

OPTIMISATION OF ALUMINIUM – ALUMINIUM EPOXY RESIN BONDS

Karim Bouleghlimat - P139479



Major Project - BENG
M6X00417
1 may 2018
FAO Kelvin Lake

I. Abstract

This project investigates into the optimisation of an aluminium curing process. However, the timeline for this project has been projected as 3 years, and is broken up into 3 sections:

1. Initial Development Section (current project)
 - a) Initial development of a thermal finite element analysis software package for a specific scenario.
 - b) Further research into the bonding processes involved in curing epoxy resins, failure modes in joint design and future coding strategies.
 - c) Recommendations as to how to improve the current FEA model, and on how to proceed with this project in the future.
2. FEA Model design and production (future work)
3. Optimisation of process

In the next stages of this project, the FEA model will be further developed, and transferred to the chosen coding language for full design and manufacture. When the model is working it will be used to optimise the curing process of a chosen component, and ideally lead to the creation of an optimised set of guidelines for the process.

During this initial work, two experiments were completed, the first to begin to correlate the model to a real life scenario. It was found that both the results from the code and the results from the real life test followed similar trend lines, which is enough to show that the model has promise, and requires future developmental work. The second experiment investigated calculating the convective heat transfer coefficient, which is used to determine the heat loss the part experiences whilst being heated. The current model is capable of mapping the thermal propagation through an aluminium part, however with some limitations:

1. The model currently works for 2D
2. The model currently only works for simple geometries
3. The model currently doesn't account for heat loss
4. The model currently only works for aluminium
5. There are some problems in the mathematics derivation that need consideration

Moving forward to the next stage of this project, these are the recommendations for future work:

1. It is recommended that the model be transferred from Matlab, to a more suitable programming language, such as R.
2. It is recommended that the mathematics regarding the finite difference method be reviewed to ensure that they are fully capable of completing the task.
3. It is recommended that the code be expanded further, to account for 3D components and complex geometries.
4. It is recommended that further research is completed into current FEA software packages, and the features involved.

'This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.'

Signed:

Date:

Table of Contents

I.	Abstract.....	i
II.	List of Figures	iv
III.	List of Tables.....	iv
1.0	Introduction	1
2.0	Methodology of work	2
2.1	Identifying the problem	2
2.2	How the problem will be solved.	4
2.3	Project Timing	5
2.4	Background Research.....	6
2.41	What is adhesion/cohesion?.....	6
2.42	“A model for brittle failure in adhesive lap joints of arbitrary joint configuration”	8
2.43	“Preliminary tests of steel-to-steel adhesive joints”	9
2.44	“Functionally graded adhesive joints – A review and prospects”	10
2.45	“On the strength prediction of adhesively bonded FRP-steel joints using cohesive zone modelling”	11
2.46	Environmental research.....	12
2.5	Impact on manufacturing	13
2.51	Literature Review	13
2.6	Mathematics Derivation	15
2.61	Taylor series Definition	15
2.7	Experimental section - Experimental method	24
2.71	Experiment 1	24
2.72	Experiment 2	28
2.8	Code explanation	33
2.9	Limitations of code.....	36
2.91	Suggested improvements	38
2.10	What should be done if this project were to continue? – FEA section	38
2.101	FEA software currently used in industry.....	39
2.102	Table 6 - Comparison of FEA software in industry.....	41
2.11	What should be done if this project were to continue – Programming language section.....	44
2.111	Matlab – current language.....	44
2.112	‘R’ Programming language – suggested new language	44
3.0	Conclusion.....	47
4.0	References	48
	References – Images	49

5.0 Appendices.....	50
Appendix 1 – Experiment 1 Screenshots	50
Appendix 2 – Experiment 2 Results Table (10 seconds)	56
Appendix 3 – Experiment 2 Results Graph (10 seconds)	57
Appendix 4 – authors previous work 1	58
Appendix 5 – authors previous work 2	69
Appendix 6 – Letter from Richard Morgan	81

II. List of Figures

Figure number	Title	Page number
1	Basic project diagram	8
2	Idealised timing plan	10
3	Adhesion vs cohesion	11
4	Typical point P	25
5	Simple implicit scheme implementation	27
6	Thermal camera set up	29
7	Screenshot showing test in progress	30
8	Temperature vs time graph	31
9	Temperature at point vs time graph	32
10	Experimental set up 1	33
11	Experimental set up 2	33
12	Temp loss over time graph	36
13	ANSYS modules	45

III. List of Tables

Table number	Title	Page number
1	Experiment 1 results	26
2	Analysis of experiment 1 results	27
3	Analysis of experiment 2 results	30
4	Code explanation	33-36
5	Results from Code	37
6	Comparison of FEA software in industry	41-42

1.0 Introduction

This project investigates the optimisation of an aluminium / aluminium curing process, however, the timeline for this project has been set at 3 years, and is broken up into 3 sections:

1. Initial Development Section
2. FEA Model design and production
3. Optimisation of process

This project will complete the initial development stage of the whole project. The aims for this section of the project are as follows:

1. To begin the development of a thermal finite element analysis software package for a specific scenario.
2. To further investigate the bonding processes involved in curing epoxy resins, failure modes in joint design and future coding strategies.
3. Make recommendations as to how to improve the current FEA model, and how to proceed with this project in the future.

In the next stages of this project, the FEA model will be further developed, and transferred to the chosen coding language for full design and manufacture. When the model is working it will be used to optimise the curing process of a chosen component, and ideally lead to the creation of an optimised set of guidelines for the process.

2.0 Methodology of work

2.1 Identifying the problem

Currently in industry, many manufacturers are moving towards using epoxy glues and resins to bond materials together rather than traditional joining techniques, such as riveting and welding, due to what Hulimka and Kałuža (2017) describes as increasing demand for lightweight structures that are efficient and effective in use and in manufacture, due to global concerns regarding resource sustainability and the environment. This is especially true when it comes to aluminium bonding, as Hulimka and Kałuža (2017) explain the mechanical joining of two metals significantly weakens the connection zone of the component. Which is one of the reasons why this research into adhesively bonded steel has been completed by Hulimka and Kałuža (2017), as an alternative to mechanical bonding or welding.

Companies are starting to use epoxy glues to bond many components together, the epoxy glues on the market today can produce bonds comparable to that of traditional techniques. The problem lies in the fact that no guidelines have been established for this process, when it comes to dissimilarly sized parts. Currently in industry, if two dissimilarly sized parts need to be bonded using epoxy glues, companies heat the components above the curing temperatures, to ensure that both parts are hot enough before bonding occurs.

They do this because if they heated both parts in an oven for the same amount of time as shown below in figure 1, part 2 would reach the curing temperature first, causing the bond to form unevenly, as described by von Fraunhofer (2012), Sub optimal curing processes of the adhesive during the curing process will affect the curing reactions of both inorganic and organic adhesive materials, commonly reducing strength, bonding efficiency and the degree of cure.

This issue was further highlighted by R. Morgan (2018) as shown:

“Following our discussion on the topic, I am writing this not to confirm that recent industrial projects in which staff from the school of Engineering have been engaged, have highlighted the significant challenges many companies face when bonding structural elements. One such challenge is being able to accurately model heat transfer and dissipation in ‘heat cured’ adhesive applications.

A number of organisations have expressed interest in undertaking work to develop a thermal analysis model which would facilitate a greater understanding of their processes. It is envisaged that such models would allow process engineers to refine and optimise various process parameters, such as oven temperature, convection current speed and over cycle time.”

Please see Appendix 6 for copy of the letter.

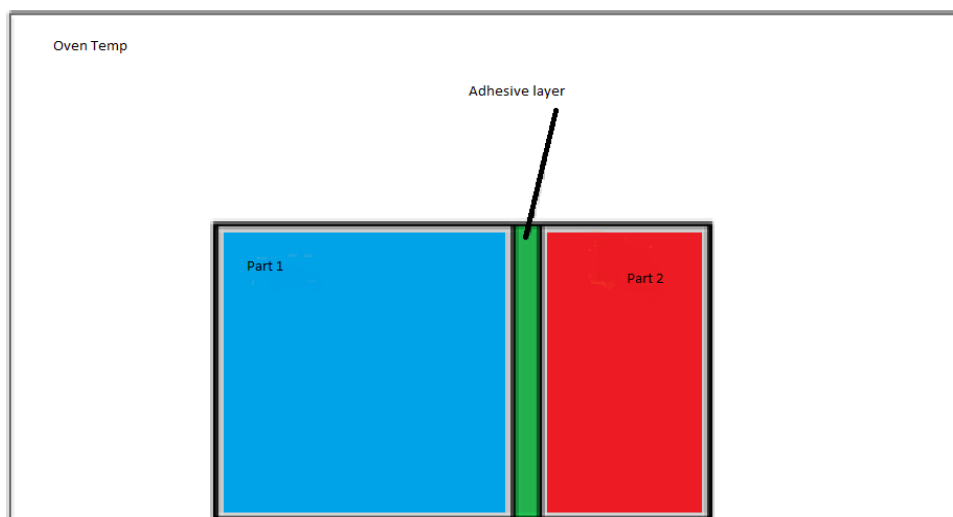


Figure 1 - Basic Project Diagram (author's own, 2018)

To solve this problem, a set of guidelines needs to be produced to define exactly how long, and at what temperatures parts need to be heated in an oven, to produce satisfactory bonds. In solving this problem, the obvious benefit is the optimisation of the curing process, however, the secondary benefits are that when the process is optimised, less energy will be wasted in the heating of the parts, and less time will be wasted also. Meaning that the company can produce more parts per unit time, while keeping wastage to a minimum.

2.2 How the problem will be solved.

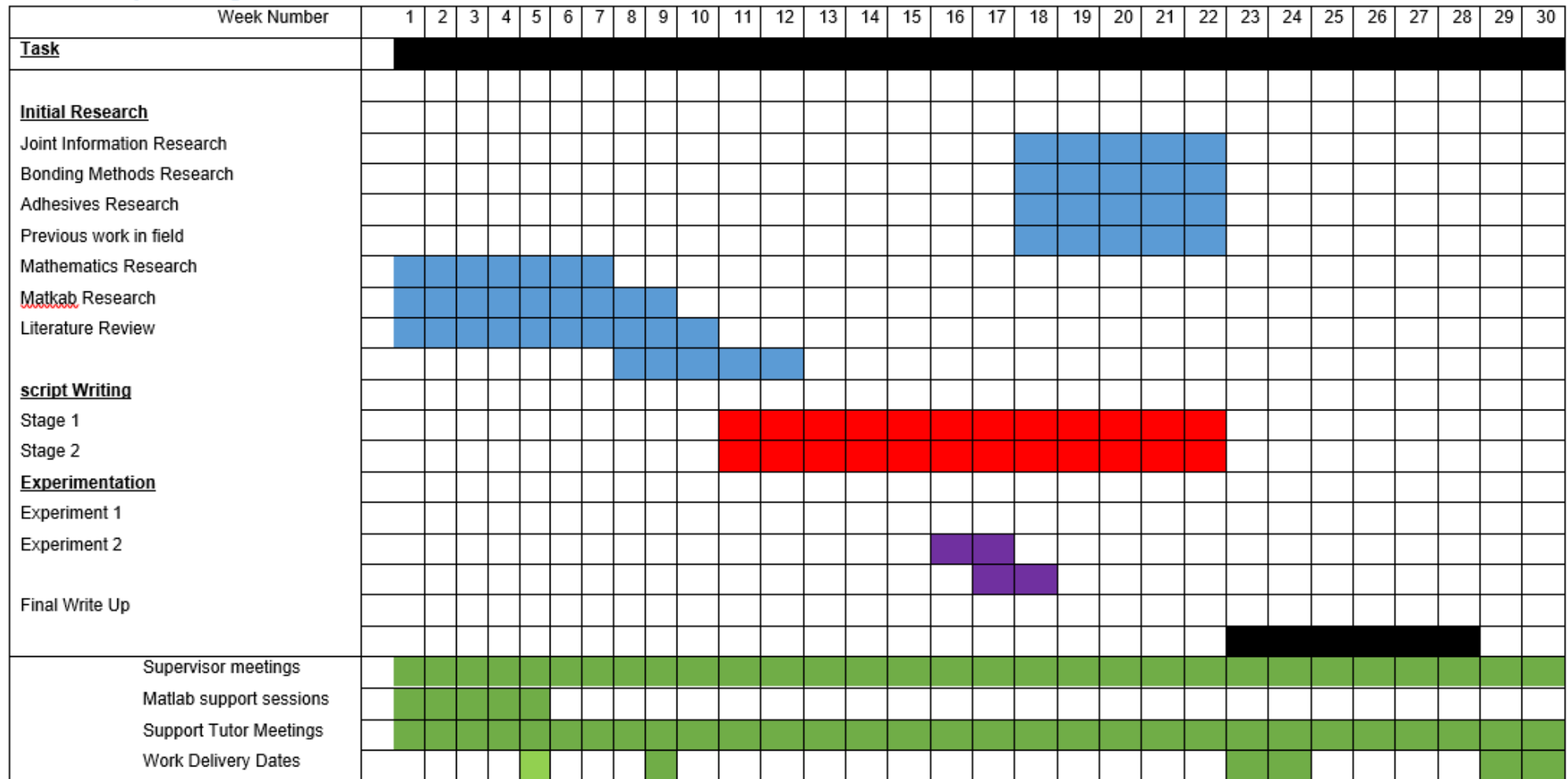
To solve this problem this project will;

- Need to find a way to map the thermal propagation through the parts accurately. This will be completed by using the Matlab software package to create a Thermal Finite element analysis software suite, as detailed in the mathematics derivation section, the script should be able to accurately map the thermal propagation through a square 2D aluminium plate, with defined conductive boundary conditions, showing accurate temperatures at each node for each time step.
- At this stage, the script will only account for heat in, therefore an experiment will be required to account for the heat out.
- Experiment 1 will be performed to establish the convective coefficient, radiative coefficient.
- Experiment 1 will be performed to begin to correlate the real life results to that of the model.
- Experiment 2 will be performed to calculate the heat loss coefficient that the part undergoes. (these values will be analysed and used during the next stage of the project)

During this project this is where the end point will be. However, if the project were to continue, there are further goals:

- Due to the limitations of MATLAB, It has been decided to change the kind of programming used if the project were to continue. Using a different programming language, the author would reproduce the model up to the above section and continue from there.
- The code currently only works for 2D, the author would like to improve the meshing techniques used. In the Matlab code it is only viable to create square domains, consisting of numbers in matrixes, however, using a more robust programming method, a meshing strategy would be produced, similar to ones used in programmes such as ANSYS Fluent, and Solidworks, to allow the user to create more complex and 3D parts.
- Once the script is complete, it would be used in conjunction with real life testing, to optimise the curing times and temperatures of a specific component set, using a specific epoxy glue.

2.3 Project Timing



See above the idealised timing plan for the entire project. As stated this is the ideal timing plan, and not the actual timing plan. Timings are expected to vary throughout the project, therefore extra time has been allotted where required in case of unforeseen circumstances.

Figure 2 – Idealised project timing plan (Author's own, 2018)

2.4 Background Research

This paper, “Adhesion and Cohesion”, while loosely looking into the same topics, is based in the dental industry, so not all factors are relevant. However, the paper consists of many informative sections on the mechanisms of adhesion and cohesion.

2.41 What is adhesion/cohesion?

Adhesion is defined as “the mechanical attraction exerted between the surfaces of bodies in contact” by the Merriam-Webster dictionary. Von Fraunhofer (2012) then goes on to further explain that adhesion is defined as any attractive process between two different molecular species. Whereas cohesion is defined as “the molecular attraction by which the particles of a body are united throughout the mass.” By the Merriam-Webster dictionary. Then goes on to further explain that cohesion is defined as the process of attraction that occurs between two or more similar molecules. This is primarily due to forces formed by chemical bonds that form between the components of the adhesive.

In simple terms, adhesion is the force that makes things stick to other things, and cohesion is known as the internal strength that holds the adhesive together. Adhesion and cohesion in combination determine the overall bonding effectiveness, in every situation involving an adhesive and a substrate.

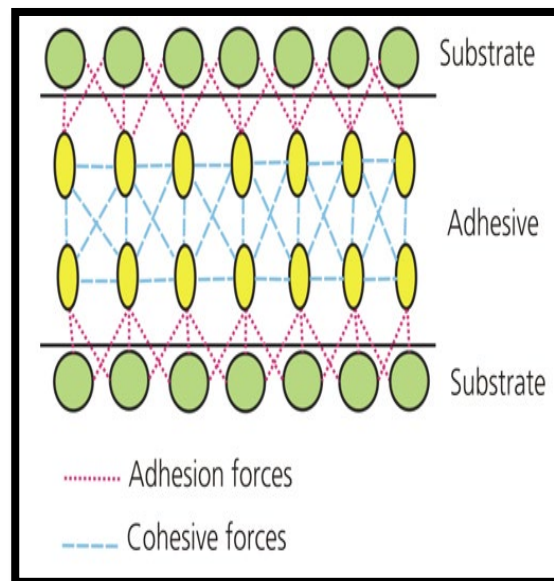


Figure 3 –Adhesion vs COhesion (taken from <http://www.adhesives.org/adhesives-sealants/science-of-adhesion/adhesion-cohesion>, 2018)

Forces involved in adhesion/cohesion?

The cohesive strength of an adhesive can be determined by a number of molecular forces, regardless of what adhesive is being used, and can be summed up in several points:

- 1) The chemical bonds within the adhesive material
- 2) Intermolecular interactions between the adhesive molecules

Depending on which adhesive is being used, different levels of chemical bonding will occur. These intermolecular interactions affect mainly the uncured adhesive, typically affecting the viscosity and flow properties of the adhesive. However, once curing has occurred, these problems are solved as the adhesive solidifies, through bonds formed between the adhesives molecules, this is done by strengthening existing bonds, and by forming new bonds. In this process typically, crosslinking will occur in short chains of molecules, and then will form into longer chains, and eventually into 3D networks of molecular chains. This is commonly used in dentistry, specifically in the setting of zinc oxide-based dental cements. Where it has been shown that suboptimal conditions in curing will cause adhesive to lack cohesive strength von Fraunhofer (2012) Describes sub optimal curing processes of the adhesive during the curing process will affect the curing reactions of both inorganic and organic adhesive materials, commonly reducing strength, bonding efficiency and the degree of cure.

When looking at adhesion, the forces involved are intermolecular forces generally. Adhesion can be further broken down into three categories;

- Chemical adhesion
- Dispersive adhesion
- Diffusive adhesion
- However, in the case of aluminium and epoxy resin, chemical bonds can form between the substrate and the adhesive, forming a strong bond.

Research studies that cover various segments of adhesive bonding:

- A model for brittle failure in adhesive lap joints of arbitrary joint configuration
- Preliminary tests of steel-to-steel adhesive joints
- Functionally graded adhesive joints – A review and prospects
- On the strength prediction of adhesively bonded FRP-steel joints using cohesive zone modelling

2.42 “A model for brittle failure in adhesive lap joints of arbitrary joint configuration”

This paper investigates efficient general finite fracture mechanic model, for crack initiation in adhesively bonded joints. Stein, et al. (2015) completed their investigation by modelling the mechanical behaviour of a bonded joint, using a sandwich-type model, with composite adherents, analysing the results. Stein, et al. (2015) describe the behaviour of failure as a physically sound combined stress and energy criterion which only needs two failure parameters:

- Strength of adhesive
- Fracture toughness of adhesive

Using experimental, numerical results, and the results of an elaborated cohesive zone model are used to compare the results. From these results Stein, et al. (2015) states that the predictions made for the failure loads are in agreement with the experimental results.

Stein, et al. (2015) go on to say that both the Cohesive Zone Model, and the real life experimentation, accurately capture the effect of changing geometric parameters, and especially the thickness of the adhesive layer of a single lap joint. Which is important as other purely stress based failure models are not able to cover the effect well enough. However, due to the complicated nature of this model, several simulations are required completed to determine the failure load predictions of specific joint designs, which ultimately is very time consuming. In comparison to simple methods such as Finite fracture mechanics that can predict the failure load of an adhesive in seconds, however, with less accuracy.

2.43 “Preliminary tests of steel-to-steel adhesive joints”

Hulimka et al. (2017) begin by highlighting the problem in this area as:

The mechanical joining of two metals (specifically steel) significantly weakens the connection zone of the component. Which is why this research into adhesively bonded steel has taken place, as an alternative to mechanical bonding or welding. The main problem this paper looks at identifying proper bonding factor providing a full utilisation of joined components.

The main aims of the research are to select the most suitable adhesive to use when joining high strength steel components. This was completed by testing 3 different adhesives, to create double lap joints, subjected to static tension. The paper shows the results from a preliminary lab test of statically loaded double-lap shear specimens, made of high-strength steel. Analysing the load bearing joint capacity, as well as the stress distribution in the adhesive layer. Ultimately this study found that consistent load bearing joint capacity was achieved, however, only for specimens with an anchorage length of 400mm, with smaller pieces the load bearing capacity wasn't reached, and sudden specimen failure occurred.

These results are not representative of all adhesives, or scenarios. Therefore, more research in this area is vital to prove Hulimka et al. (2017)'s theory. However, this study does justify further research, as well as indicating that these adhesives may be suitable for structural steel applications.

