MAD) XRMON- GF*, Swedish Space Corporation, Stockholm, 2011

APPENDIX A - REQUIREMENT SPECIFICATION

Requirement Specification

-Development of thermal analysis tool for evaluating payload temperature in sounding rockets

A	Introduction
A-1	Overview

This MSc degree project aims to develop and verify a model that can be used to calculate temperatures inside experiment modules intended for the MASER rocket program. The model should offer a satisfying approximation of the inside temperature. The main aspect of the program is the ability to handle different set-ups of equipment and allow the user to adjust the elements of the module to match the intended design.

A-2	System overview

The computed model should be able to calculate the thermal effect of experiment modules inside the MASER vehicle. The system can be divided into five different thermal stages; pre-launch, ascending, microgravity, descending and transport. The model should be flexible for different types of modules/rockets. Input parameters should be able to be adjustable and the data should be presented in a user-friendly interface. Softwares avalible at SSC is MATLAB

Description

B	Assumptions
B-1	Pressure assumed to be 0 in space.
B-2	No thermal bridges between module and rocket skin
B-3	No thermal expansion
B-4	The conductivity of material is assumed to be not dependable on the temperature.
B-5	No heat generated by friction between elements
B-6	No forced convection caused by the acceleration during ascending and descending
B-7	All gases are assume to be ideal.
B-8	The generated heat from electronic componentsbase is based on their maximum power.
B-9	During exit and re-entry of the atmosphere the skin of module is assumed to experience the same heat flow. The heating is in other words independent of the skin location.
B-10	No atmosphere above 90 km altitude.

System requirements

The program should be able to handle following boundaries in the different phases

C	Thermal boundaries	Value	D/W*
C-1	Pre-launch phase		
C-1.1	*Static cases*		
	-Envoriment temperatur	-30 to 20 C	D
	-Heat generated by component	User input	D
	-External heater/cooler	User input	D
	-Pressure	101 kPa	D
C-2	Ascending phase		
C-2.1	*Static cases*		
	-Heat generated by components	User input	D
C-2.2	*Dynamic cases*		
	-Surrounding temeprature	-	D
	-Pressure	-	D
C-4	Microgravity phase		
C-4.1	*Static cases*		
	-Pressure	0 kPa	D

*"Demand" or "Wishes" (D/W)

	-Surrounding temeprature	User input	D
C-4.2	*Dynamic cases*		
	-Heat generated by component	User input	D
	-Radiation from sun	-	D
C-5	Descending phase		
C-5.1	*Dynamic cases*		
	-Surrounding temeprature	-	D
	-Heat generated by component	User input	D
C-6	Recovery phase		
C-6.1	*Static cases*		
	-Surrounding temeprature	-30 to 20 C	D
	-Pressure	101 kPa	D
C-6.2	*Dynamic cases*		
	-Heat generated by component	User input	D
	User Interface requirements		
D	**Adjustable parameters by user**	**Value/Unit**	
D-1	Build equipment inside module		
D-1.1	-Components placement	User input	D
D-1.2	-Components power	User input	D
D-1.3	-Components size	User input	D
D-1.4	-Compoents material properties	User input	D
D-1.6	-Rocket type	MASER	W
D-1.7	-Open or closed module	Open/Closed	D
D-1.8	-Step time	User input [s]	W
D-2	Pre-launch		
D-2.1	-Outside temeprature	-30 to +20 C	D
D-2.2	-External Heating/Cooling power	User input [Watts]	D
D-2.3	-Input power to components	User input [Watts]	D
D-2.4	-Pressure	101 kPa	D
D-4	Ascending		
D-4.1	-Input power to components	Watts	D
D-5	Descending		
D-5.1	-Input power to components	Watts	D
D-6	Recovery		
D-5.1	-Recovery time	Minutes	D
	Accuracy		
E	**Desired accuacy of the caluclations**		
E-1	Themodel is desired to correspond with previous measured data	-	W
E-1.1	Requiested accuracy of heat disitribution	10%	W
	Compatible Softwares		
E	**Avalible Software at SSC**		
E-1	MATLAB	-	
E-1.1	Version	R2012a	

*"Demand"or"Wishes"(D/W)

APPENDIX B – PROGRAM VERSIONS

The different program versions with their planned content are presented below:

Version βeta

- Schematic view of the structure of the program
    - Structure of MATLAB function
    - Grid/cell-structure. Type of Mesh

Version 1

- Create a 3D-dimensional brick grid
- Steady state of outside environment. 4 different temperatures on the outside (top, bottom and the surface area divided into 2; one facing the earth and one facing space)
    - Convection
    - Conduction
    - Radiation
- Internal heat source
- Verify with COMSOL
- The brick should consist of the same medium

Version 2

- Take version 1 into a dynamic environment. Create models for:
    - Pre-launch
    - Ascent
    - Microgravity
    - Descent
    - Recovery
- Create realistic rocket skin. Air element between rocket's skin and equipment.
- Verify surfaces temperatures with flight data.