2.44 “Functionally graded adhesive joints – A review and prospects”

Durodola (2017) explains that there is an increasing demand for lightweight structures that are efficient and effective in use and in manufacture, due to global concerns regarding resource sustainability and the environment. Many sectors of industry have been adopting adhesive bonding in various assembly arrangements, rather than using mechanical fasteners, this is due in part to the relative reduction of stress concentrations in joint. This has great potential for future applications, as this use of stress reduction could be used to design tailored stress distribution as desired in adhesive layers, in many components and assemblies.

The paper covers a large section of work, from theoretical to experimental, with information from the 1960's - current. The main point being that using optimised bond line design, the strength of adhesively bonded joints can be significantly increased. This study focused on several geometries, lap joints, double lap joints, and tubular lap joints, tested using tensile loading, as well as bending and torsion. Finally the need for optimisation of the scheme has been highlighted. However, Durodola (2017) highlights that several areas in this study require further attention:

- *“Development of readily applicable optimisation methods for the design of functionally graded bonded joints rather than parametric studies is still a challenge that has not received sufficient attention. Optimisation methods will permit the use of desired objectives directly in solving Functionally graded adhesive (FGA) joint problems.”*
- *“Development of methods for the controlled dispensing of multi adhesive with rather limited or no mixing. This initiative can benefit from latest precise printing technology. Adhesive mixing can be avoided by leaving appropriately determined gaps between dispense adhesive. Computation fluid dynamics can be used to study flow patterns to determine limits. This can then be validated by experimental testing. Although some progress was made by using silicone rubber and nylon separators more attention to this issue will be helpful. Further progress on the use of robot dispensing and 3D printing have started to appear.”*
- *“Statistically based sensitivity analysis can be carried out to show the influence of realistic variabilities in the joint parameters. Some studies have been carried out on single adhesive bond layers in which it was noted that peel stresses were more sensitive to changes in modulus than shear stress.”*

• *“There is need to continue to search for new curing methodologies that will lead to desired property gradation. This will require a fresh consideration of new and existing methods such as direct heat, electric heaters, infra-red radiation, induction heating, high frequency dielectric heating, low voltage electric heating and ultrasonic activation curing methods. The work by Carbas et al. deserves a particular mention as an encouraging pointer to what can be achieved in terms of property gradation through the use of an adaptable curing process.”*

• *“Development of multi adhesive and adherend compatibility studies suitable for the analysis of functionally graded adhesive joints.”*

• *“While static gains in strength has been studied, the possible gains in structural integrity and reliability under dynamic loading has not received much attention. It is expected that remarkable gains in life under fluctuating loading would ensue.”*

2.45 “On the strength prediction of adhesively bonded FRP-steel joints using cohesive zone modelling”

This study consisted of using tools such as CZM (cohesive zone modelling), to work around the complexities of designing fibre reinforced polymer/ steel joints. The Primary focus of this investigation was, using CZM, to analyse various test configurations of joints and then complete a comprehensive numerical study. Many parameters were taken into account by Heshmati et al. (2018) such as:

- shape and type of cohesive law
- Crack path location
- Length of damage process zone
- Vibrations of adhesive
- Different failure modes, including cohesive, and interfacial debonding.
- And the effect these factors have on the strength of joints.

Heshmati et al. (2018) describes the main findings of this study were:

- Generally, the predictions produced were in agreement with the experimental results acquired, so long as all possible failure modes are taking into account simultaneously. Heshmati et al. (2018) explains that the behaviour of the Double strap joint (DLS) specimen, was more accurately predicted when the combined cohesive and interfacial failure modes were modelled, in comparison to other test samples.

- In the case of DLS, it was found that 3D modelling could predict the ultimate tensile strength (UTS) more accurately than the 2D version, with errors of 1.3% and 3.4% respectively. Heshmati et al. (2018) states that this is due to the non-uniform stress distribution in the adhesive layer of these joints. Essentially meaning that because the adhesive layer is 3D in practice, the 2D model doesn't account for stress variations in the adhesive layer.
- Heshmati et al. (2018) also notes that when using symmetrical boundary conditions, the strength was over predicted by 6%. This is deemed negligible in this specific case, however, it highlights the need for more research into this area.
- The length of overlap in the DLS joints shows that it is directly proportional to the length of the damage process zone in the adhesive layer. This can be used in the design phase of a product, to ensure that the anchorage length is accounted for, including the effects that environmental parameters such as moisture and temperature can have.

2.46 Environmental research

There is a need to address environmental issues, this study looks into bio epoxy resin as a replacement for already existing polymer materials, such as epoxy, and polyester.

Johnson et al., (2018) concludes that, through experimental studies on epoxy bio resin, that:

- The results of tensile testing show that the tensile strength of the bio resin, when compared to commercially available matrix materials, is dramatically better. This was further confirmed by completing work on a series of test specimen obtaining an average result.
- The water resistance of the bio resin is better when compared to commercially available matrix materials. Which is a large factor in the integrity of the bond as Johnson et al., (2018) describes.
- A final soil test was completed to determine the weight loss via decomposition over the course of 10 days. It was observed that a small loss of 0.001gm occurred. However, after 20-30 days the resin began to biodegrade, and it was concluded that the bio resin starts to decompose after 30 days.

2.5 Impact on manufacturing

At this stage of the project, it is difficult to accurately gauge the impact that this will have on the manufacturing processes in future. However the aims of this project with regards to this are to:

- 1) Increase efficiency, by saving power in the heating of parts.
- 2) Decreasing cycle times, by improving timings.
- 3) Optimise bonding once the model is finished.

2.51 Literature Review

From the above research it has been noted that there is an increasing demand for the design of efficient, effective and lightweight structures, this is due to the global concerns regarding the environment and resource sustainability. Currently in industry, it is common practice to join metals via mechanical fixing methods, such as riveting, bolting and welding. However, several problems occur in this area:

- Firstly when using mechanical fixing such as bolting, and riveting, the connection zone of the component is significantly weakened, due to the holes required to join.
- Secondly when welding metals, due to the high temperatures required, can cause the materials to anneal and lose some of their mechanical properties, this is especially important for aluminium, as although the welding temperatures are generally around 200°C, and the annealing temperatures are roughly 400°C, annealing can occur in concentration areas, causing soft spots in the component.

Many sectors in industry are actively searching for new ways to join components. At present the emerging method is to use epoxy bonding. Which works via adhesion and cohesion, which in simple terms, adhesion is the force that makes things stick together, and cohesion is the internal strength that holds the adhesive together. Adhesion can be determined by a number of molecular forces, the chemical bonds within the adhesive material, and the intermolecular interactions between these molecules.

The problem lies in the curing process of the epoxy resins, as sub optimal curing processes of adhesives will affect the curing of both inorganic and organic adhesives. The

above reasons are why there is research being completed into the optimisation of curing processes of epoxy resins.

Some research has already be conducted with regards to the failure of adhesive joints, where Heshmati et al. (2018) compares the results of real world testing, with the results of a Cohesive zone modelling FEA software.

Cohesive zone modelling is a finite element software package that can simulate the cohesive zone of an adhesive. This is done to aid in the design of different joint configurations using adhesives. The study took into account several factors:

- Crack path location
- Length of damage process zone
- Vibrations of adhesive
- Different failure modes, including cohesive, and interfacial debonding.
- And the effect these factors have on the strength of joints.

“On the strength prediction of adhesively bonded FRP-steel joints using cohesive zone modelling” found that generally the software predictions matched that of real life testing, the UTS of 3D models can accurately be predicted and that the length of the overlap of joints is directly proportional to the length of the damage process zone in the adhesive layer. This is important as in the later stages of the project, it may be important to use a similar software package to accurately model the cohesive zone of an adhesive, whilst also simulating the thermal propagation through the part to accurately optimise the process.

However, it was also found that sufficient research and studies into the optimisation methods of epoxy bonding hasn’t been completed, showing the need for this project to be completed. In particular more research is required into the methodologies of the curing process, in regards to:

- The temperatures being used
- The timings being used
- The type of heat source being used (direct heat, convection, infrared heating, etc.)
- The properties and applications of different adhesives used in industry
- The properties of structures and bonds whilst undergoing dynamic loading, since the studies that are being looked at consist of static testing.

2.6 Mathematics Derivation

How can the thermal diffusion equation be modelled to produce results?

The problem of how to accurately map the thermal propagation through a part can be solved by estimating the values using finite differences and specifically the Taylor and Maclaurin series', which are explained below:

This section was produced with aid from Croft et al. (1989)'s text books regarding the subject.

2.61 Taylor series Definition

The Taylor series shows that in certain cases, a function $f(x)$ can be written in the form of an infinite amount of other function, as shown:

$$f(x) = f(x_0) + f'(x_0) \cdot \frac{(x - x_0)^1}{1!} + f''(x_0) \cdot \frac{(x - x_0)^2}{2!} + f'''(x_0) \cdot \frac{(x - x_0)^3}{3!} + f^{(4)}(x_0) \cdot \frac{(x - x_0)^4}{4!} + \dots$$

And in compact form using the summation notation:

$$f(x) = \sum_{n=0}^{\infty} f^{(n)}(x_0) \cdot \frac{(x - x_0)^n}{n!}$$

The two examples of the Taylor series above where $x_0 = 0$ is known as the Maclaurin series.

2.611 Numerical differentiation methods

From this series, it is possible to approximate the first and second derivatives of a function. This is important because it uses information from functions to approximate this value for any point.

To do this, the Taylor series at point x_0 is compared to a function at a nearby point ($x_0 + h$) as shown:

$$\begin{aligned} f(x_0 + h) &= f(x_0) + h \cdot f'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{3!} f^{(3)}(x_0) \\ &+ \dots \end{aligned} \tag{1}$$

Which can then be rearranged to:

$$\frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0) + \frac{h}{2!} f''(x_0) + \frac{h^2}{3!} f^{(3)}(x_0) + \dots \tag{2}$$

And this can then be rearranged to produce the **forward difference approximation** to $f'(x_0)$ as shown:

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (3)$$

However, this method as stated in the name is only an approximation, meaning that the values at this point could be inaccurate. But the approximation gets more accurate the lower the value of h . this is because:

Let ϵ_f signify the error incurred using this forward difference approximation, then

$$\begin{aligned} \epsilon_f &\equiv f'(x_0) - \frac{f(x_0 + h) - f(x_0)}{h} \\ &= -\frac{h}{2!}f''(x_0) - \frac{h^2}{3!}f^{(3)}(x_0) \\ &\quad + \dots \end{aligned} \quad (4)$$

This approximation is in the order of $O(h)$. Meaning that the value of h is directly proportional to the value of the error.

This error can be improved, however, this can be shown by using the Taylor series to create the **backwards difference approximation**. As shown below:

$$\begin{aligned} f(x_0 - h) &= f(x_0) - h.f'(x_0) + \frac{h^2}{2!}f''(x_0) - \frac{h^3}{3!}f^{(3)}(x_0) \\ &\quad + \dots \end{aligned} \quad (5)$$

This can be rearranged to:

$$\begin{aligned} \frac{f(x_0 - h) - f(x_0)}{h} &= -f'(x_0) + \frac{h}{2!}f''(x_0) - \frac{h^2}{3!}f^{(3)}(x_0) + \dots \end{aligned} \quad (6)$$

And from here the **backward difference approximation** can be produced:

$$\begin{aligned} f'(x_0) &\approx \frac{f(x_0) - f(x_0 - h)}{h} \end{aligned} \quad (7)$$

Again here we are faced with an approximation that gets more accurate as h tends to 0.

The next step is to quantify how the error depends on h .

Inspection of the equations shows:

$$\epsilon_b \equiv f'(x_0) - \frac{f(x_0) - f(x_0 - h)}{h}$$

$$\begin{aligned}
&= \frac{h}{2!} f''(x_0) - \frac{h^2}{3!} f^{(3)}(x_0) \\
&\quad + \dots
\end{aligned} \tag{8}$$

Again, this approximation is of the order $O(h)$. Meaning it is of the same order as the forward difference approximation method. However, now both methods can be combined to produce the **Central difference approximation method**.

By subtracting equation (5) from equation (1) we see that

$$\begin{aligned}
f(x_0 + h) - f(x_0 - h) &= \left[f(x_0) + h.f'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{3!} f^{(3)}(x_0) + \dots \right] \\
&\quad - \left[f(x_0) - h.f'(x_0) + \frac{h^2}{2!} f''(x_0) - \frac{h^3}{3!} f^{(3)}(x_0) + \dots \right] \\
&= 2. \left[h.f'(x_0) + \frac{h^3}{3!} f^{(3)}(x_0) + \dots \right]
\end{aligned} \tag{9}$$

Therefore,

$$\begin{aligned}
&\frac{f(x_0 + h) - f(x_0 - h)}{2.h} \\
&= f'(x_0) + \frac{h^2}{3!} f^{(3)}(x_0) + \dots
\end{aligned} \tag{10}$$

Now the **central difference approximation can be produced:**

$$\begin{aligned}
&f'(x_0) \\
&\approx \frac{f(x_0 + h) - f(x_0 - h)}{2.h}
\end{aligned} \tag{11}$$

The error can be worked out by:

$$\begin{aligned}
\epsilon_c &\equiv f'(x_0) - \frac{f(x_0 + h) - f(x_0 - h)}{2.h} \\
&= -\frac{h^2}{3!} f^{(3)}(x_0) \\
&\quad + \dots
\end{aligned} \tag{12}$$

This shows that the central approximation method, has an error in the order of $O(h^2)$, meaning that the error when using this method in comparison to the others is much smaller.

2.612 Second Order Derivatives:

Similarly, higher order derivatives can be found by adding equation (1) and equation (5) as shown:

$$\begin{aligned}
 f(x_0 + h) + f(x_0 - h) &= \left[f(x_0) + h.f'(x_0) + \frac{h^2}{2!}f''(x_0) + \frac{h^3}{3!}f^{(3)}(x_0) + \dots \right] \\
 &\quad + \left[f(x_0) - h.f'(x_0) + \frac{h^2}{2!}f''(x_0) - \frac{h^3}{3!}f^{(3)}(x_0) + \dots \right] \\
 &= 2. \left[f(x_0) + \frac{h^2}{2!}f''(x_0) + \frac{h^4}{4!}f^{(4)}(x_0) \right. \\
 &\quad \left. + \dots \right] \tag{13}
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 &\frac{f(x_0 + h) - 2.f(x_0) + f(x_0 - h)}{h^2} \\
 &= f''(x_0) + \frac{h^2}{12}f^{(4)}(x_0) + \dots \tag{14}
 \end{aligned}$$

This produces **central difference approximation** to the 2nd derivative

$$\begin{aligned}
 &f''(x_0) \\
 &\approx \frac{f(x_0 + h) - 2.f(x_0) + f(x_0 - h)}{h^2} \tag{15}
 \end{aligned}$$

Again here the error is of order $O(h^2)$ and is given by

$$\begin{aligned}
 \epsilon_c &\equiv f''(x_0) - \frac{f(x_0 + h) - 2.f(x_0) + f(x_0 - h)}{h^2} \\
 &= -\frac{h^2}{12}f^{(4)}(x_0) + \dots \tag{16}
 \end{aligned}$$

2.613 Time-Dependent Thermal Diffusion Model using the Finite Difference Method

Conduction in solids is described by the following thermal diffusion equation

Conduction is one of the 3 modes of heat transfer, in which energy exchange happens in solids and liquids at rest, due to a temperature gradient. Conduction requires direct physical contact between the two parts. The thermal energy is transferred at the molecular, and atomic levels. Conduction in solids is produced by migration of free electrons, and by lattice vibrations. In fluids, the energy is transferred by interactions on the molecular level. Conduction in solids can be described by the following equation.

$$\nabla^2 T = \frac{1}{\alpha} \cdot \frac{\partial T}{\partial t} \quad (17)$$

Where ∇^2 is defined by $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ in three dimensions.

- t is time,
- T is the temperature,
- $\alpha = k/\rho \cdot c$ is the thermal diffusivity [$\text{m}^2 \cdot \text{s}^{-1}$],
- k is the thermal conductivity [$\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$],
- ρ the density [$\text{kg} \cdot \text{m}^{-3}$], and
- c The specific heat [$\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$].

If there is an internal heat source Q in the problem under consideration, the thermal diffusion equation (17) becomes

If there is an internal heat source present (Q) the thermal diffusion equation becomes:

$$\begin{aligned} k \cdot \nabla^2 T + Q \\ = \rho \cdot c \cdot \frac{\partial T}{\partial t} \end{aligned} \quad (18)$$

Below is a diagram showing the method of notation used, where the **grid spacing** is $h = \Delta x = \Delta y = \Delta z$.

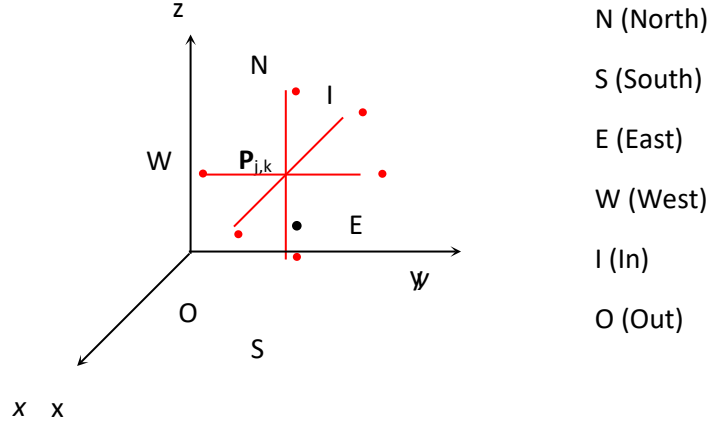


Figure 4 Typical point P

The basic idea behind the finite difference method is to substitute derivatives, in the above differential equations, by using the forward, central and backwards differences, according to the Taylor series expansion. From this we achieve a set of algebraic equations, that are solvable, providing that initial and boundary conditions are applied.

2.614 Explicit Method

By using the central difference method, equation (18) can be represented by discretising $\nabla^2 T$, and by using the forward difference method, to discretise $\partial T / \partial t$.

At point P

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} = \frac{T_O^n - 2.T_P^n + T_I^n}{h^2} \\ \frac{\partial^2 T}{\partial y^2} = \frac{T_E^n - 2.T_P^n + T_W^n}{h^2} \\ \frac{\partial^2 T}{\partial z^2} = \frac{T_N^n - 2.T_P^n + T_S^n}{h^2} \end{cases} \quad (19)$$

And,

$$\frac{\partial T}{\partial t} = \frac{T_P^{n+1} - T_P^n}{\Delta t} \quad (20)$$

Using the notation $T_P^n = T_P(t)$ temperature is evaluated at time-level t and location P ,

and $T_P^{n+1} = T_P(t + \Delta t)$ temperature is evaluated at time-level $t +$ time step Δt and location P .

By substituting equations (19) & (20) into the thermal diffusion equation (18) shows:

$$\frac{1}{F_0} T_P^{n+1} = T_O^n + T_I^n + T_E^n + T_W^n + T_N^n + T_S^n - \left(6 - \frac{1}{F_0}\right) \cdot T_P^n + \frac{h^2 \cdot Q}{k} \quad (21)$$

Where F_0 is the **Fourier number**, a dimensionless expression.

$$F_0 = \frac{\alpha \cdot \Delta t}{h^2} \quad (22)$$

The representation of the final difference method shown in equation (21), is known as the explicit method. This is due to the unknown temperature (T_P') can be determined explicitly by knowing the temperatures in the previous time steps. However, this method has a large disadvantage, in that the values for α and h are set values, this means that there are limitations to the maximum time step that can be used Δt . which should not exceed the value of the stability criteria as shown

$$0 < F_0 \leq \frac{1}{6} \quad \text{in 3D} \quad (23)$$

For the given values of α and h . the time step cannot exceed the limit of Δt that are imposed by the stability criterion. If exceeded the numerical calculations will become unstable, amplifying the error terms, and ultimately making the results inaccurate.

Similarly derivation can be done for a 2D scenario:

$$\frac{1}{F_0} T_P^{n+1} = T_O^n + T_I^n + T_E^n + T_W^n - \left(4 - \frac{1}{F_0}\right) \cdot T_P^n + \frac{h^2 \cdot Q}{k} \quad (24)$$

Producing the stability criterion:

$$0 < F_0 \leq \frac{1}{4} \quad \text{in 2D} \quad (25)$$

2.615 Implicit Method

When using the explicit method, the computation is fairly simple. However, the user is held back by the stability criterion, which becomes an issue for calculations being performed over large periods of time. This is due to the restriction of the stability criterion means that Δt tends to be small, leading the number of calculations required to be excessively large. The implicit method was developed to help to work around this limitation. Below is equation (18) with the implicit scheme:

$$-\frac{1}{F_0}T_P^n = T_O^{n+1} + T_I^{n+1} + T_E^{n+1} + T_W^{n+1} + T_N^{n+1} + T_S^{n+1} - \left(6 + \frac{1}{F_0}\right) \cdot T_P^{n+1} + \frac{h^2 \cdot Q'}{k} \quad (26)$$

This equation is unconditionally stable. This scheme is known as implicit, because at each time step, the nodal temperatures at the future time steps are required for the points about point P. this contrasts to the explicit method (21) where only the nodal temperatures at past time steps are required. This creates the requirement for a series of simultaneous equations to be calculated for the entire 3D grid, to determine the nodal temperatures for the next time step.

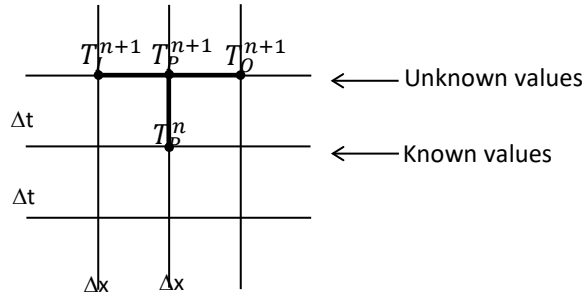


Figure 5 Simple implicit scheme implementation.

The figure above shows how the implementation of a simple implicit scheme would look in one dimension. The temperatures at the next time step (T_P^{n+1}) cannot be directly worked out. For each unknown temperature, the model must produce an equation, and solve it simultaneously with all other nodes of unknown temperature for that timestep. Due to this, this method can be very computationally heavy, however, it eliminates the limitations of the stability criterion.

2.616 Boundary Conditions

By using a grid system that covers the domain of the solution, whilst using boundaries that match the gridlines. A finite difference equation must be completed at every node of the model, which consists of all boundaries and internal conditions. Two cases often appear in heat transfer problems:

- 1) Boundary surface temperature is known (**Dirichlet** known boundary condition), in which case

$$T_B = T_G$$

Where T_G is the known temperature, and T_B would be an unknown.

- 2) Boundary surface temperature is not known (**Neumann** derivative boundary condition)

In this situation, there is an additional problem in the outward normal gradient temperature:

$$\frac{\partial T}{\partial n} = K \quad (27)$$

K is simply a given value;

$$\frac{\partial T}{\partial n} = -\frac{h}{k}(T_B - T_A) \quad (\text{convective}) \quad (28)$$

Where:

h is the Heat transfer coefficient

k is conductivity and

T_B and T_A are boundary and ambient temperatures

2.7 Experimental section - Experimental method

This section consists of two experiments;

- The first experiment will be used to begin to correlate the FEA model's answers to real life answers.
- Secondly an experiment to establish the heat loss that a component experiences, this is because the original formulas used do not account for heat out, giving false answers.

2.71 Experiment 1

2.711 Reasoning behind the experiment

This experiment is required as in its current state, the FEA model is purely theoretical. It has no real use as nothing has been done to prove that the values it produces are correct. By completing this experiment, it can prove that the values produced tend towards real world values, beginning the validation of the FEA model.

2.712 Equipment required:

- Heat source (Bunsen burner or heating element)
- Thermometer
- Thermography camera and computer with software]
- Matt black test piece
- Bench vice
- All required PPE



*Figure 6 – Thermal Camera set up
(Author's own, 2018)*

2.713 Method

- Set up thermography camera on a desk, adjust the focal length.
- Set up the experimental area on the desk opposite, with a gap of roughly 2m between it and the camera.
- Clamp the test piece in vice, ensuring that it is within the cameras view.
- Add heat source, in this case a flame was used, and measured to produce a temperature of roughly 100 degrees °C.
- Start the camera's recording feature (previously set up to record for 30 seconds, with a frequency of 1 fps, and a 3 designated test area as pictured)
- Once finished, the results were collated and exported to excel.
- Once this is completed, compare it to the results taken from the FEA model.

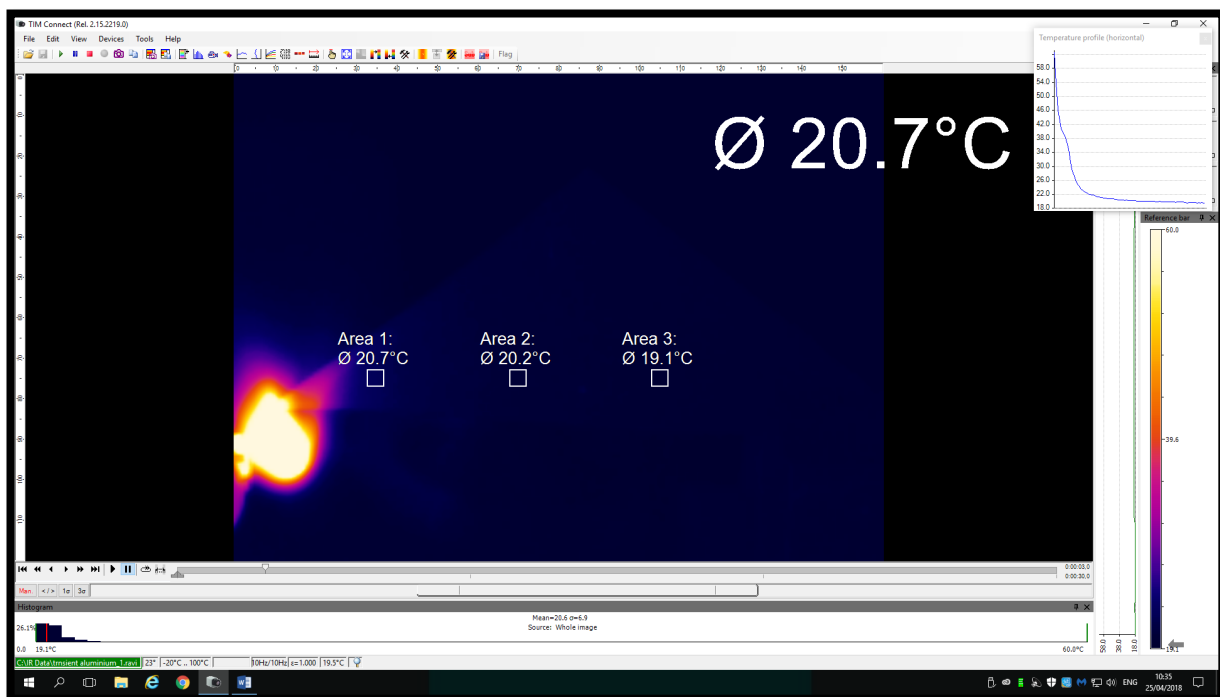


Figure 7 – Screenshot of the test in progress

2.714 Analysis of experiment 1 results

Table 1 – Experiment 1 results

Time	Area 1 Temp. (°C)	Area 2 Temp.	Area 3 Temp.	Area 4 Temp.
0	20.7	20.7	19.6	100
3	20.7	20.2	19.1	100
6	21.8	20.7	19.2	100
9	23.5	21.8	19.2	100
12	24.8	23.2	19.3	100
15	26.4	24.5	19.3	100
18	28.3	26.1	19.4	100
21	30	28	19.3	100
24	29.7	29.9	19.5	100
27	30.4	31	19.6	100
30	30.5	31.7	19.7	100

The purpose of this experiment was to obtain real life values of heat transfer through the part, and to compare them with analytical values from the FEA model. As shown in the graph and table below, the increase in temperature from the constant heat source is fairly linear throughout the experiment, until a plateau is reached, suggesting that as long as a similar increase is seen in the FEA model that the underlying mathematics defining the temperature changes are correct. Further justifying taking this project to its next stages.

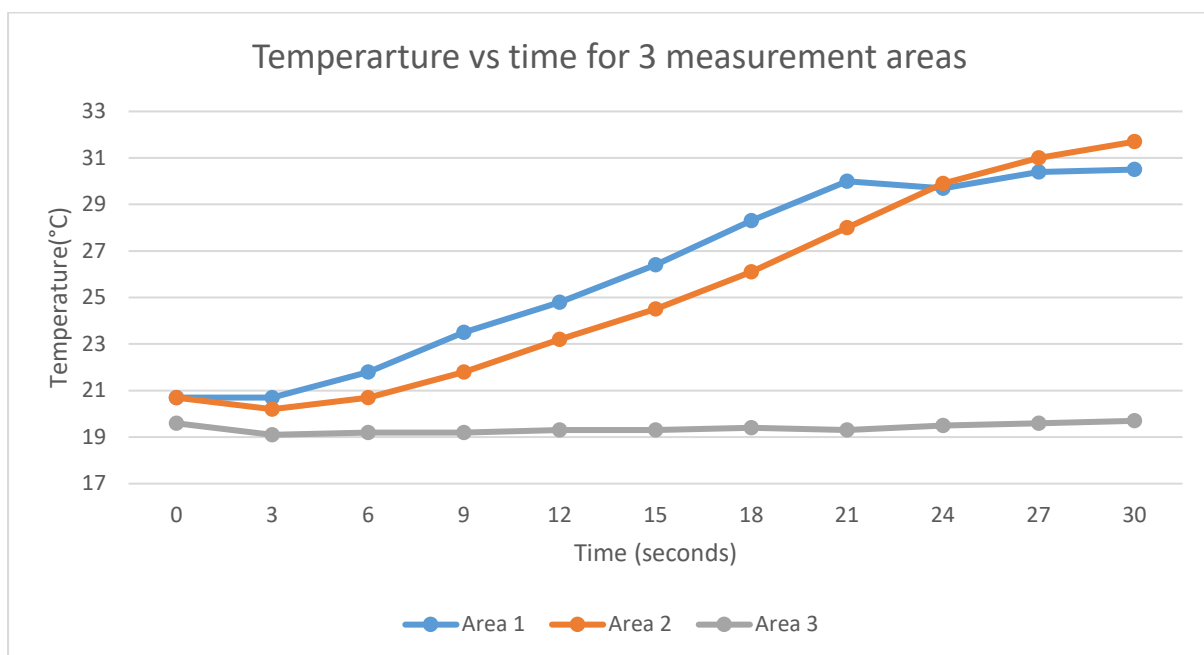


Figure 7 – Temperature Vs Time Graph (Author's Own, 2018)

2.715 Analysis of code results

As shown in the graph and table below, the heat propagation through the part follows a similar relationship to that of the graph produced in experiment 1. It begins with the temperature increasing linearly to a point and then it begins to plateau. This suggests that the underlying mathematics in the code is working as intended. The results were obtained by selecting a point in the central domain, then graphing the temperature gain per time step.

Table 2 – Analysis of experiment 1 results

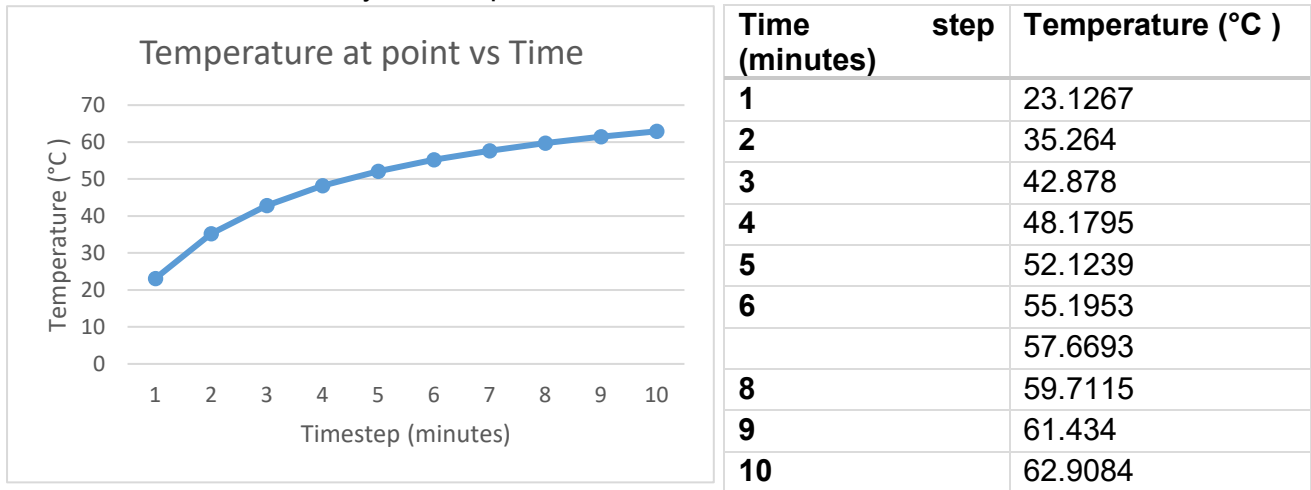


Figure 8 – temperature at point vs time graph(Author's own, 2018)

However, due to limitations in the testing methods, the results cannot be fully correlated.

As explained below:

- The timing differs in the two tests, as the real life test only lasts 30 seconds, whereas the FEA models time step is 1 minute, and the test lasts for 10 minutes.
- The size of the plate is different, in real life the plate is small, smaller than 200mmx200mmx2mm, whereas the plate used in the code is 10m x 10m.
- The heat source used in real life was a flame, whereas the heat source in the FEA model is a Gaussian distribution which crudely mimics a flame.
- The real life model is 3D whereas the code is 2D, meaning that the actual heat loss in real life will be much more substantial than in the code, as the code only accounts for the boundary around the edge of the domain. Also in reality the whole surface area of the part will be the boundary.

Overall the author is happy with the outcome and conclusions drawn from these tests, however, extensive work is required in future to further prove the theory of the project.

2.72 Experiment 2

2.721 Reasoning behind experiment.

This experiment is required because the initial FEA model doesn't account for heat loss in anyway, this it gives accurate readings for the amount of heat transfer, however, the values for actual temperature are inaccurate. To account for this, the experiment will be performed to calculate the values for the emissivity of the material, using the amount of energy lost through convection and radiation can be calculated, and this can be converted to a coefficient and introduced into the FEA model.

2.722 Equipment required:

- Bowl filled with water at above room temperature (temperature doesn't matter, as long as it's above room temp.)
- Thermometer
- Thermography camera and computer with software
- Matt black test piece
- Cloth to dry test piece
- Experimental area (consisting of two black carbon fibre plates)
- All required PPE

2.723 Method

- Set up thermography camera on a desk, adjust the focal length.
- Set up the experimental area on the desk opposite, with a gap of roughly 2m between it and the camera.
- Place the test piece in the 80°C water for 2 minutes, to ensure that the whole piece is heated.
- Remove the test piece from the water, dry it off to remove the shine from the water, and place it in the test area.
- Start the camera's recording feature (previously set up to record for 1 minute, with a frequency of 100 fps, and a designated test area as pictured)
- Once finished, the software produced a graph of the results, and the data was then exported to Excel to be analysed.

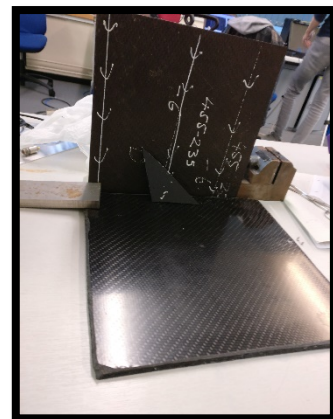


Figure 9 – experimental set up 11 (Author's own. 2018)



Figure 10 – experimental set up 12 (Author's own. 2018)

2.724 Analysis of experiment 2 results

The purpose of this analysis is to calculate the value for the overall heat transfer coefficient (h).

This was done using the following formulae:

- *Total heat loss (Q)(J) = Mass (kg) * Cp (J.K) * Change in temperature (* c)*

- *Heat transfer coefficient (h) = $\frac{\text{Heat loss}(Q)}{\text{Change in time (seconds)} * \text{Area (m}^2\text{)} * [\text{start temp} - \text{ambient temp}](^{\circ}\text{C})}$*

This was calculated for the full 1 minute of testing, and in steps of 10 seconds. Please see and Results table (3) on overleaf:

Table 3 – Analysis of experiment 2 results

time start	time end	change in time	Temp start	Temp end	change in temp	Ambient temp	Ts - TA	Area (m²)	mass (kg)	C p	Qlost (J)	h
0	10	10	46	43.5	2.5	22	24	0.004995	0.0176	900	39.6	33.03303
10	20	10	43.5	41.5	2	22	21.5	0.004995	0.0176	900	31.68	29.49927
20	30	10	41.5	40.1	1.4	22	19.5	0.004995	0.0176	900	22.176	22.76738
30	40	10	40.1	38.9	1.2	22	18.1	0.004995	0.0176	900	19.008	21.02434
40	50	10	38.9	37.9	1	22	16.9	0.004995	0.0176	900	15.84	18.76433
50	60	10	37.9	37	0.9	22	15.9	0.004995	0.0176	900	14.256	17.95003
0	60	60	46	37	9	22	24	0.004995	0.0176	900	142.56	19.81982

As expected the highest rate of heat loss occurs within the first 10 seconds of testing, with a value of 39.6J lost.

- The rate of heat loss gradually slows down to the lowest value recorded between 50 and 60 seconds of 14.256J loss.
- This rate of heat loss produces an exponential loss curve, as shown below in figure (11) Please see appendices for full results.

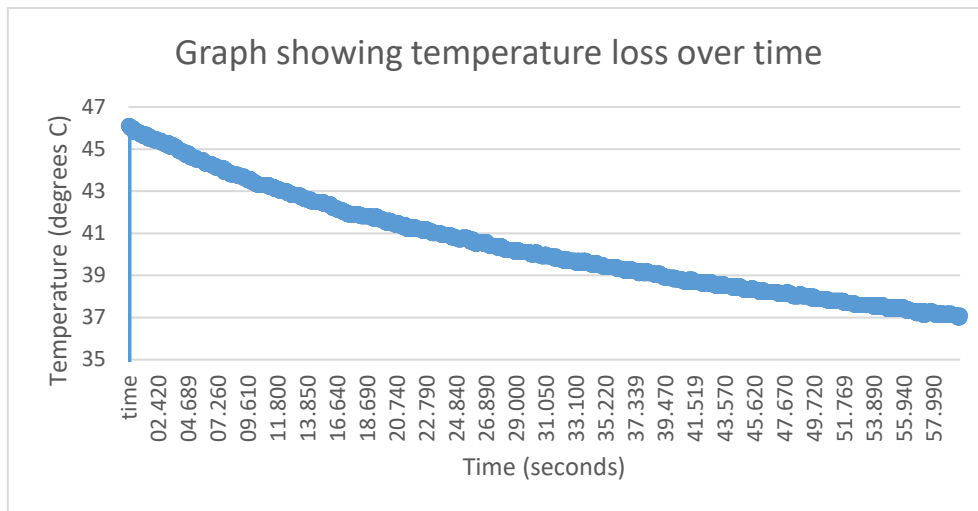


Figure 11 – Graph showing temperature loss over time (Author's own, 2018)

The values of 'h' vary from 33.03303 to 17.95003, which is within reasonable values. This value varies due to the nature of the heat transfer coefficient, as it depends on the values used for Q, change in time, and most importantly it varies due to the difference in temperature at the start and the ambient temperature. Because of this the chosen value to be used for 'h' is the reading from the full minute of testing, with a value of 19.81982, as the 'h' value for the entire minute simulates several phases of heat loss, not just the section with the largest or smallest heat loss.

However, this value of 'h' is the overall heat transfer coefficient, so it includes the convection coefficient and radiation coefficient as shown:

$$\begin{aligned} \text{Overall heat transfer coef. (h)} \\ = \text{Convective heat transfer coef.} + \text{Radiation heat transfer coef.} \end{aligned}$$

To work out the Convective heat transfer coefficient it is necessary to work out the radioactive heat transfer coefficient, and then take it from the overall.

This can be done by:

- $Q_{lost\ Radiation}(J) = Area(m^2) * Emissivity * SBC * (T_{start}^4 - T_{amb}^4) * change\ in\ Temp(*\ c)$

Where,

- Emissivity is assumed to be 0.95 (matt black test piece)
- SBC = Stefan Boltzmann constant = 5.6703×10^{-8}

However, it was found that the value for radiation is negligible, therefore it will be ignored, and it will be assumed that 'h' = Convective Heat Transfer Coefficient.

2.725 How to use h value in code?

To incorporate this 'h' value to the code, it must be used for the correct application. For this project it needs to be a boundary condition, as convection only occurs along the boundaries of the plate, and not in the central domain. Due to this, it is a simple case of referring back to the section regarding Boundary conditions (section 2.616) where it states:

$$\frac{\partial T}{\partial n} = -\frac{h}{k}(T_B - T_A)$$

This is the equation used to add convection to the boundary conditions, where:

h = Convective heat transfer coefficient (W/m².K)

k = thermal conductivity of the material (W/m.2.K)

T_B= Boundary temperature (°C)

T_A= Ambient temperature or temperature of surrounding medium (°C)

However, it was decided that this section would be saved for use in the future development of the code, although this would bring the test values closer to real life values, due to a number of factors it still wouldn't be a reliable result. Therefore, the correlation of the code cannot be completed until the project has progressed further.