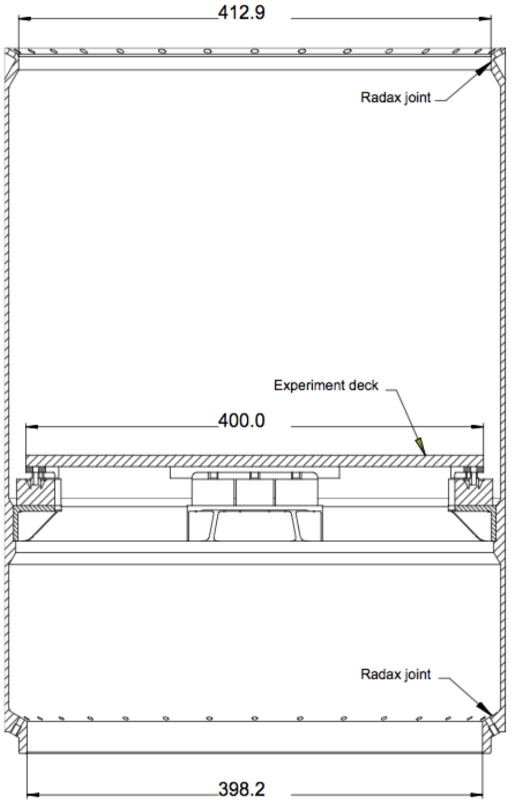
Version 3

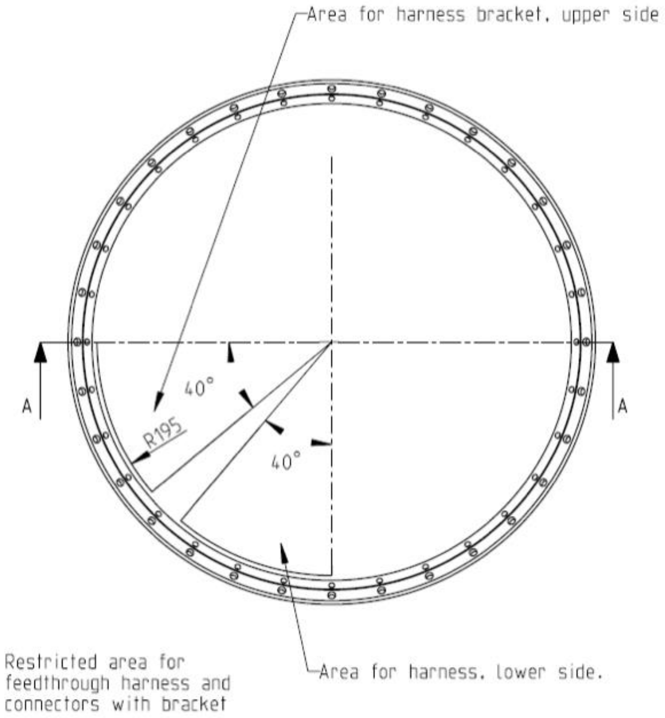
- Placement of components in different segments.
- Set power for each component
- Different attributes for different sections of the grid.
- Build specific case and verify with test data from previous flights
- Finish user interface
- User adjustable number of segments, sections and radius
- Develop user interface

Version 4 (Bonus version)