2.8 Code explanation

Table 4 – Code explanation

Line Number	Line of code	Description
1	clc	
2	clear all	
3	%Given values	
4	conductivity_K=143;	Thermal conductivity (K) = 143
5	Density=2.8*10^3;	Density (roe) = 2.8*10^3 Kg/m2
6	SpecificHeat=795;	Specific heat capacity = 795
7	Alpha = conductivity_K/(SpecificHeat*Density);	Alpha having the value of 6.4241e-5
8	%Dimplate x and y	<u>Dimplate x and y</u>
9	Dimplate_x = input('Please enter X dimension in metres ');	This is an input that requests that the user enter a value for the X dimension.
10	while Dimplate_x<=0	This is a verification step, stopping users from giving values of less than 0
11	Dimplate_x= input ('Please enter positive X dimension in metres ');	this is a verification step, stopping the user from using negative values of x
12	end	End
13	Dimplate_y = input('Please enter Y dimension in metres ');	This is an input that requests that the user enter a value for the Y dimension.
14	while Dimplate_y<=0	This is a verification step, stopping users from giving it a value of less than 0
15	Dimplate_y= input ('Please enter positive Y dimension in metres ');	This is a verification step, stopping the user from using negative values of Y
16	end	End
17	%Time step	<u>Time step</u>
18	TS = input('Please enter time step in seconds ');	This is an input that requests that the user enter a value for the time step
19	while TS <=0	This is a verification step, stopping the user from giving values of less than 0
20	TS= input ('Please enter positive time step in seconds ');	This is a verification step, stopping the users from giving a negative value of time step
21	end	End
22	%Node spacing	<u>Node spacing</u>
23	h = input('Please enter node spacing in metres ');	This is an input that requests that the user enter a value for the node spacing
24	while h<=0	This is a verification step, stopping the user from giving values of less than 0
25	h= input ('Please enter positive node spacing in metres ');	This is a verification step, stopping the users from giving a negative value of node spacing
26	end	End
27	%remainder for nodes, must equal 0	<u>Remainder for nodes</u>
28	gr_l=rem(Dimplate_x,h);	This line works out whether the node spacing and plate dimensions are correct. By ensuring that
29	gr_w=rem(Dimplate_y,h);	There is no remainder when divided together. Both for length and width
30	while h>=Dimplate_y h>=Dimplate_x gr_l~=0 gr_w~=0	This line lays out a set of rules that the h value and remainder functions must follow

31	disp('Node spacing must comply with plate size ')	This line tells the user if they have input incorrect values.
32	h=input('Enter valid node spacing');	This line prompts the user to re-enter a value for h, if required
33	gr_l=rem(Dimplate_x,h);	See line 32 & 33
34	gr_w=rem(Dimplate_y,h);	See line 32 & 33
35	end	end
36	%Time step verification	<u>Time step Verification</u>
37	while TS > (h^2/(4*Alpha))	This line defines the time step that can be used due to limitations from the stability criteria
38	disp('Ensure time step is not greater than: ')	This will display to let the user know what sizes of time step are viable
39	disp(h^2/(4*Alpha))	This displays the value for time step
40	TS=input('Enter Valid time step ');	This line prompts the user to re-enter a value for time step if required.
41	end	End
42	%NROW/COL	<u>NROW and NCOL</u>
43	NRows= Dimplate_y/h+1;	This line works out the number of rows that the plate consists of
44	NCol= Dimplate_x/h+1;	This line works out the number of columns that the plate consists of
45	T=zeros(NRows,NCol);	This produces a matrix with the same size as the plate, populated by zeros
46	T(1,:)=50;	This defines the value for the first row of nodes
47	MidC=((Dimplate_x+2)/2);	This defines the value of Mid C
48	%Gaussian distribution	<u>Gaussian Distribution</u>
49	corner = 50;	This section, using predefined values, sets the top boundary of the plate to a Gaussian distribution
50	amplitude = 100;	Ranging from 50 - 100 degrees °C, this was done to simulate a flame as a heat source.
51	Sigma=sqrt(-(1-MidC)^2)/(2*log(corner/amplitude)));	Where corner, and amplitude are the values limiting the distribution
52		Sigma is the calculation for the standard deviation of the distribution
53	for c=1:NCol	
54	T(1,c)=amplitude*exp(-(c-MidC)^2/(2*Sigma^2));	This ensures that the values are in the correct positions
55	end	end
56	T2=T;	
57	Fo=(Alpha*TS)/h^2;	This is the calculation Fourier number, which is explained further in the text. (cannot exceed 0.25 for 2D)
58	%while	
59	i=0;	This is the number of iterations that have taken place, and is given the value of 0 to begin with
60	T4=ones(NRows,NCol);	T4 and T3's function is to produce two matrices, one populated with 0s and the other with 1s, which are
61	T3=zeros(NRows,NCol);	Compared against each other as a verification step. Each is a matrix of differences, and when the
62		Differences equal = 0 equilibrium has been reached.

63	while isequal(T3,T4)==0	It begins with a while function that checks when T3 and T4 are equal, and while that is true
64	T=T2;	The code continues to run
65	for r=2:NRows	
66	for c=1:NCol	Firstly c must be more than one, c must be less than the number of columns,
67		And r must not be equal to the number of rows. If they are all true the next line will be used.
68	if c>1 && c<NCol && r~=NRows %domain where we have all 4 directions	This part implements the thermal diffusion equations for different scenarios
69	T2(r,c)=(T(r,c+1)+T(r+1,c)+T(r-1,c)+T(r,c-1)-(4-(1/Fo))*T(r,c))*Fo;	If all 4 directions are available
70	end	
71		
72	if c==1 && r~=NRows %left side mid points	
73	%T2(r,c)=(2*T(r,c+1)+2*T(r+1,c)-(4-(1/Fo))*T(r,c))*Fo;	Where only the North, south and east directions are available, so east is used twice
74	T2(r,c)=(2*T(r,c+1)+T(r+1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;	
75	end	
76	if c==1 && r==NRows %left side bottom corner	
77	T2(r,c)=(2*T(r,c+1)+2*T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;	Where only the North and east directions are available
78	end	
79		
80	if r==NRows && c~=1 && c~=NCol %bottom side midpoints	
81	T2(r,c)=(T(r,c-1)+T(r,c+1)+T(r-1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;	Where only the North, west and east directions are available
82	end	
83		
84	if c==NCol && r~=NRows %the right side mid, where west direction will be used twice	
85	T2(r,c)=(T(r,c-1)+T(r,c-1)+T(r+1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;	Where only the North, south and west directions are available, so west is used twice
86	end	
87	if r==NRows && c==NCol %right side bottom corner	
88	T2(r,c)=(T(r,c-1)+T(r,c-1)+T(r-1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;	Where only the North and west directions are available
89	end	
90	end	
91	end	

92	i=i+1;	This makes the number of iterations increase by 1 each time.
93		
94	T3=abs(T-T2)<0.1;	This works out the maximum difference between T and T2, when it is less than 0.1 equilibrium is reached
95	pause(0.01)	This makes the code pause for 0.1 seconds when checking for equilibrium
96	contourf(flip(T2),10)	this is the feature that tells Matlab to produce a graph
97		
98	colorbar	This tells the program to add a colour bar to the graph produced
99	end	
100	disp('the time to reach thermal equilibrium in seconds: ');	This displays the time taken to reach equilibrium in seconds
101	Time_taken = i*TS	
102	T	This displays T
103	T2	This displays T2

2.9 Limitations of code

Table 5- Results from code

Grid spacing(metres)	X (metres)	Y (metres)	time step(seconds)	Tolerance value	Equilibrium time(seconds)
----------------------	------------	------------	--------------------	-----------------	---------------------------

1	10	10	3600	0.1	1119600
2	10	10	3600	0.1	1112400
5	10	10	3600	0.1	1058400
4	20	20	3600	0.1	698400
5	20	20	3600	0.1	712800
10	20	20	3600	0.1	594000
Grid spacing(metres)	X (metres)	Y (metres)	time step(seconds)	Tolerance value	Equilibrium time(seconds)
1	10	10	360	0.1	73080
1	10	10	720	0.1	142560
1	10	10	1080	0.1	322920
1	10	10	1440	0.1	535680
1	10	10	1800	0.1	678600
1	10	10	3600	0.1	1119600
Grid spacing(metres)	X (metres)	Y (metres)	time step(seconds)	Tolerance value	Equilibrium time(seconds)
1	10	10	3600	0.1	1119600
1	10	10	3600	0.01	2570400
1	10	10	3600	0.001	4021200
1	10	10	3600	0.0001	5472000
1	10	10	3600	0.00001	6922800
1	10	10	3600	0.000001	8377200

As shown above, various testing was performed on the code to compare the time to reach thermal equilibrium whilst changing one variable per test. This was done to show that the FEA model at this stage still has many issues that need to be overcome.

- 1) X and Y dimensions were set to 10m, time step set at 1 hour, and tolerance set at 0.1 whilst varying the grid spacing. 1-5 and showed similar results, this was repeated dimensions of 20m, with grid spacing varying from 4-10 and again similar results were obtained. From this it can be inferred that the grid spacing itself does affect the time to reach equilibrium but only slightly.
- 2) Grid spacing, X dimension and tolerance were kept constant, while the time step was varied from 6 minutes – 60 minutes. The results from this vary massively, it is believed that this is because the time step being low causes a small difference between T & T2, reaching tolerance prematurely.
- 3) Everything was kept constant whilst varying tolerance from 0.1-0.000001. The results obtained increase almost linearly as the tolerance value drops. This is due to the tolerance being too small, so the plate must be heated for much longer before the variation drops that much.

As well as the limitations highlighted above from the results, there are also another set of limitations:

- 1) The Fourier number, as explained above is the stability criterion that the code must follow, meaning if the Fourier number exceeds 0.25 (for 2D) the code will break.
 As $Fo = (\alpha \cdot TS) / h^2$;
 Where Alpha is a constant at 6.4241×10^{-5} . Therefore, only h and TS can vary.
 Because of this, h and TS have limitations.
- 2) The tolerance value has its limitations, if it is too high it will give false results, but if it's too low the script will take a very long time to solve.
- 3) The script only allows for conductive boundary conditions at the moment, so to simulate a curing over, further work is required.

2.91 Suggested improvements

Suggested improvements for this code are:

- 1) The code should be expanded to work for 3D models, which could be achieved relatively easily in terms of the finite difference model, by simply adding two new directions into the code (in and out), as well as decreasing the Fourier number to below 0.125. However, the issue here lies in the modelling capabilities of the model.
- 2) Eliminating the above limitations is imperative for this code to be feasible.
- 3) Expanding the GUI so that the user can have much more control over the scenarios.
- 4) And finally, the author would like to work out how the code could be used to model some more complex geometries, such as parts with differing cross sections, or parts that are not simple rectangles.
- 5) Expanding the code to use more complex geometries, here there are two separate issues, firstly in modelling, and secondly in the coding itself. However, this will be a very important part of future work.

The above information is taken from the author's previous work in the field.

2.10 What should be done if this project were to continue? – FEA section

Moving forward, this project still has a large portion of work to complete, so this section is dedicated to highlighting the issues that need to be worked through in future, firstly a section highlighting and comparing the limitations and advantages of two FEA software packages that are currently used in industry, this was done to help decide which parts could be incorporated into this project. This section is from the author's previous work in this field.

2.101 FEA software currently used in industry

Currently in industry there is a wide selection of FEA software packages that can be used for a wide variety of applications, from stress analysis to electrical conduction. However, the market is lacking in packages designed specifically to address the problems in thermal propagation. See below the comparison of the two chosen software packages.

2.1011 The first software package being looked into is ANSYS FLUENT.

ANSYS is a software package that can be used to simulate many different scenarios such as:

- 4) Physics
- 5) Structural
- 6) Vibration
- 7) Fluid dynamics
- 8) Heat transfer
- 9) and electromagnetics for engineers

This package allows the user to simulate real world scenarios virtually, so that prototypes for products can be tested before manufacture begins. A large part of this software is the features that allow it to determine and improve weak points, computing life, and foreseeing possible future complications.

The software package is modular as shown in the below figure, meaning that users only need to access the required features, as well as this ANSYS can work integrated with other engineering software packages, by adding CAD and FEA modules.

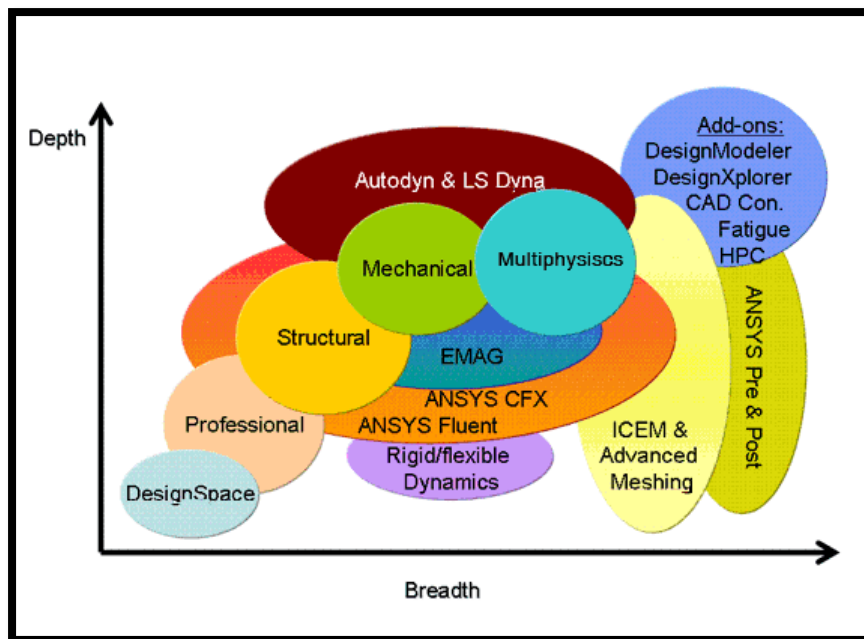


Figure 12 - ANSYS Modules (taken from <http://www.figes.com.tr/english/ansys/ansys.php>)

2.1012 The second software package that is being looked into is SOLIDWORKS.

As described by anon. (2018).

The SOLIDWORKS simulation suite allows the user to evaluate performance, quality and innovation by establishing real world scenarios to test products prior to manufacture. The software package allows the user to model, and simulate many parameters:

- 10) durability
- 11) static and dynamic response
- 12) motion assembly
- 13) heat transfer
- 14) fluid dynamics
- 15) and plastic injection

2.102 Table 6 - Comparison of FEA software in industry

	ANSYS FLUENT	SOLIDWORKS
Ease of use	ANSYS FLUENT unlike Solidworks, is very complicated to use, as it allows the user to change nearly any parameter imaginable. As well as this the user MUST define each parameter manually, as ANSYS is meant for a skilled user, rather than a novice.	The Solidworks package is widely seen to be fairly simple to use, as it is designed ergonomically, making it very intuitive, and only allowing the user to change and define certain parameters. Allowing novice users to simulate relatively simple scenarios. Due to this it can be used to produce a simulation and obtain results fairly easily.
Accuracy of results	The accuracy of the results obtained in ANSYS are completely dependent on the user inputs. As mentioned previously, ANSYS makes the user input many more parameters, and allows the user to change even more. Meaning that if the user is not experienced, it is likely that the results produced will be completely wrong. However because of this level of detail in the inputs, an experienced user could produce results much more accurate and detailed than that of a Solidworks user.	The issue of accuracy is similar in Solidworks, as if the user inputs are wrong, the results obtained from the analysis would be wrong, however Solidworks only allows the user to change certain parameters, and many are automatically generated. This means that a non-experienced user could obtain good results by using this package. However again if the user inputs are wrong, the results are likely to be wrong.
Meshing strategies	The meshing strategies of both Solidworks and ANSYS at their basic level are quite similar. In that they can both produce a mesh on complex components using a range of shapes.	However Solidworks has several limitations in this area. As if the mesh size is too large, it can cause the simulation to completely ignore sections of the design and not account for them in simulation. However Solidworks has an

	However ANSYS is capable of meshing components with much more detail and precision than Solidworks. Also There is no automatic meshing feature in ANSYS, which makes meshing very complex and time consuming.	automatic meshing feature, which meshes parts very quickly if the inputs are viable.
Modelling strategies	As ANSYS is primarily an FEA package, the modelling features it contains are sub-par. The UI used for modelling is poor, and it is relatively simple in comparison to Solidworks. So ideally the user would need to model the component in a different software package and then analyse it in ANSYS.	As the solid works package is mainly a CAD package, with the added extra of having an FEA module, the modelling in Solidworks is of a high standard, allowing you to easily model components, and then simulate simple scenarios.

Information in table collated from several sources:

<https://forum.solidworks.com/thread/83117>

<https://www.solidsolutions.co.uk/solidworks/simulation/packages/default.aspx>

From the comparison above, it is suggested that the FEA Model should include:

1) Ease of use

Solidworks is very simple, allowing novice users to begin to learn the FEA method, and ANSYS is very complex, allowing Professional users to obtain good results. In my model the author would like to incorporate both of these advantages. By designing the user interface with ease of use and ergonomics in mind, whilst still keeping the level of detail achieved in ANSYS. The author would do this by introducing separate competency levels for differing complexities in design. For example, a novice user setting, would much like Solidworks only allow the user to change certain parameters, and auto generate the rest. However, if the professional user setting is used, it would open up the parameters to be changed, much like in ANSYS.

2) Accuracy of results

This issue would be resolved partially by the above ease of use section, as the user would be only using the software to the level that they are capable. However, the ideal goal it to be able to produce accurate results, regardless of which version is used.

3) Modelling strategies

The modelling in both SOLIDWORKS and ANSYS are at a much higher level than the modelling in my model. This is because at the moment the modelling is limited to simple geometries. This could be overcome this by introducing a separate modelling feature, similar to Solidworks' modelling section, to allow the user to design the part they wish to model accurately.

As ANSYS is primarily an FEA package, the modelling features it contains are sub-par. The UI used for modelling is poor, and it is relatively simple in comparison to Solidworks. So ideally the user would need to model the component in a different software package and then analyse it in ANSYS.

As the solid works package is mainly a CAD package, with the added extra of having an FEA module, the modelling in Solidworks is of a high standard, allowing you to easily model components, and then simulate simple scenarios.

4) Meshing strategies

The meshing strategies in both SOLIDWORKS and ANSYS are at a level much higher than the meshing in the model developed. This is because in the model, the mesh is defined by the user who decides node spacing, plate size etc. This could be achieved, by implementing similar meshing strategies to Solidworks mainly, with an auto meshing feature, however whilst still keeping the flexibility of the meshing in ANSYS.

2.11 What should be done if this project were to continue – Programming language section

See below further research into the advantages and limitations of the currently used language, as well as a comparison with the suggested new language. This was done to highlight the drawbacks of using Matlab in comparison to other software.

2.111 Matlab – current language

As described by M,Haughn (2015) 'Matlab is a fourth generation programming language and numerical analysis environment', that is currently used in industry by many professionals in fields such as engineering, computer science, signal processing, etc. with many applications such as:

- User interface creation
- Matrix calculations and manipulations
- Running and developing algorithms
- Etc.

Matlab was initially developed by Cleve Moler in the 1970's as a tool to aid his computer science students, it was then transferred to the 'C' programming language for commercialisation in 1984.

Due to this, Matlab has several limitations in use:

- The program was developed over 40 years ago, because of this many of its features are outdated in comparison to more modern software
- The user is limited when trying to use Matlab with other software packages
- Certain software packages are not suitable for use with Matlab
- Code reuse is not supported
- C++ code generation is not supported
- Conversions of models in some cases is not supported.

With these limitations it has been decided that although Matlab is a good base to start this project, moving forward it is imperative that a different coding method is put into use.

2.112 'R' Programming language – suggested new language

R is a programming language designed specifically for use with statistical computing and graphical information, which was developed by Bell laboratories (formally AT&T), by John Chambers and his colleagues. R allows for a wide range of statistical, and graphical techniques. R can be considered as a branch of the language 'S' which is the previous iteration of 'R'. Where S is generally the language used in research, and R provides an

open source application to participate with. R and S are available for free, via the Free Software Foundation's GNU general public license (open source), meaning that this software package could be used to develop the code at no cost.

See below the advantages and disadvantages of the R programming language, as described by Anon (2012)

Advantages

- R is the most comprehensive statistical analysis software package on the market at current, incorporating all standard statistical test methods, models and analysis methods. Alongside a comprehensive language that is used to manage and manipulate data.
- R is widely used as a test area to develop new products and technologies
- R is designed specifically for statistical analysis by statisticians and researchers and is now maintained by a core team of 19 developers, including very senior statisticians.
- R has high level graphical capabilities, providing a fully functional graphic language that is far beyond most statistical and graphic packages.
- R is validated through its open source nature, as it can be modified by anyone who uses it. R has been reviewed by numerous statisticians.
- Anyone is allowed to provide bug fixes, code enhancements and new packages.
- There are no licensing restrictions when using R, so it can be run anywhere, at any time, and products produced using R can even be sold.
- R has a large library of pre-existing packages available to be used by anyone, in topics such as data mining, spatial analysis, etc.
- R is usable cross platform, as it is compatible with many operating systems, and different hardware configurations. It is popularly used in conjunction with Microsoft Windows, Linux.
- R can be used with a multitude of other tools, such as Microsoft Excel, CSV les, SAS, SQLite, etc. and can be used to produce graphics in many formats, from JPG to PDF.
- R has active user forums that anyone can use to ask questions that can be answered by experienced users, experienced statisticians, and the developers themselves.

- R works well whilst prototyping, as long as the data sets being used are relatively small. Many large companies use R as a sandbox environment to test their data. A developer may produce a project in the R environment initially, and if successful it will be transferred to a more suitable language.

Disadvantages

- R is not a simple language to learn, meaning that it has a steep learning curve. However, this is the case for many of the high end statistical languages. R is not ideal for a novice user, however developers have produced several GUIs (graphical user interfaces) that encompass the point and click interactions.
- The documentation is often very complex, it will require a high level knowledge of statistics to understand.
- As R is open source, some of the packages available aren't fully finished products that you'd expect to see on the market from other languages.
- There is no one to complain to if something goes wrong, as R was developed by many people over a long period, as people have freely devoted their time to produce it. Problems are usually dealt with by opening the mailing lists and asking for help, or by paying someone with more experience to help.
- R in general doesn't give much thought to memory management, meaning that if left unchecked, R could easily use all available memory. This can cause problems in that it can massively slow down the analysis being performed if this isn't accounted for, and older computers may not be able to handle and complex process.
- R experiences a bottlenecking issue when large data sets are being used, this issue has been looked into by many statisticians and at the moment it hasn't been overcome, as explained by Dexlab (2016)

From the above analysis it has been suggested that the R programming language will be further researched and implemented when future work begins.

3.0 Conclusion

By making use of the thermal diffusion equation, in conjunction with the Taylor and Maclaurin series' finite difference methods Croft et al, (1989) the initial iteration of the FEA model was designed. The model at current is capable of mapping the thermal propagation through an aluminium part, however with some limitations:

1. The model only works for 2D currently
2. The model only works for simple geometries currently
3. The model doesn't account for heat loss currently
4. The model only works for aluminium currently
5. There are some problems in the mathematics derivation that need consideration

From this initial work, two experiments were completed, the first to begin to correlate the model to a real life scenario. It was found that both the results from the code and the results from the real life test followed similar trend lines, which is enough to indicate that the model has promise, and requires future developmental work. The second experiment investigated calculating the convective heat transfer coefficient, which is used to determine the heat loss the part experiences whilst being heated.

Moving forward to the next stage of this project, here are the recommendations for future work:

1. It is recommended that the model be transferred from Matlab, to a more suitable programming language, such as R.
2. It is recommended that the mathematics regarding the finite difference method be reviewed to ensure that they are fully capable of completing the task.
3. It is recommended that the code be expanded further, to account for 3D components and complex geometries.
4. It is recommended that further research is completed into current FEA software packages, and the features involved.

4.0 References

Analyticstrainings.com. (2018). *Advantages and Disadvantages of R – AnalyticsTrainings.com*. [online] Available at: <http://analyticstrainings.com/?p=101> [Accessed 23 Apr. 2018].

Butterfield, A. and Ekembe Ngondi, G. (2016). A dictionary of computer science. Oxford: Oxford University Press.

Croft, D. and Stone, J. (1989). Heat transfer calculations using finite difference equations. PAVIC Publications.

DexLab Analytics | Big Data Hadoop SAS R Analytics Predictive Modeling & Excel VBA. (2018). *The Limitation of R Programming*. [online] Available at: <https://www.dexlabanalytics.com/blog/the-limitation-of-r-programming> [Accessed 23 Apr. 2018].

Durodola, J. (2017). Functionally graded adhesive joints – A review and prospects. *International Journal of Adhesion and Adhesives*, 76, pp.83-89.

Figes.com.tr. (2018). What is ANSYS. [online] Available at: <http://www.figes.com.tr/english/ansys/ansys.php> [Accessed 26 Mar. 2018].

Forum.solidworks.com. (2018). SW's automatic meshing vs. ANSYS ICEM | SOLIDWORKS Forums. [online] Available at: <https://forum.solidworks.com/thread/83117> [Accessed 26 Mar. 2018].

Hulimka, J. and Kałuza, M. (2017). Preliminary Tests of Steel-to-steel Adhesive Joints. *Procedia Engineering*, 172, pp.385-392.

Heshmati, M., Haghani, R., Al-Emrani, M. and André, A. (2018). On the strength prediction of adhesively bonded FRP-steel joints using cohesive zone modelling. *Theoretical and Applied Fracture Mechanics*, 93, pp.64-78.

R-project.org. (2018). *R: What is R?*. [online] Available at: <https://www.r-project.org/about.html> [Accessed 23 Apr. 2018].

Solidsolutions.co.uk. (2018). SOLIDWORKS Simulation (FEA) Packages | Solid Solutions. [online] Available at: <https://www.solidsolutions.co.uk/solidworks/simulation/packages/default.aspx> [Accessed 26 Mar. 2018].

Stein, N., Weißgraeber, P. and Becker, W. (2015). A model for brittle failure in adhesive lap joints of arbitrary joint configuration. *Composite Structures*, 133, pp.707-718.

Tutorial.math.lamar.edu. (2018). Calculus II - Taylor Series. [online] Available at: <http://tutorial.math.lamar.edu/Classes/CalcII/TaylorSeries.aspx> [Accessed 26 Mar. 2018].

Uk.mathworks.com. (2018). *Limitations- MATLAB & Simulink- MathWorks United Kingdom*. [online] Available at: https://uk.mathworks.com/help/physmod/simscape/ug/limitations.html?s_tid=gn_loc_drop [Accessed 23 Apr. 2018].

von Fraunhofer, J. (2012). Adhesion and Cohesion. *International Journal of Dentistry*, 2012, pp.1-8.

References – Images

<http://www.adhesives.org/adhesives-sealants/science-of-adhesion/adhesion-cohesion>

<http://www.figes.com.tr/english/ansys/ansys.php>

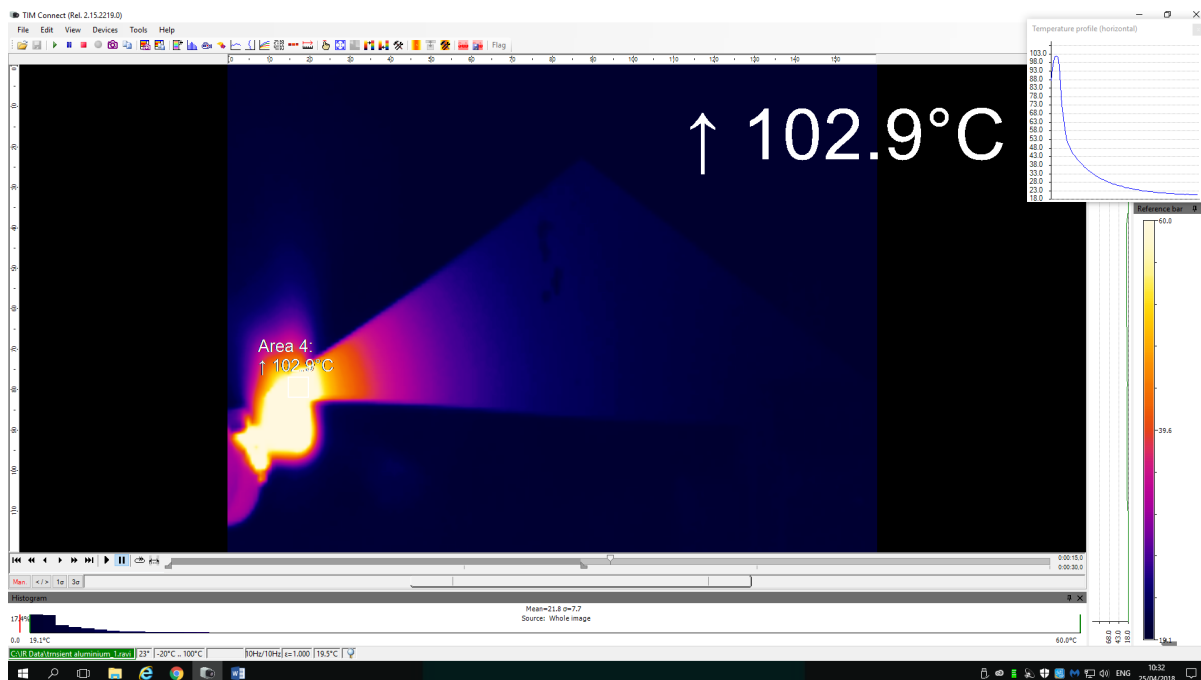
References – Author's previous work

K.Bouleghlimat (2018) 'Advanced Stress Analysis & FEA' - *M6X0042*' University of Trinity Saint David