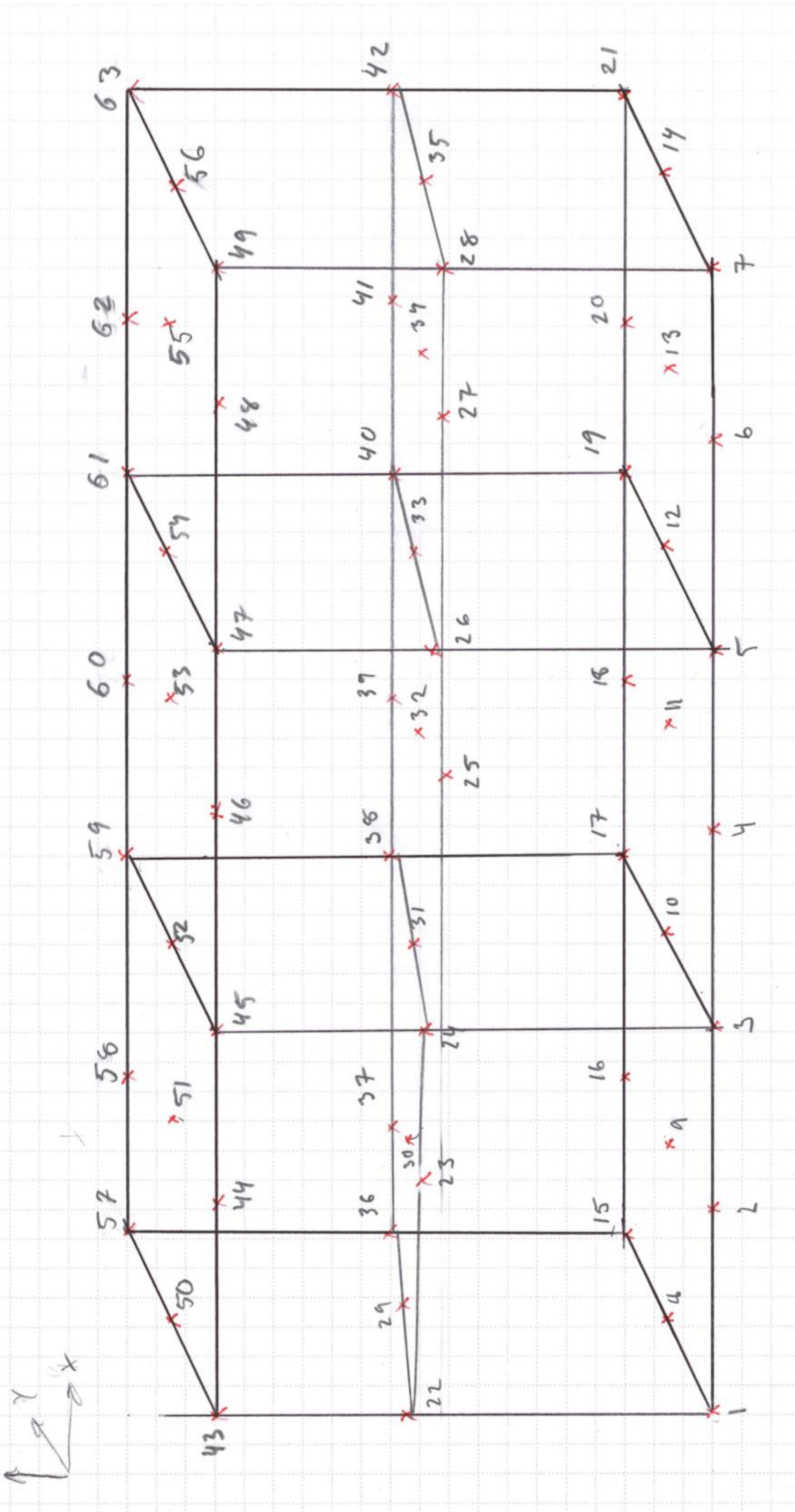
- Several modules. Take the placement of the module into consideration.
- Adjust the program for type of rocket. MASER or MAXUS

APPENDIX C – DIMENSIONS FOR MASER MODULE





APPENDIX D – MODELL *VERSION I*



APPENDIX E – SHAPE FUNCTIONS

Shape functions for a 27-node brick element (Delmas, 2013)

$$w_1=\frac{1}{8}x(x-1)\,y(y-1)\,z(z-1)$$

$$w_2=\frac{1}{8}x(x+1)\,y(y-1)\,z(z-1)$$

$$w_3=\frac{1}{8}x(x+1)\,y(y+1)\,z(z-1)$$

$$w_4=\frac{1}{8}x(x-1)\,y(y+1)\,z(z-1)$$

$$w_5=\frac{1}{8}x(x-1)\,y(y-1)\,z(z+1)$$

$$w_6=\frac{1}{8}x(x+1)\,y(y-1)\,z(z+1)$$

$$w_7=\frac{1}{8}x(x+1)\,y(y+1)\,z(z+1)$$

$$w_8=\frac{1}{8}x(x-1)\,y(y+1)\,z(z+1)$$

$$w_9=\frac{1}{4}(1-x^2)\,y(y-1)\,z(z-1)$$

$$w_{10}=\frac{1}{4}x(x+1)(1-y^2)\,z(z-1)$$

$$w_{11}=\frac{1}{4}(1-x^2)\,y(y+1)\,z(z-1)$$

$$w_{12}=\frac{1}{4}x(x-1)(1-y^2)\,z(z-1)$$

$$w_{13}=\frac{1}{4}x(x-1)\,y(y-1)(1-z^2)$$

$$w_{14}=\frac{1}{4}x(x+1)\,y(y-1)(1-z^2)$$

$$w_{15}=\frac{1}{4}x(x+1)\,y(y+1)(1-z^2)$$

$$w_{16}=\frac{1}{4}x(x-1)\,y(y+1)(1-z^2)$$

$$w_{17}=\frac{1}{4}(1-x^2)\,y(y-1)\,z(z+1)$$

$$w_{18}=\frac{1}{4}x(x+1)(1-y^2)\,z(z+1)$$

$$w_{19}=\frac{1}{4}(1-x^2)\,y(y+1)\,z(z+1)$$

$$w_{20}=\frac{1}{4}x(x-1)(1-y^2)\,z(z+1)$$

$$w_{21}=\frac{1}{2}(1-x^2)(1-y^2)\,z(z-1)$$

$$w_{22}=\frac{1}{2}(1-x^2)\,y(y-1)(1-z^2)$$

$$w_{23}=\frac{1}{2}x(x+1)(1-y^2)(1-z^2)$$

$$w_{24}=\frac{1}{2}(1-x^2)\,y(y+1)(1-z^2)$$

$$w_{25}=\frac{1}{2}x(x-1)(1-y^2)(1-z^2)$$

$$w_{26}=\frac{1}{2}(1-x^2)(1-y^2)\,z(z+1)$$

$$w_{27}=(1-x^2)(1-y^2)(1-z^2)$$

APPENDIX F – PROGRAM STRUCTURE