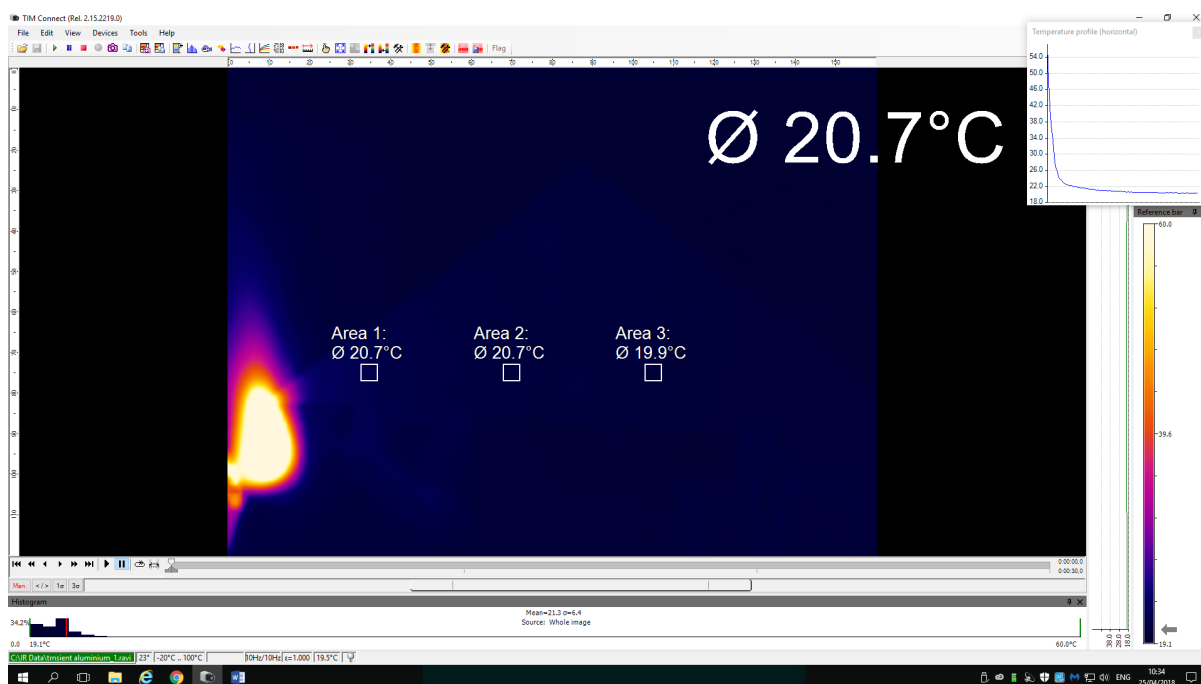
5.0 Appendices

Appendix 1 – Experiment 1 Screenshots

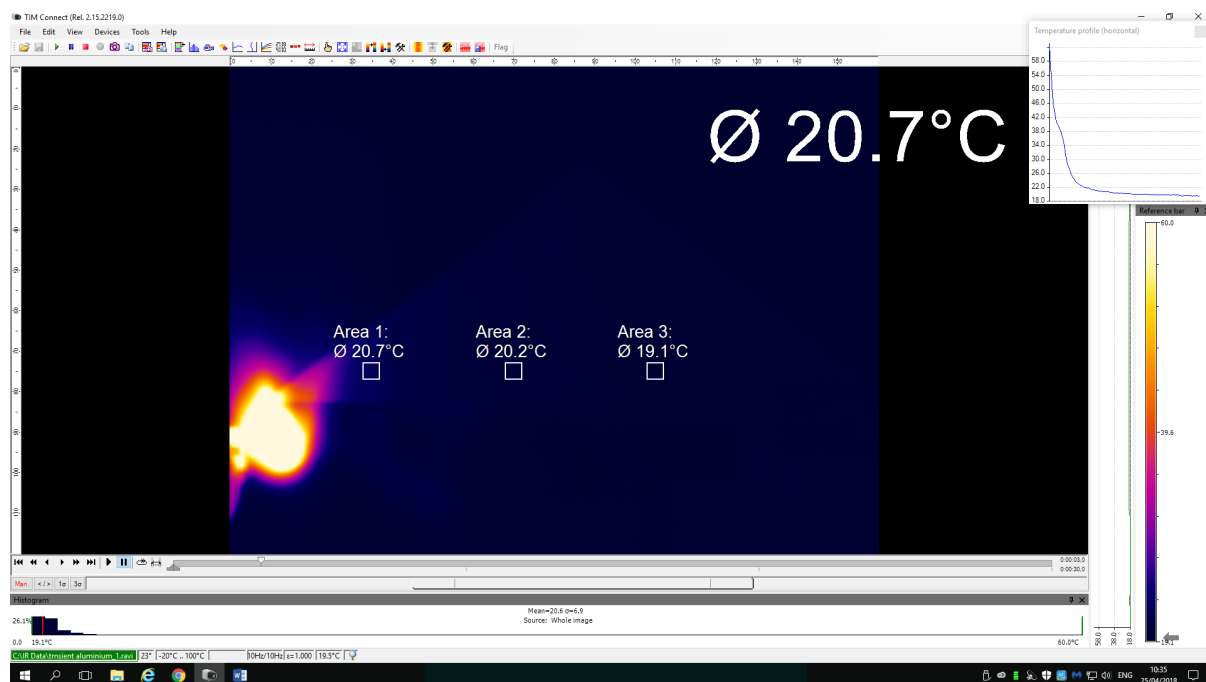
0 seconds



3 seconds

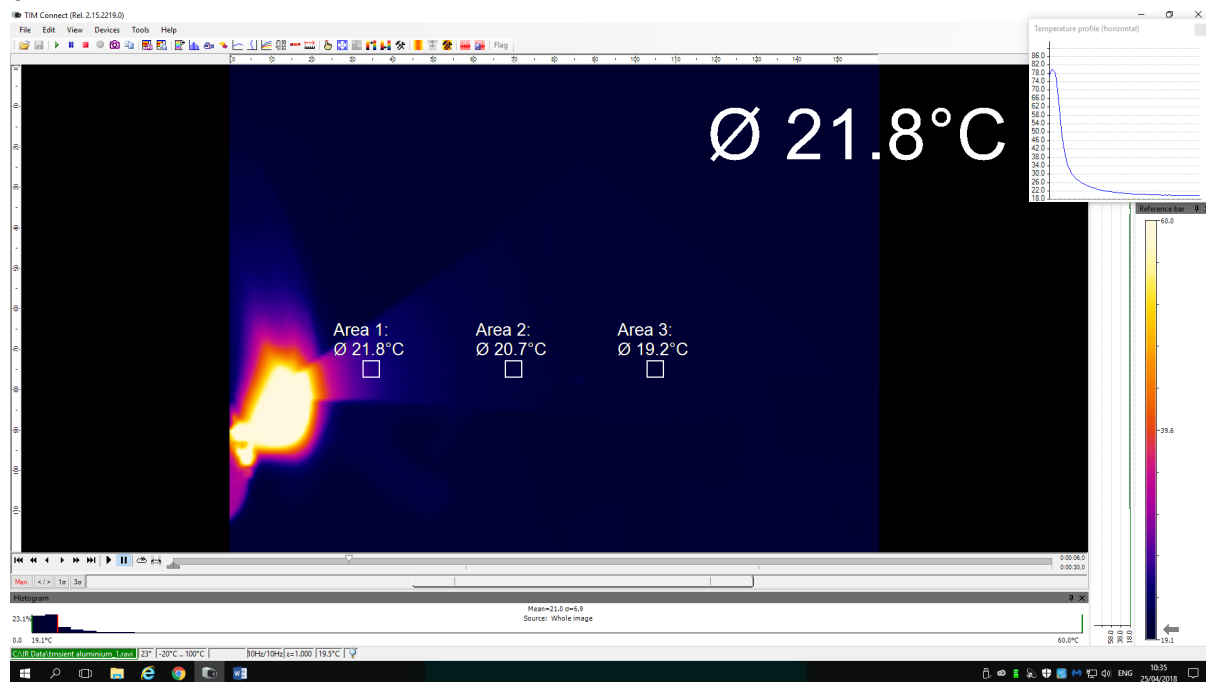


6 seconds

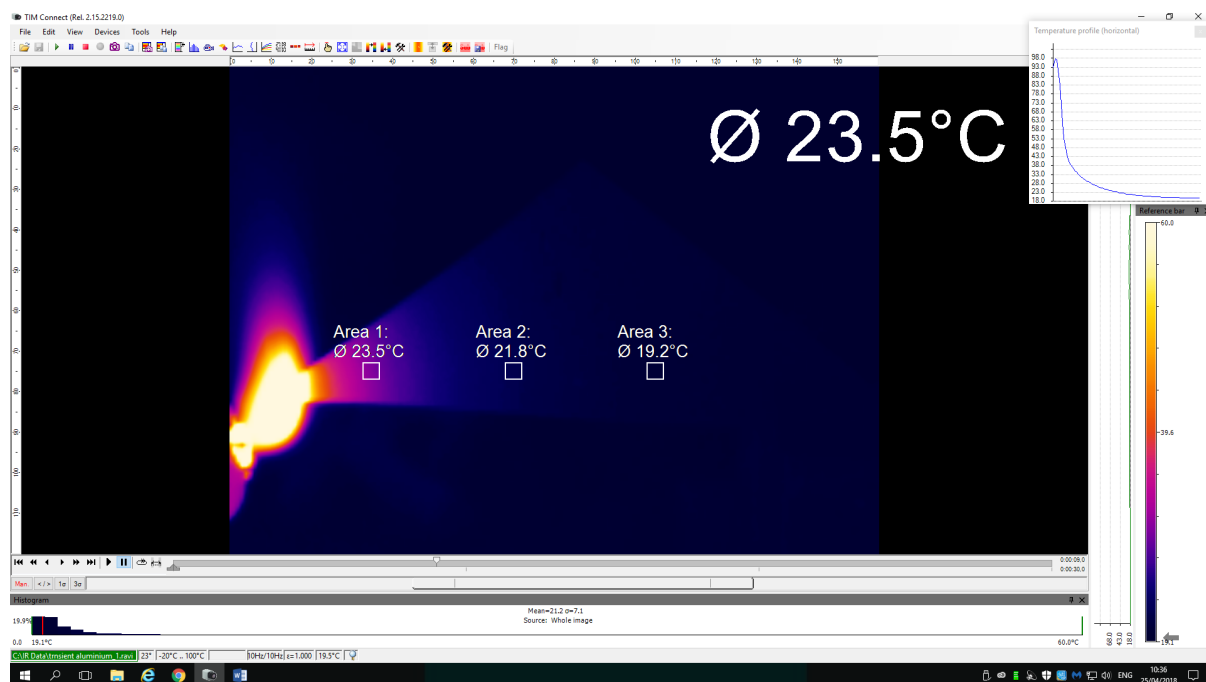


9

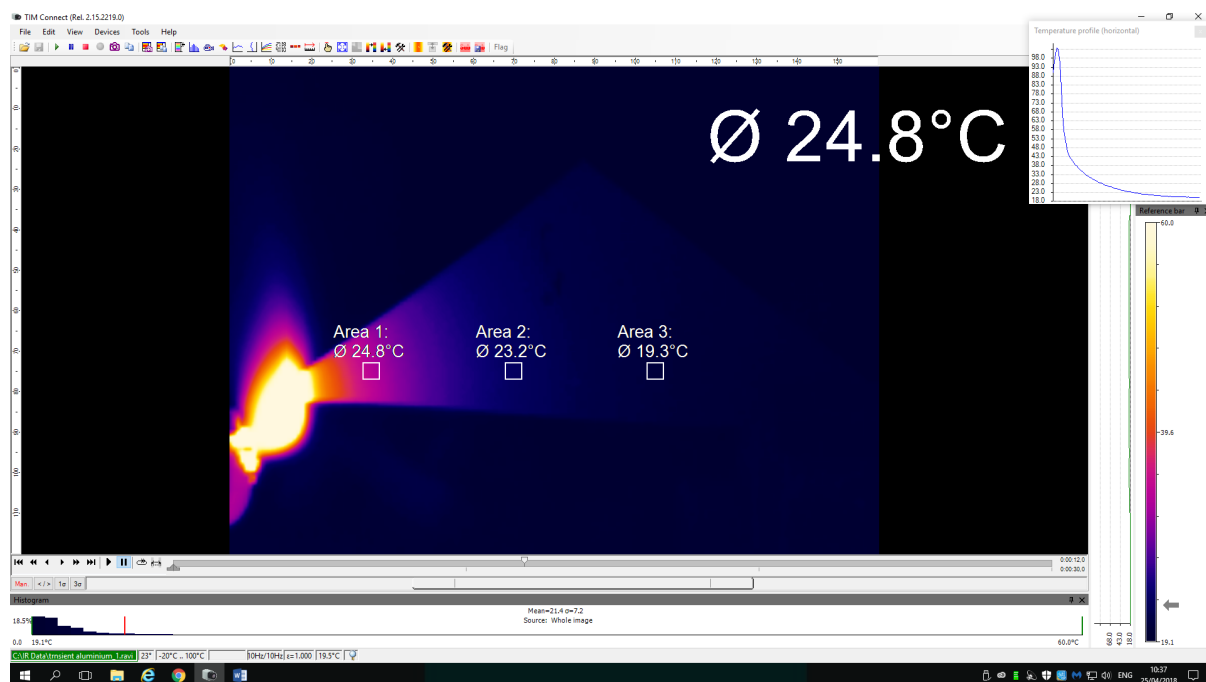
seconds



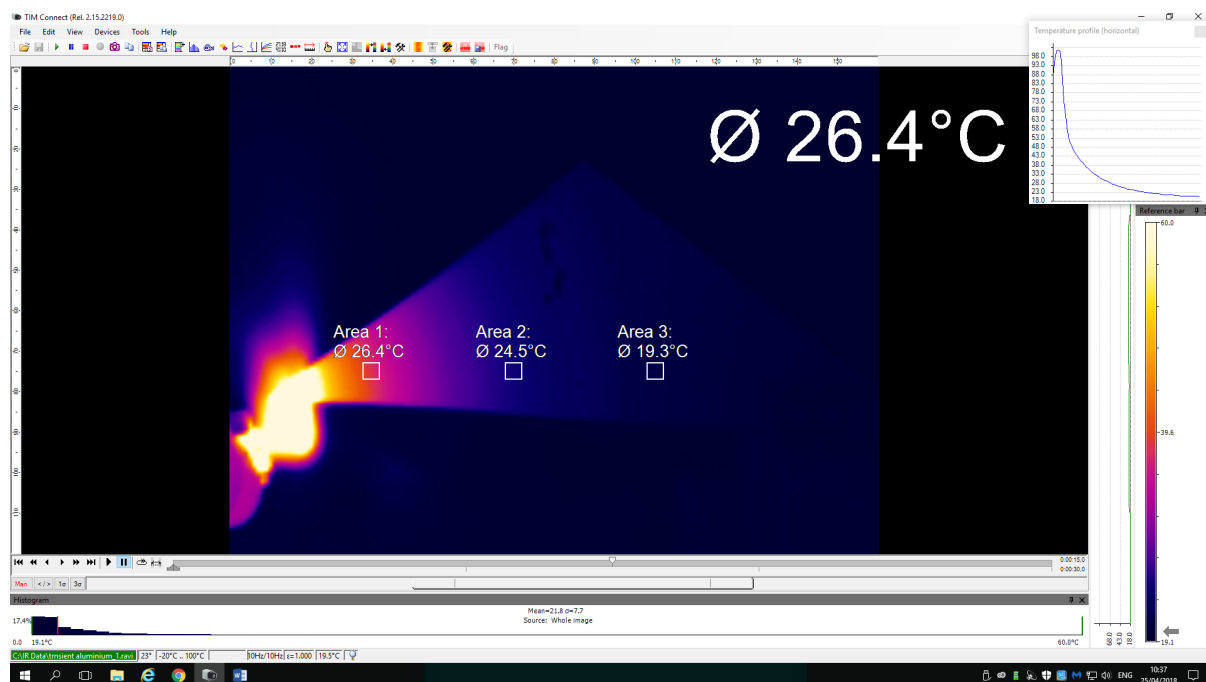
12 seconds



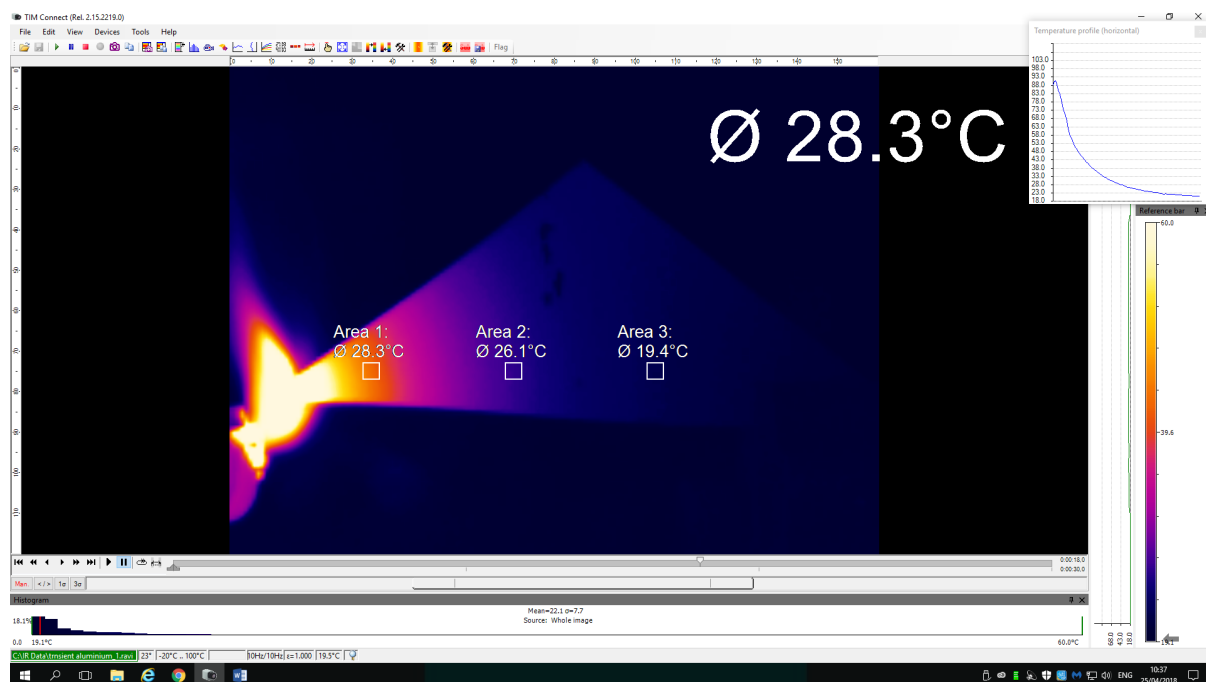
15 seconds



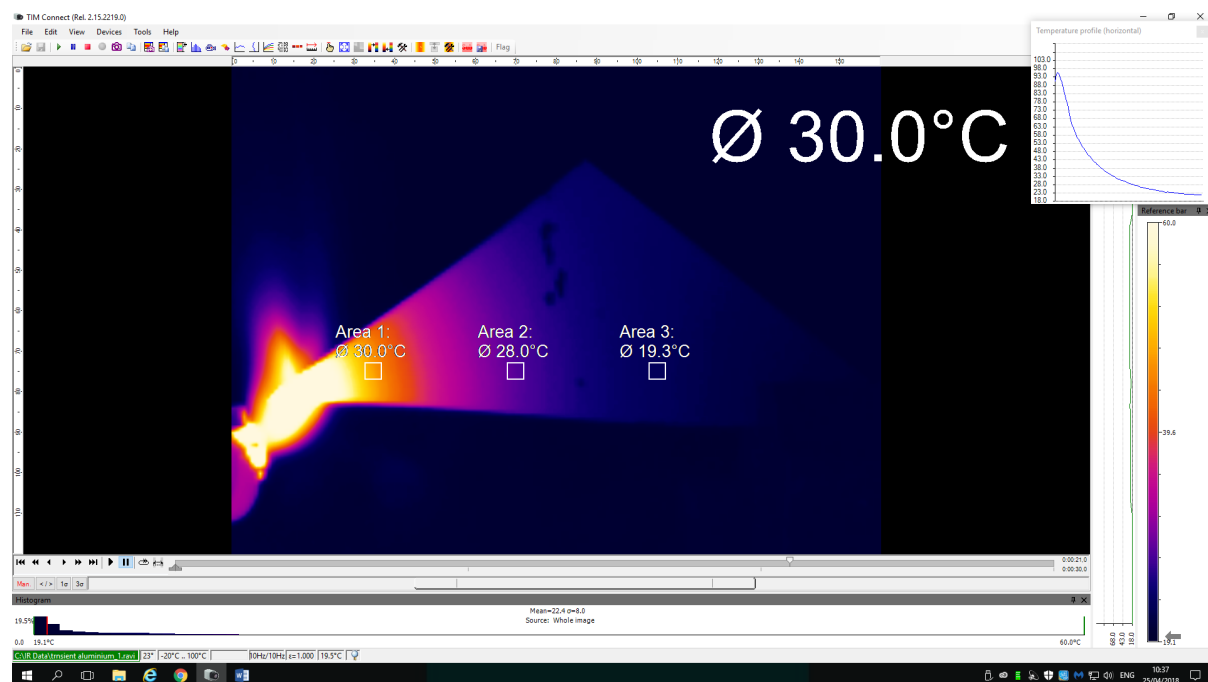
18 seconds



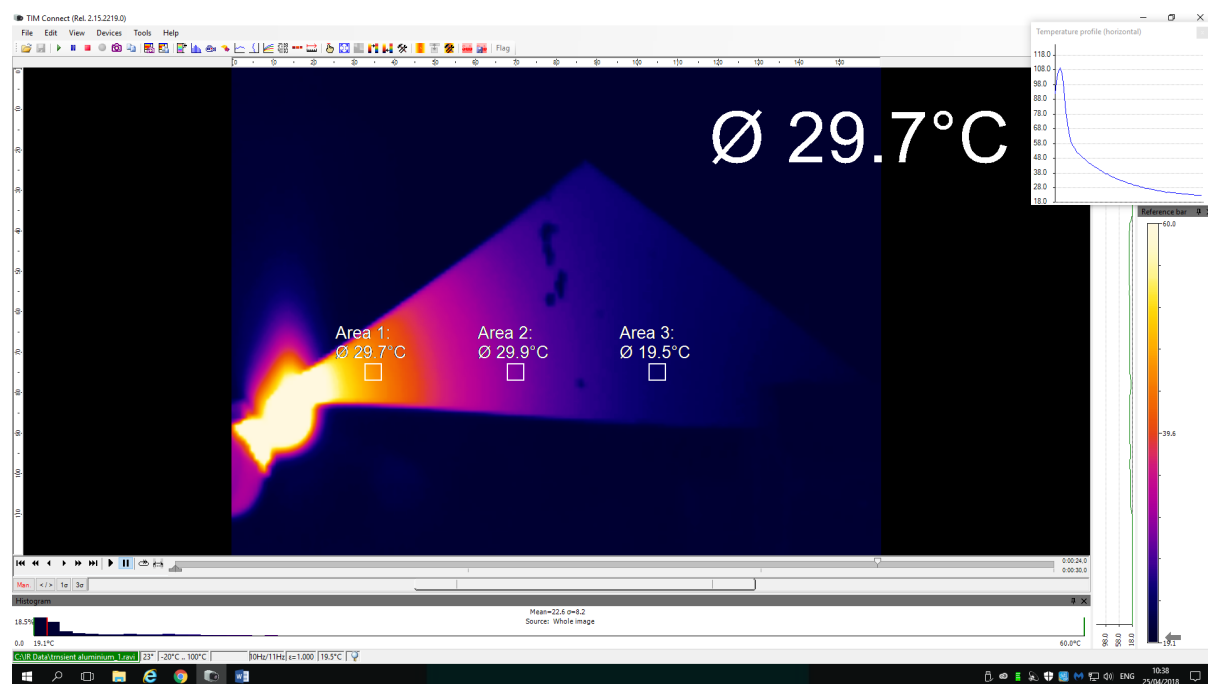
21 seconds



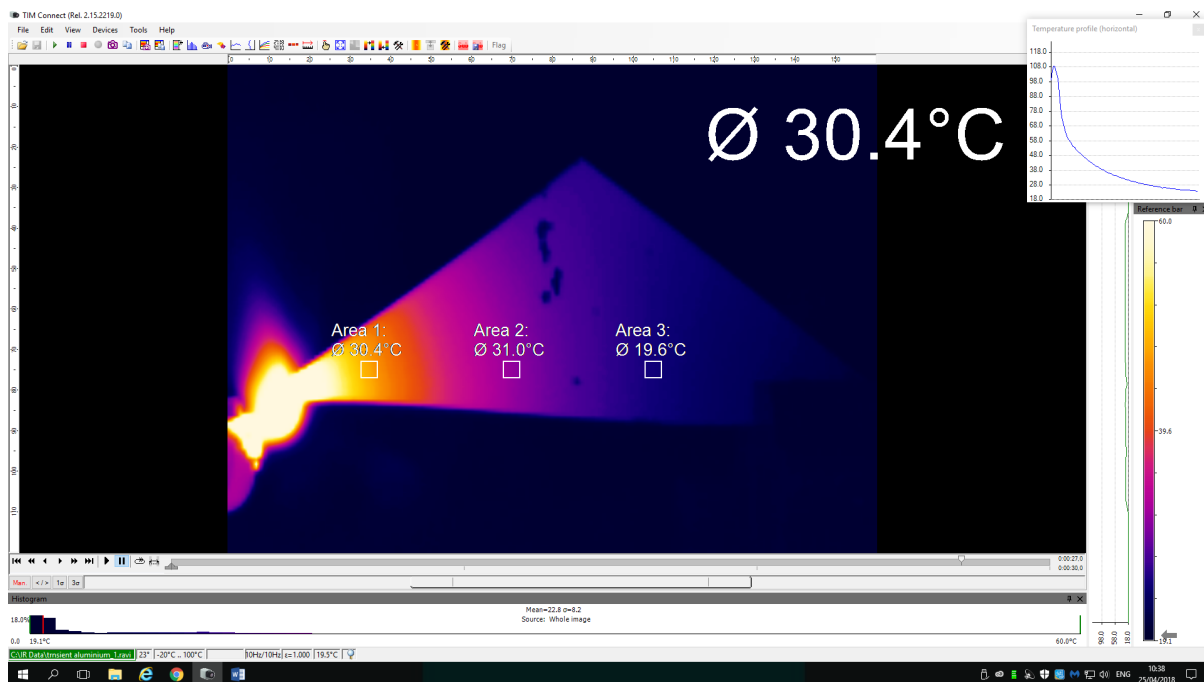
24 seconds



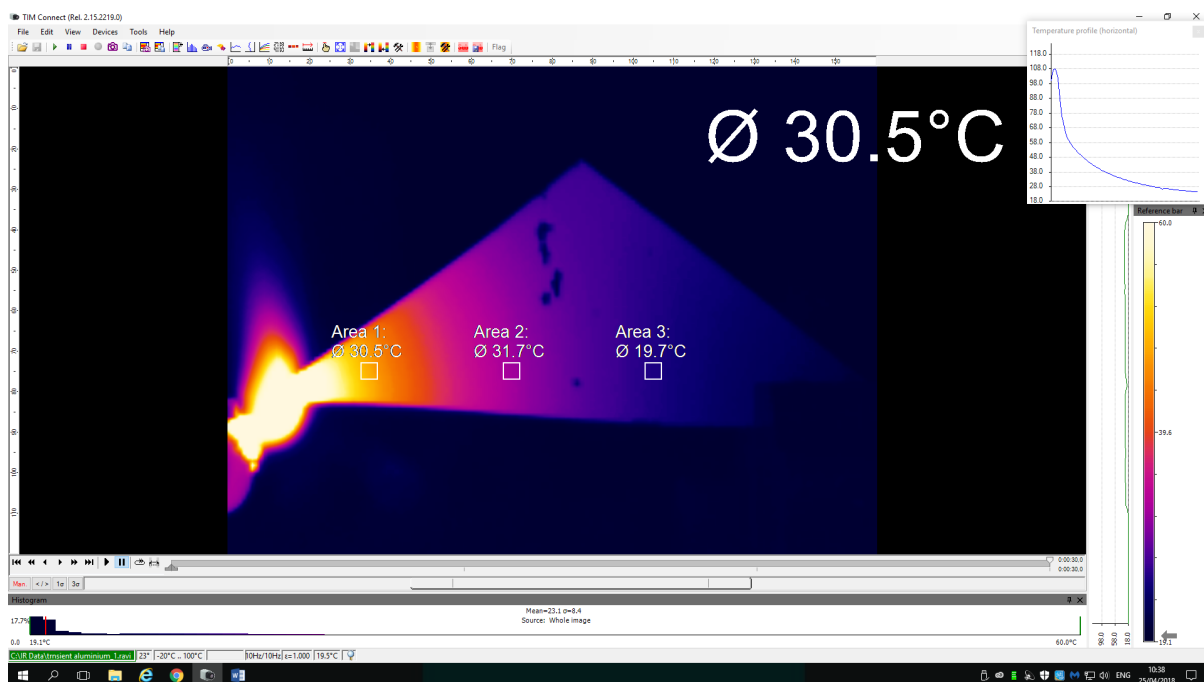
27 seconds



30 seconds



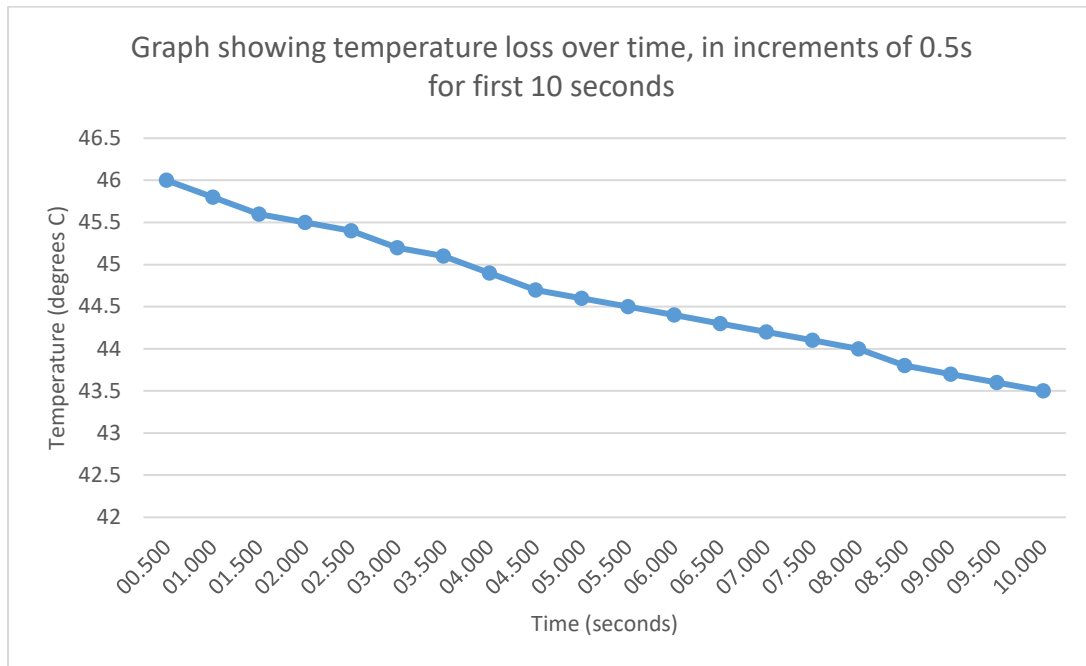
END



Appendix 2 – Experiment 2 Results Table (10 seconds)

time (seconds)	temp (°C)	temp (°K)
00.500	46	319
01.000	45.8	318.8
01.500	45.6	318.6
02.000	45.5	318.5
02.500	45.4	318.4
03.000	45.2	318.2
03.500	45.1	318.1
04.000	44.9	317.9
04.500	44.7	317.7
05.000	44.6	317.6
05.500	44.5	317.5
06.000	44.4	317.4
06.500	44.3	317.3
07.000	44.2	317.2
07.500	44.1	317.1
08.000	44	317
08.500	43.8	316.8
09.000	43.7	316.7
09.500	43.6	316.6
10.000	43.5	316.5

Appendix 3 – Experiment 2 Results Graph (10 seconds)



Appendix 4 – authors previous work 1

Table of Contents

1.0 Introduction	59
2.0 Summary	60
Q1. A very brief outline describing your dissertation	61
Q2a. What you would like to assess via FEA including what type of Analysis you would carry out. ...	63
The first software package being looked into is ANSYS FLUENT.	39
The second software package that is being looked into is SOLIDWORKS.	40
Comparison	41
Q2b. What it would achieve, how you would use this to add value to your dissertation and how you would measure the success of your analysis.	67
Q3. What software package you would use to carry-out this analysis and why.	67
4.0 References	68

1.0 Introduction

This assignment will firstly give a brief outline of my Major Project. Due to the nature of my project, I will be answering the questions provided in a slightly different way.

It will then look into two software packages, ANSYS FLUENT and SOLIDWORKS simulation suite, and compare several important factors, including ease of use, meshing strategies etc.

And from this information I will highlight the areas of each software package that I'd like to incorporate into my major project.

And the final question will be answered as if I were to choose one of the software packages to optimise the curing process, rather than using the one I am producing.

2.0 Summary

As above, the aims of this assignment were to look into several different FEA packages and compare them to each other, finally deciding which one I would choose to use to improve my Major project.

From my research I found that both packages excel in different areas:

- SOLIDWORKS is primarily a CAD program that is very good while modelling and producing assemblies, however the FEA features are at a lower level than that of ANSYS.
- On the other hand, ANSYS is a FEA package that is very good at modelling real life scenarios in much detail, however the modelling features are at a lower level than that of SOLIDWORKS.

But overall, I chose ANSYS as the software package I would use to improve my major project for several reasons:

- 16) *ANSYS is a more Industrial aimed FEA package, which can simulate real world scenarios at a much higher level than SOLIDWORKS, which in its essence is a CAD package, with an FEA feature.*
- 17) *The accuracy of the results from ANSYS can be of a higher standard.*
- 18) *The modelling features in SOLIDWORKS outperform ANSYS in many areas, however due to the modular nature of ANSYS, it is possible to import the models from separate software. Therefore I would choose ANSYS for analysis, whilst using a different software such as SOLIDWORKS for modelling.*
- 19) *The meshing in ANSYS is much more comprehensive, as you can define many more parameters than that of SOLIDWORKS.*

Q1. A very brief outline describing your dissertation

1. A very brief outline describing your dissertation. (30 marks – 1000 words)

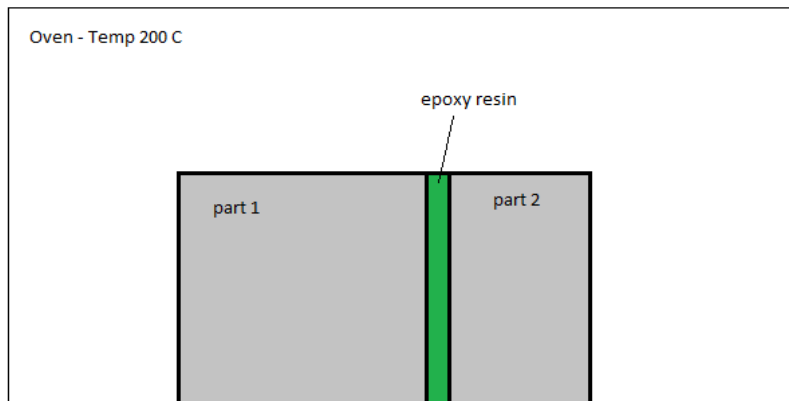


Figure 4 - Basic Project Diagram (author's own, 2018)

My dissertation is based on optimising a curing process that uses aluminium bonding epoxy resins, by producing an FEA model. With the main issues being;

- 1) Dissimilarly sized parts being heated as above in figure 1 at the same rate will cause part 2 to heat up faster, and reach curing temperature before part1, therefore the epoxy will begin to adhere unevenly, causing problems in the integrity of the bond.
- 2) Companies are unsure how long exactly to keep parts under heat to reach the optimal curing temperature, so in industry at the moment, it is common practice to overheat both parts, to ensure that they are hot enough. However this is a waste of energy in the heating, and time as they keep the parts in for longer than required to be safe.

I will do this by:

- Using Matlab to produce a code that can map thermal propagation through a component for each time step. By generating a matrix of connected nodes that create the size of the plate you wish to map.
- Initially with conductive boundary conditions, then moving onto convective boundary conditions, to simulate an oven.
- The code will map work by using the Taylor series as described by C, Maclaurin(1730) and B, Taylor(1715) and covered in a tutorial by P, Dawkins (2018) which basically lets you rewrite a function in terms of infinite other functions. Which means that you can relate a value at point x_0 to the value of the function at a nearby point $(x_0 + h)$.

$$f(x) = f(x_0) + f'(x_0) \cdot \frac{(x - x_0)^1}{1!} + f''(x_0) \cdot \frac{(x - x_0)^2}{2!} + f'''(x_0) \cdot \frac{(x - x_0)^3}{3!} + f^{(4)}(x_0) \cdot \frac{(x - x_0)^4}{4!} + \dots$$

$$f(x_0 + h) = f(x_0) + h \cdot f'(x_0) + \frac{h^2}{2!} f''(x_0) + \frac{h^3}{3!} f^{(3)}(x_0) + \dots$$

We can rearrange the above to get

$$\frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0) + \frac{h}{2!} f''(x_0) + \frac{h^2}{3!} f^{(3)}(x_0) + \dots$$

And this can be used to make the **forward difference approximation** to $f'(x_0)$ thus

$$f'(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h}$$

- Next using the Taylor series, the first and second derivatives can be estimated via numerical differentiation.
- This can be achieved by using either the forward, backwards or central difference method, for this application the forward difference method, specifically the explicit method, was used, due to being the simplest method to work out, and it is suitable for the application.
- Now that the underlying maths has been finished, it can be transferred to the form of the code.

The Aluminium properties are:

- Conductivity: $k = 143 \text{ W/m.C}$
- Density: $\rho = 2.8 \times 10^3 \text{ kg/m}^3$
- Specific Heat: $c = 795 \text{ J/kg.C}$

The prescribed initial and boundary conditions are:

Boundary Condition along North edge: Gaussian distribution of temperatures of amplitude

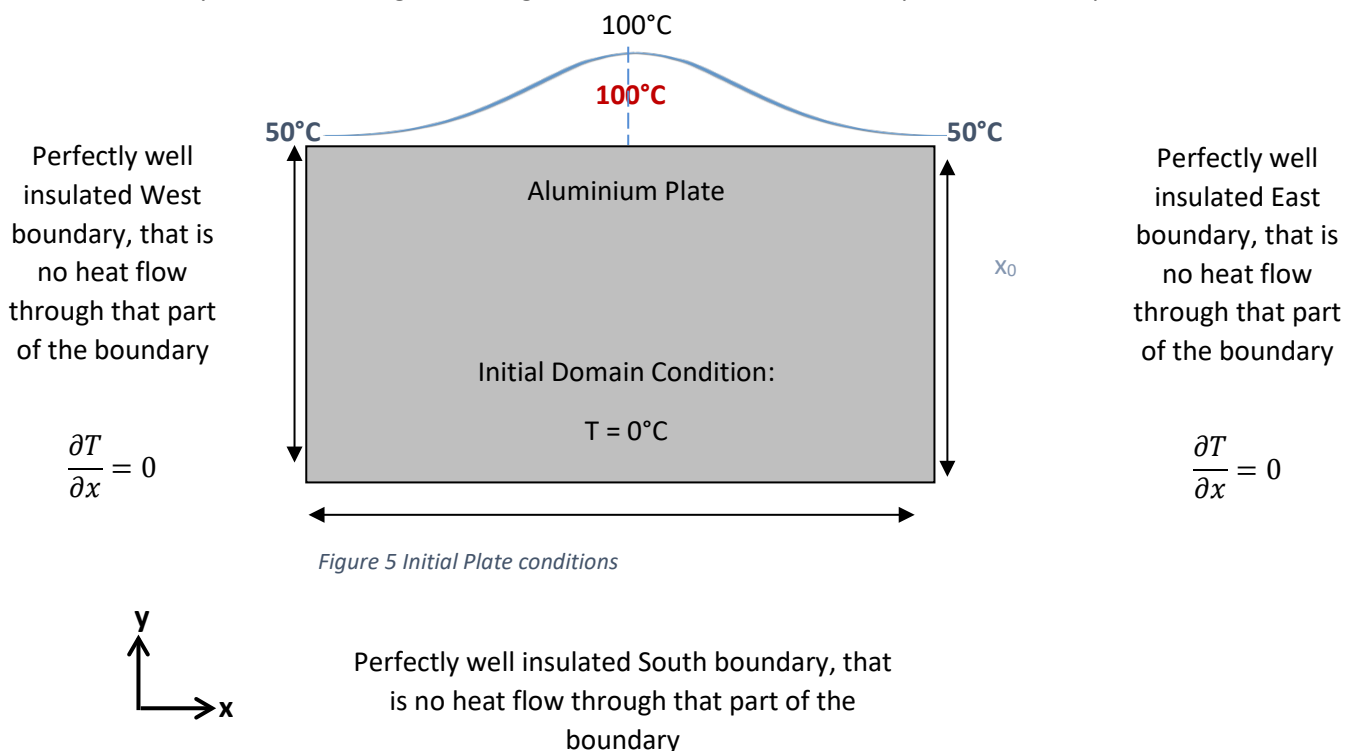


Figure 5 Initial Plate conditions

- The initial boundary conditions were set as above. The initial boundary conditions are conduction based, this was because the maths behind the conduction are relevant in both the convective and conductive conditions, as in the convection problem, only the boundary will be undergoing convection and radiation whereas the whole interior domain will be undergoing conduction.
 - Once this section is completed, the code will work for 2D Conduction problems. However the code only accounts for heat in, and not heat out.
 - So using a thermographic camera I will:
 - Firstly perform an experiment to calculate the heat out of the plate, by heating a plate, recording the loss in heat over time, then working out the heat loss --> energy loss --> convective energy loss-- > radiation energy loss
 - Then once these results have been added to the code, I will perform an experiment to ensure that my results from real life testing now match the results from the code.
 - The next step of the project would be to expand the code, to work for 3D parts, this could be achieved fairly simply for non-complex parts, as it is a case of modifying the code slightly so that when it performs the forward difference method, it currently accounts for North, South, East and West, and I would need to add "in" and "out". However with the initial set up the code only has one matrix. To make this work, extra matrices would need to be added to account for material thickness, however the maths itself would stay the same, other than the section known as the stability criteria.
- 20) The Fourier number is known as the Stability criteria for this code, if the Fourier number comes out with a value of over 0.25 the code will not work for a 2D problem, and similarly for a 3D model it must not exceed 0.125. Because of this the values in the formula to calculate the Fourier number have limitations.
- 21) As $Fo = (\alpha \cdot TS) / h^2$;
- 22) Where α doesn't change and has a value of $6.4241e-5$. Therefore only h (node spacing) and TS (time step) can be changed. Because of this and the strict boundary of 0.25 for the Fourier number, h and TS have limitations. " h " cannot exceed a defined value, and the time step can't be too small.
- 23) The final issue for the code is working out how to apply convective boundary conditions.
- 24) Once applied the code will be complete and in theory should be able to map the thermal propagation of a material, that is placed either in an oven with convective boundary conditions, or with conductive boundary conditions.
- 25) Then if I were to continue this project, I would want to optimise the curing process of a specific epoxy bonding resin that is used in industry, by mapping the thermal propagation through a part, and then by trial and error, working out the optimal temperatures for curing, as well as the depth the temperature requires to reach for optimal bonding. Ultimately to reduce time and energy used in these processes, whilst keeping the structural integrity of the bonds the same.

Q2a. What you would like to assess via FEA including what type of Analysis you would carry out.

- a) Due to the nature of my project, I will not be using any form of FEA, however I will be researching into the topic to find out the limitations and advantages of several different FEA packages on the market that can perform similar tasks to what I want from my project.

The first software package being looked into is ANSYS FLUENT.

ANSYS is a software package that can be used to simulate many different scenarios such as:

- 26) Physics
- 27) Structural
- 28) Vibration
- 29) Fluid dynamics
- 30) Heat transfer
- 31) and electromagnetics for engineers

This package allows the user to simulate real world scenarios virtually, so that prototypes for products can be tested before manufacture begins. A large part of this software is the features that allow it to determine and improve weak points, computing life, and foreseeing possible future complications.

The software package is modular as shown in the below figure, meaning that users only need to access the required features, as well as this ANSYS can work integrated with other engineering software packages, by adding CAD and FEA modules.

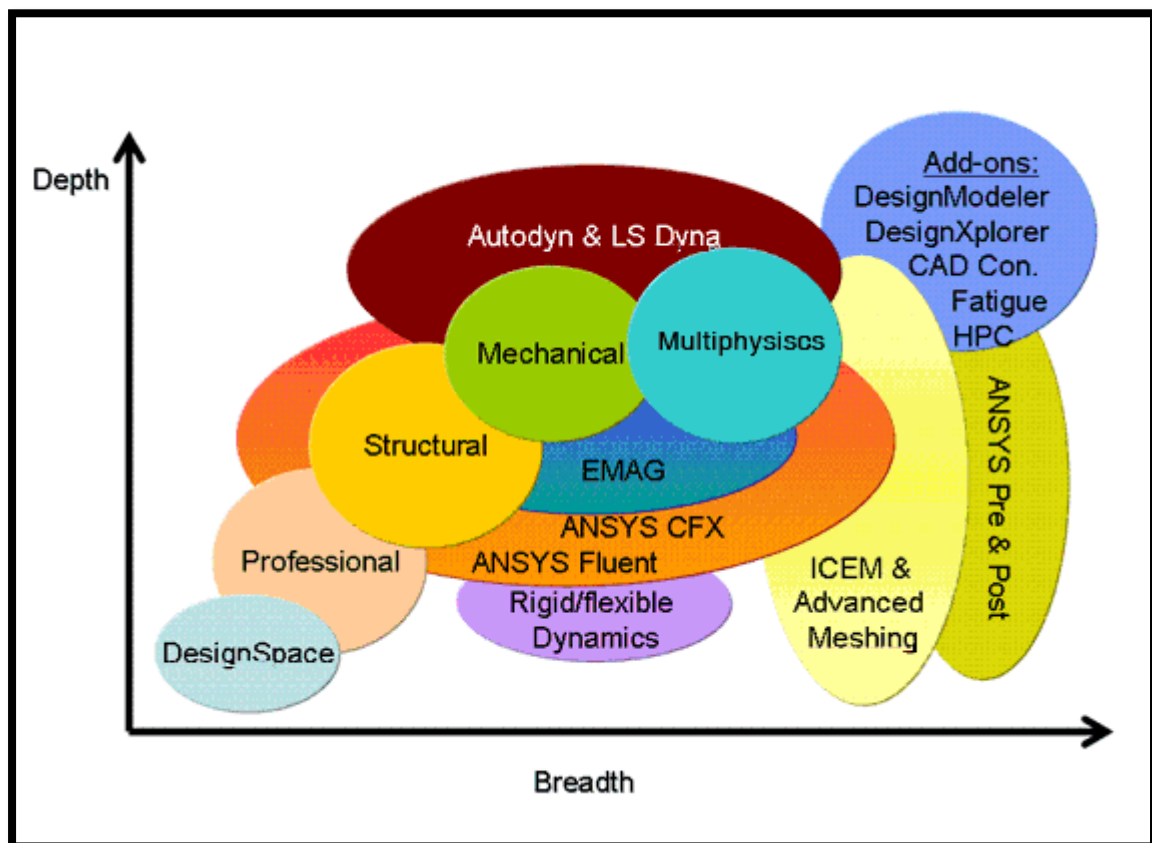


Figure 6 - ANSYS Modules (taken from <http://www.figes.com.tr/english/ansys/ansys.php>)

The second software package that is being looked into is SOLIDWORKS.

The SOLIDWORKS simulation suite allows the user to evaluate performance, quality and innovation by establishing real world scenarios to test products prior to manufacture. The software package allows the user to model, and simulate many parameters:

- 32) durability
- 33) static and dynamic response
- 34) motion assembly
- 35) heat transfer
- 36) fluid dynamics
- 37) and plastic injection

As described by anon 2018,
<https://www.solidsolutions.co.uk/solidworks/simulation/packages/default.aspx>

455 words	ANSYS FLUENT	SOLIDWORKS
Ease of use	ANSYS FLUENT unlike Solidworks, is very complicated to use, as it allows the user to change nearly any parameter imaginable. As well as this the user MUST define each parameter manually, as ANSYS is meant for a skilled user, rather than a novice.	The Solidworks package is widely seen to be fairly simple to use, as it is designed ergonomically, making it very intuitive, and only allowing the user to change and define certain parameters. Allowing novice users to simulate relatively simple scenarios. Due to this it can be used to produce a simulation and obtain results fairly easily.
Accuracy of results	The accuracy of the results obtained in ANSYS are completely dependent on the user inputs. As mentioned previously, ANSYS makes the user input many more parameters, and allows the user to change even more. Meaning that if the user is not experienced, it is likely that the results produced will be completely wrong. However because of this level of detail in the inputs, an experienced user could produce results much more accurate and detailed than that of a Solidworks user.	The issue of accuracy is similar in Solidworks, as if the user inputs are wrong, the results obtained from the analysis would be wrong, however Solidworks only allows the user to change certain parameters, and many are automatically generated. This means that a non-experienced user could obtain good results by using this package. However again if the user inputs are wrong, the results are likely to be wrong.
Meshing strategies	The meshing strategies of both Solidworks and ANSYS at their basic level are quite similar. In that they can both produce a mesh on complex components using a range of shapes. However ANSYS is capable of meshing components with much more detail and precision than Solidworks. Also There is no automatic meshing feature in ANSYS, which makes meshing very complex and time consuming.	However Solidworks has several limitations in this area. As if the mesh size is too large, it can cause the simulation to completely ignore sections of the design and not account for them in simulation. However Solidworks has an automatic meshing feature, which meshes parts very quickly if the inputs are viable.
Modelling strategies	As ANSYS is primarily an FEA package, the modelling features it contains are sub-par. The UI used for modelling is poor, and it is relatively simple in comparison to Solidworks. So ideally the user would need to model the component in a different software package and then analyse it in ANSYS.	As the solid works package is mainly a CAD package, with the added extra of having an FEA module, the modelling in Solidworks is of a high standard, allowing you to easily model components, and then simulate simple scenarios.

Information in table collated from several sources:

<https://forum.solidworks.com/thread/83117>

<https://www.solidsolutions.co.uk/solidworks/simulation/packages/default.aspx>

Q2b. What it would achieve, how you would use this to add value to your dissertation and how you would measure the success of your analysis.

Due to the nature of my dissertation, in this section I will be looking into the features of current FEA software packages that I would like to include in my model.

From the comparison above, in my FEA Model I would like to include:

38) Ease of use

Solidworks is much simple, allowing novice users to begin to learn the FEA method, and ANSYS is very complex, allowing Professional users to obtain good results. In my model I would like to incorporate both of these advantages. By designing the user interface with ease of use and ergonomics in mind, whilst still keeping the level of detail achieved in ANSYS. I would do this by introducing separate competency levels for differing complexities in design. For example, a novice user setting, would much like Solidworks only allow the user to change certain parameters, and auto generate the rest. However if the professional user setting is used, it would open up the parameters to be changed, much like in ANSYS.

39) Accuracy of results

This issue would be resolved partially by the above Ease of use section, as the user would be only using the software to the level that they are capable. However the ideal goal it to be able to produce accurate results, regardless of which version is used.

40) Modelling strategies

The modelling in both SOLIDWORKS and ANSYS are at a much higher level than the modelling in my Model. This is because at the moment the modelling is limited to simple geometries. I would like to overcome this by introducing a separate modelling feature, similar to Solidworks' modelling section, to allow the user to design the part they wish to model accurately.

As ANSYS is primarily an FEA package, the modelling features it contains are sub-par. The UI used for modelling is poor, and it is relatively simple in comparison to Solidworks.

So ideally the user would need to model the component in a different software package and then analyse it in ANSYS.

As the solid works package is mainly a CAD package, with the added extra of having an FEA module, the modelling in Solidworks is of a high standard, allowing you to easily model components, and then simulate simple scenarios.

41) Meshing strategies

The meshing strategies in both SOLIDWORKS and ANSYS are at a level much higher than the meshing in my Model. This is because in my model, the mesh is defined by the user who decides node spacing, plate size etc. I would like to achieve this in my model, by implementing similar meshing strategies to Solidworks mainly, with an auto meshing feature, however whilst still keeping the flexibility of the meshing in ANSYS.

Q3. What software package you would use to carry-out this analysis and why.

Due to the nature of my dissertation, it doesn't make sense to choose a software package to carry out the analysis, as there is no analysis to be done using FEA software. However, if I were to choose a program from the two to use to try to optimise a curing process, I would choose ANSYS FLUENT.

This is because:

- 42) ANSYS is a more Industrial aimed FEA package, which can simulate real world scenarios at a much higher level than SOLIDWORKS, which in its essence is a CAD package, with an FEA feature. As this project is looking into a real world industrial problem, it only makes sense to choose the more comprehensive software.*
- 43) The accuracy of the results from ANSYS can be of a higher standard although it is much harder to learn and use, assuming that a professional is using it to perform the analysis.*
- 44) As discussed above, the modelling features in SOLIDWORKS outperform ANSYS in many areas, however due to the modular nature of ANSYS, it is possible to import the models from separate software. Therefore I would choose ANSYS for analysis, whilst using a different software such as SOLIDWORKS for modelling.*
- 45) The meshing in ANSYS is much more comprehensive, as you can define many more parameters than that of SOLIDWORKS. For example Solidworks generally only uses tetrahedral elements, whereas in ANSYS the element shape can be changed to suit the needs of the component.*

4.0 References

Butterfield, A. and Ekembe Ngondi, G. (2016). *A dictionary of computer science*. Oxford: Oxford University Press.

Figes.com.tr. (2018). *What is ANSYS.* [online] Available at: <http://www.figes.com.tr/english/ansys/ansys.php> [Accessed 26 Mar. 2018].

Forum.solidworks.com. (2018). *SW's automatic meshing vs. ANSYS ICEM | SOLIDWORKS Forums.* [online] Available at: <https://forum.solidworks.com/thread/83117> [Accessed 26 Mar. 2018].

Solidsolutions.co.uk. (2018). *SOLIDWORKS Simulation (FEA) Packages | Solid Solutions.* [online] Available at: <https://www.solidsolutions.co.uk/solidworks/simulation/packages/default.aspx> [Accessed 26 Mar. 2018].

Tutorial.math.lamar.edu. (2018). *Calculus II - Taylor Series.* [online] Available at: <http://tutorial.math.lamar.edu/Classes/CalcII/TaylorSeries.aspx> [Accessed 26 Mar. 2018].

Appendix 5 – authors previous work 2

Table of Contents

1.0	Summary	70
-----	---------	----

2.0 Introduction	72
3.0 Method	72
3.1 Given values	73
3.2 Dimplat x and y	73
3.3 Timestep & Node Spacing	74
3.4 Remainder for nodes, must equal 0	74
3.5 Time step verification	74
3.6 Calculation Number of rows and columns, and MidC	75
3.7 Gaussian distribution	75
3.8 Implementing the thermal diffusion formulae	76
3.9 Final Display functions:	77
4.0 Results	78
5.0 Conclusion	79
5.1 Limitations of script	36
5.2 Suggested improvements	38
7.0 Appendix 1 – Full script	Error! Bookmark not defined.

Figure number	Page number
1	3
2	4
3	4
4	5
5	6
6	7
7	7
8	8
9	8

Summary

This report look at initially the construction of the script to be used, explained step by step. With diagrams to show how it has been implemented to solve the time dependant thermal diffusion equation using the finite difference method to calculate the temperature distribution within a rectangular plate with a prescribed initial and boundary conditions.

From implementing the code, I found that it has several limitations, in the Stability criteria, time step and tolerance values which ideally need to be reduced for the script to run to its full potential. These limitations were even more noticeable from my results, as they show that using value for the time step that are too small cause the script to give incorrect results, due to previously mentioned issues, and if the tolerance value isn't correctly set the results again could be incorrect.

Moving forward I would like to continue adding to the code so that it could be expanded to produce 3D models or use different materials.

2.0 Introduction

During this assignment I will implement a model in Matlab to solve the time-dependent thermal diffusion equation using the finite difference method to calculate the temperature distribution within a rectangular plate with prescribed initial and boundary conditions.

I have been tasked with calculating the temperature distribution within an aluminium plate through time.

The calculations will be done considering the region of interest (that is the 2 dimensional aluminium plate) covered by a uniform grid, as shown in the following figure, where the grid spacing is $h = \Delta x = \Delta y$.

The temperature calculations will be based on the **Explicit Method**.

The Aluminium properties are:

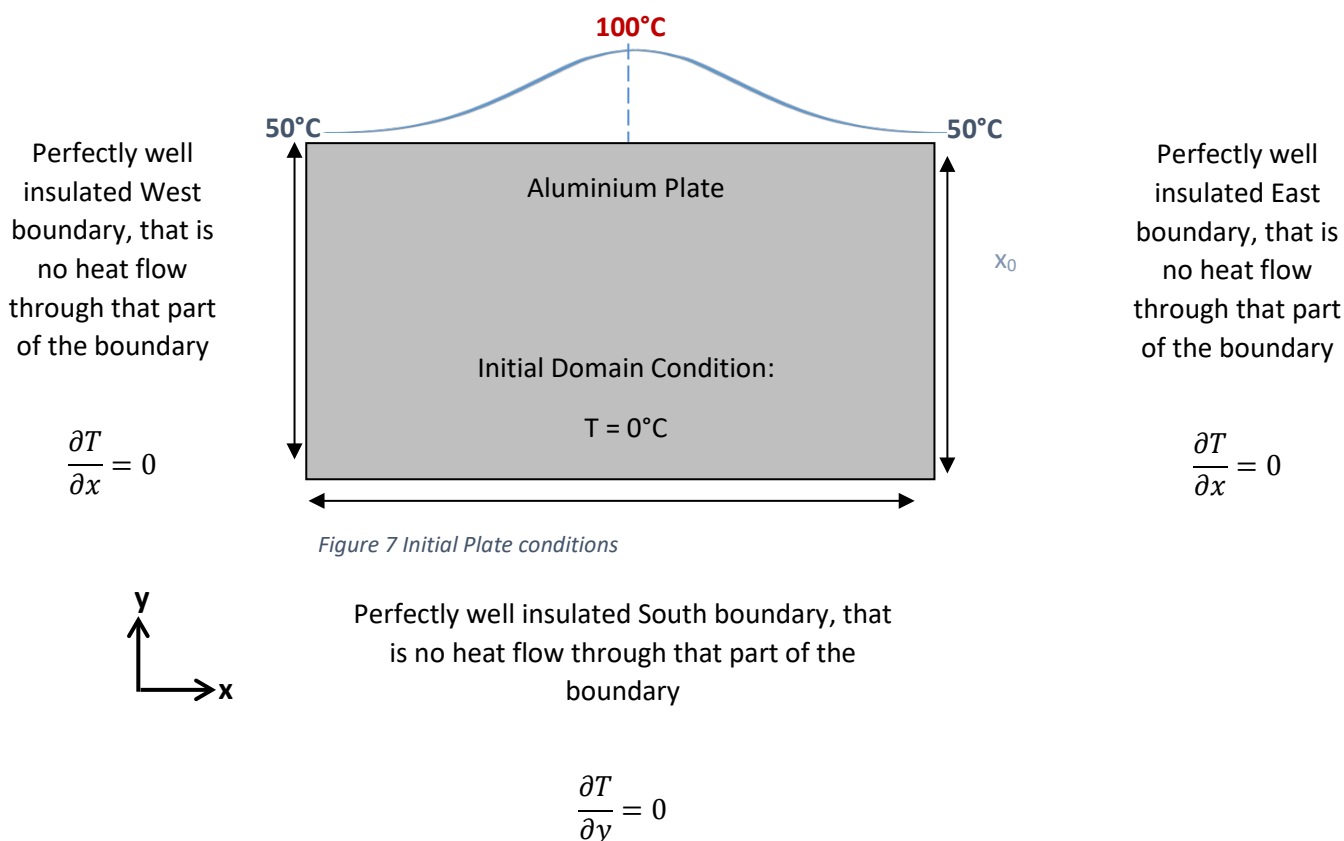
Conductivity: $k = 143 \text{ W/m.C}$

Density: $\rho = 2.8 \times 10^3 \text{ kg/m}^3$

Specific Heat: $c = 795 \text{ J/kg.C}$

The prescribed initial and boundary conditions are:

Boundary Condition along North edge: Gaussian distribution of temperatures of amplitude 100°C



3.0 Method

3.1 Given values

```

1 -   clc
2 -   clear all
3 -   %Given values
4 -   conductivity_K=143;
5 -   Density=2.8*10^3;
6 -   SpecificHeat=795;
7 -   Alpha = conductivity_K/(SpecificHeat*Density);

```

Figure 8 Given values

Shown above are the values given in the assignment brief;

(4) Thermal conductivity (K) = 143

(5) Density (ρ) = 2.8×10^3 Kg/m³

(6) Specific heat capacity = 795

Finally the formula to relate these three values is given, with:

(7) Alpha having the value of 6.4241×10^{-5}

3.2 Dimplate x and y

```

9   %Dimplate x and y
10  Dimplate_x = input('Please enter X dimension in metres ');
11  while Dimplate_x <= 0
12      Dimplate_x = input('Please enter positive X dimension in metres ');
13  end
14  Dimplate_y = input('Please enter Y dimension in metres ');
15  while Dimplate_y <= 0
16      Dimplate_y = input('Please enter positive Y dimension in metres ');
17  end
18  %Timestep
19  TS = input('Please enter timestep in seconds ');
20  while TS <= 0
21      TS = input('Please enter positive timestep in seconds ');
22  end
23  %Node spacing
24  h = input('Please enter node spacing in metres ');
25  while h <= 0
26      h = input('Please enter positive node spacing in metres ');
27  end

```

Figure 9 Dimplate x and y

This portion of the code defines the X dimension and Y dimension of the plate being used, and are called “Dimplate_x” and “Dimplate_y” respectively.

The first line in the code (10) is an input that requests that the user enter a value for the X dimension.

The second part is a verification step (11-13), so if Dimplate_x is given a value below 0, a prompt will show asking for a Positive X value.

This is then repeated for the Y value (14-17).

Once this is complete the script knows the size of plate that you wish to model.

3.3 Timestep & Node Spacing

The next step is to define the Time step (19-22). This part is very similar to the Dimplate_x part, as it asks for a time step from the user, goes through a verification step and if it passes it moves on to the next part of code.

Again this step repeats the same process, where the user is asked to input a value for the node spacing (h)(24-27). This value is important as it is used to define the number of columns and rows will be modelled in the code.

The only difference between this and the previous code is that the verification step used ensures that the value of “h” is above 1. As using values below 1 cause problems in the code

3.4 Remainder for nodes, must equal 0

```

27 - end
28 - %remainder for nodes, must equal 0
29 - gr_l=rem(Dimplate_x,h);
30 - gr_w=rem(Dimplate_y,h);
31 - while h>=Dimplate_y || h>=Dimplate_x || gr_l~=0 || gr_w~=0
32 -     disp('Node spacing must comply with plate size ')
33 -     h=input('Enter valid node spacing');
34 -     gr_l=rem(Dimplate_x,h);
35 -     gr_w=rem(Dimplate_y,h);
36 - end
37 - %Timestep verification
38 - while TS > (h^2/(Alpha))
39 -     disp('Ensure timestep is not greater than: ')
40 -     disp(h^2/(4*Alpha))
41 -     TS=input('Enter Valid timestep ');
42 - end
43 -
44 - %NROW/COL
45 - NRows= Dimplate_y/h+1;
46 - NCol= Dimplate_x/h+1;
47 - T=zeros(NRows,NCol);
48 - T(1,:)=50;
49 - MidC=((Dimplate_x+2)/2);

```

Figure 10 Remainder for nodes

This next part (28-30) uses the X and Y dimensions, as well as the h value to calculate whether the number of columns and rows is correct for this model. As $\text{Dimplate}_x / h$ must be a whole number, to give a whole number of nodes. The function “rem” allows you to check whether the equation gives a remainder. If it does then it fails as a remainder value of 0 is required for the code to continue.

gr_l is the remainder for the x value

And gr_w is the remainder for the y value

Again there is a verification step (31-36), so if “h” is above the X or Y value, the user is prompted, or if the answer comes out with a remainder, the script will display a message stating that “node spacing must comply with plate size”.

3.5 Time step verification

This step is a verification step for the time step (38-42) which relates to a different portion of the code, known as the Fourier number. This ensures that the value for the time step is within a reasonable boundary, so that it doesn't break the code.

3.6 Calculation Number of rows and columns, and MidC

NRows calculates the number of rows in the plate, from the x and h values, similarly NCol, works out the number of columns (45, 46).

Next $T = \text{zeros}(\text{NRows}, \text{NCol})$ (47) produces a matrix with the size of NCol and NRows, that is populated by 0.

$T(1,:) = 50$ (48) this makes is so that all the columns in the first row display 50.

And finally (49) using the x dimension with h, MidC is calculated, to be used in the next part.

3.7 Gaussian distribution

```

50
51     %gaussian distribution
52 -   corner = 50;
53 -   amplitude = 100;
54 -   Sigma=sqrt((- (1-MidC)^2) / (2*log(corner/amplitude)));
55
56 -   for c=1:NCol
57 -       T(1,c)=amplitude*exp(-(c-MidC)^2/(2*Sigma^2));
58 -   end
59
60
61 -   T2=T;
62 -   Fo=(Alpha*TS)/h^2;
63 -   %while
64 -   i=0;
65 -   T4=ones(NRows,NCol);
66 -   T3=zeros(NRows,NCol);

```

Figure 11 Gaussian Distribution

This part of the code defines the boundary conditions of the matrix. Those boundaries being a Gaussian distribution on the top face of the plate, ranging from 50-100 Degrees Celsius.

Corner(52), and amplitude(53) are the values limiting the distribution, with 50 being the lowest value in the corners, and 100 being the highest In the centre.

Next Sigma(54) is calculated, using values for MidC, and corner, amplitude, giving the standard deviation of the distribution.

The next section ensures that the values are entered in their correct positions.(56-58)

The next section is where the Fourier number comes up. This is known as the stability criteria of the script, and MUST have a value below 0.25 for a 2D model, and 0.125 for a 3D model. (62)

This is why the values for h and TS are so important, as they must work with the Fourier number formula to give a reasonable result, if the value is above the upper limit, it will break the code.

"l" (64) is the number of iterations that have taken place, and is given a value of 0 to begin with.

The final part T3(65), and T4(66) are codes to produce two matrices, which are again used as a verification step, however this step is to find out when thermal equilibrium has been reached. Each one creates a matrix of differences, and when the differences equal 0 thermal equilibrium has been reached.

3.8 Implementing the thermal diffusion formulae

```

67 - while isequal(T3,T4)==0
68 -     T=T2;
69 -     for r=2:NRows
70 -         for c=1:NCol
71 -
72 -             if c>1 && c<NCol && r~=NRows %domain where we have all 4 directions
73 -                 T2(r,c)=(T(r,c+1)+T(r+1,c)+T(r-1,c)+T(r,c-1)-(4-(1/Fo))*T(r,c))*Fo;
74 -             end
75 -
76 -             if c==1 && r~=NRows %left side mid points
77 -                 %T2(r,c)=(2*T(r,c+1)+2*T(r+1,c)-(4-(1/Fo))*T(r,c))*Fo;
78 -                 T2(r,c)=(2*T(r,c+1)+T(r+1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;
79 -             end
80 -             if c==1 && r==NRows %left side bottom corner
81 -                 T2(r,c)=(2*T(r,c+1)+2*T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;
82 -             end
83 -
84 -             if r==NRows && c==1 && c~=NCol %bottom side midpoints
85 -                 T2(r,c)=(T(r,c-1)+T(r,c+1)+T(r-1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;
86 -             end
87 -
88 -             if c==NCol && r~=NRows %the right side main, where west direction will be used twice
89 -                 T2(r,c)=(T(r,c-1)+T(r,c-1)+T(r+1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;
90 -             end
91 -             if r==NRows && c==NCol %right side bottom corner
92 -                 T2(r,c)=(T(r,c-1)+T(r,c-1)+T(r-1,c)+T(r-1,c)-(4-(1/Fo))*T(r,c))*Fo;
93 -             end
94 -         end
95 -     end

```

Figure 12 Implementing thermal diffusion formulae

This section of the code is where the thermal diffusion equations are used.

It begins with a while function (67), that checks when $T3$ and $T4$ are equal, and while that is true the code continues to run

Line 72 is an if function, for the nodes in the central domain of the plate, with various parameters that must be kept.

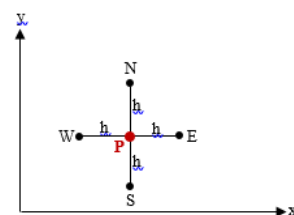
Firstly c must be more than one, c must be less than the number of columns, and r must not be equal to the number of rows. If they are all true the next line will be used.

the next sections are where $T2$ is calculated, $T2$ being the next time step's temperature matrix. This formula basically defines each point in the matrix that is being used.

$T(r,c+1)$ East

$T(r,c-1)$ West

$T(r+1,c)$ North



T(r-1,c) South

Figure 13 Coordinate Geometry

The first part (72-74) is to work out T2 where all four directions are available.

Next (76-79) is to work out the left side mid points, where only 3 directions are available, East, north and south, so East must be used twice.

Next (80-83) is to work out the left side bottom corner, where only North and East are known, so both must be used twice.

Next (84-86) is to work out the bottom mid-point

, where North, East and West are known, and North is used twice

Next (88-90) is to work out the right side midpoint, where North, West and South are known, so West is used twice.

And finally (91-95) is to work out the right bottom corner, where North, and west are known, so must be used twice.

3.9 Final Display functions:

```

97 -     i=i+1;
98 -     T2
99 -     T3=abs(T-T2)<0.1;
100 -     pause(0.01)
101 -     contourf(flip(T2),10)
102 -
103 -     colorbar
104 - end
105 - disp('the time to reach thermal equilibrium in seconds: ');
106 - Time_taken = i*TS
107 - T
108 - T2

```

Figure 14 Final display functions

This final part of code basically just produces the graph at the end of the code, as well as some final touches.

i=i+1 (97) - This part makes the number of iterations count up by 1 each time, to allow the code to calculate the temperature for each time step.

T2 (98) - This just tells the script to display T2 at the end of the code.

$T3 = \text{abs}(T - T2) < 0.1$ (99) – this part works out the maximum difference between T and T2, giving a tolerance parameter, and when it has a difference of less than 0.1 thermal equilibrium has been reached and the script stops running.

Pause(0.01) (100) – this is put in so that the graph is displayed while the script is running.

Contourf(flip(T2),10) (101) – this is the command for the script to display a graph of the results.

Colorbar (103) – this is to add a colorbar legend to the graph.

The final parts (105-108) simply display the time taken to reach thermal equilibrium as well as T and T2.

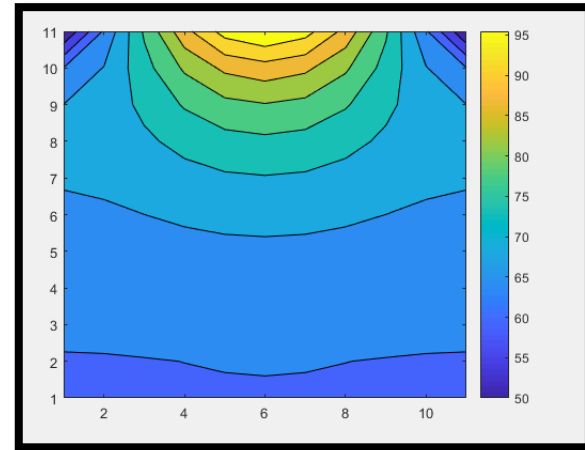


Figure 15 Final Graph showing thermal diffusion at set values

4.0 Results

Grid spacing(metres)	X (metres)	Y (metres)	time step(seconds)	Tolerance value	Equilibrium time(seconds)
1	10	10	3600	0.1	1119600
2	10	10	3600	0.1	1112400
5	10	10	3600	0.1	1058400
4	20	20	3600	0.1	698400
5	20	20	3600	0.1	712800
10	20	20	3600	0.1	594000
Grid spacing(metres)	X (metres)	Y (metres)	time step(seconds)	Tolerance value	Equilibrium time(seconds)
1	10	10	360	0.1	73080
1	10	10	720	0.1	142560
1	10	10	1080	0.1	322920
1	10	10	1440	0.1	535680
1	10	10	1800	0.1	678600
1	10	10	3600	0.1	1119600
Grid spacing(metres)	X (metres)	Y (metres)	time step(seconds)	Tolerance value	Equilibrium time(seconds)
1	10	10	3600	0.1	1119600
1	10	10	3600	0.01	2570400
1	10	10	3600	0.001	4021200
1	10	10	3600	0.0001	5472000
1	10	10	3600	0.00001	6922800
1	10	10	3600	0.000001	8377200

As shown above, various testing was performed on the code to compare the time to reach thermal equilibrium whilst changing one variable per test.

Firstly the X and Y dimensions were set to 10m, with the time step set at 1 hour, and the tolerance value set at 0.1. And the grid spacing was varied. From 1-5 and showed fairly similar results, this was

then repeated for x and y dimensions of 20m, with grid spacing varying from 4-10 and again fairly similar results were obtained. From the few results I have, I can infer that the grid spacing itself doesn't affect the time to reach equilibrium.

Second the grid spacing, X dimension and tolerance values were kept constant, while the time step was varied from 6 minutes – 60 minutes. The results from this vary massively, although I believe that they should ideally be very similar. I believe this is due to having a time step being too small, makes the difference between T and T2 too small, so it hits the tolerance value prematurely giving false results.

Finally everything was kept constant whilst varying the tolerance value from 0.1-0.000001.

The results obtained increase almost linearly as the tolerance value drops. So by then end the final equilibrium time is massive. This is due to the tolerance being very hard to hit once it becomes too small, so the plate must be heated for much longer before the variation drops that much.

5.0 Conclusion

5.1 Limitations of script

The Fourier number as explained above, the Fourier number is known as the Stability criteria for this code, if the Fourier number comes out with a value of over 0.25 the code will not work for a 2D problem, and similarly for a 3D model it must not exceed 0.125. Because of this the values in the formula to calculate the Fourier number have limitations.

As $Fo = (\alpha \cdot TS) / h^2$;

Where Alpha doesn't change and has a value of $6.4241e-5$. Therefore only h and TS can be changed. Because of this and the strict boundary of 0.25 for the Fourier number, h and TS have limitations. "h" cannot exceed a defined value, and the time step can't be too small.

Varying the time step is another limitation, as having a time step that is too small, makes the difference between T and T2 too small, so it hits the tolerance value prematurely giving false results.

The tolerance value has its limitations, as if it is too high it will be hit at the start of the code, long before thermal equilibrium is reached. But if it's too low the script will take a very long time to register thermal equilibrium so a compromise must be made.

My code has issues with values of "h" that are less than 1. The problem lies in that when "h" is below 1, it causes incorrect values for MidC, and therefore causes problems in the Time step and Fourier number calculations too.

5.2 Suggested improvements

The improvements that I'd suggest for this code are:

I would want to make the code work for 3d models, which could be achieved relatively easily by simply adding two new directions into the code (in and out), as well as decreasing the Fourier number to below 0.125, and by making slight changes to the calculations.

Another area that could be improved is to work on the limitations, to try and get around them, ideally by improving the code to accommodate.

The script could also be improved by adding the option to enter differing material properties, so that this could be modelled for more than just aluminium.

I would like to modify the section of the code that pertains to "MidC", as because of this my code has issues with values of "h" that are less than 1. The problem lies in that when "h" is below 1, it causes incorrect values for MidC, and therefore causes problems in the Time step and Fourier number calculations too.

And finally, I would like to work out how the code could be used to model some more complex geometries, such as parts with differing cross sections, or parts that are not simple rectangles.

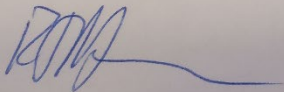
Appendix 6 – Letter from Richard Morgan

Dear Karim

Following our discussions on the topic, I am writing this note to confirm that recent industrial projects in which staff from the School of Engineering have been engaged, have highlighted the significant challenges many companies face when bonding structural elements. One such challenge is being able to accurately model heat transfer and dissipation in 'heat cured' adhesive applications.

A number of organisations have expressed interest in undertaking work to develop a thermal analysis model which would facilitate a greater understanding of their processes. It is envisaged that such models would allow process engineers to refine and optimise various process parameters, such as oven temperature, convection current speed and oven – cycle time.

Kind regards,



Richard Morgan

Programme Director – BEng Mechanical & Manufacturing Engineering

School of Engineering

University of Wales, Trinity Saint David.