UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Topic 14: Stochastic inversion of fire test data for the T-dependant thermal diffusivity of SA pine

by

Liza Stewart
21555575

Project (Civil Engineering) 458

Final Draft

Study leader: Prof N de Koker

October 2021

Declaration

Ek, die ondergetekende, verklaar hiermee dat die werk in hierdie verslag vervat, my eie oorspronklike werk is.

Signature: .....
L. Stewart

Datum: .....

Abstract

Stochastic inversion methods such as Markov Chain Monte-Carlo and optimisation methods were used to determine the thermal diffusivity of SA-Pine. Thermal diffusivity can be easily calculated from thermal conductivity, and conductivity can be more directly solved through modelling; the focus was therefore moved to the calculation of thermal conductivity. A finite element model was constructed to model a 100mm element exposed to the ISO 846 Fire curve. The model was compared to data previously obtained by S. van der Westhuyzen; this data was also used as the observed quantity in the formulation of the Bayes' model. Using Bayes' theorem of inverse problems, a posteriori probability distribution consisting of a likelihood probability and a prior probability was constructed. The finite element model was essential to the calculation of the likelihood probability. The prior probability was based off the thermal conductivity values from EN 1995:1-2-2004. New thermal conductivity values were estimated through Markov Chain Monte-Carlo inversion and the Maximum a Posteriori was found though optimisation. The Markov Chain Monte-Carlo and Maximum a Posteriori thermal conductivity were not equal, supporting the hypothesis that the distribution is not normal. The model was rerun with the new thermal conductivity estimates and the models were compared to the the measured data. The new estimated conductivity values resulted in more accurate modelling of the temperature in SA Pine when exposed to fire. From the new conductivity estimates combined with known values of density and specific heat capacity, the thermal diffusivity could be successfully calculated.

Contents

List of Figures

List of Tables

Nomenclature

Variables

κ	Thermal conductivity	$[\,\mathrm{W/m{\cdot}K}\,]$
α	Thermal diffusivity	$[\,\mathrm{m^2/s}\,]$
c_p	Heat capacity of pine	$[\,\mathrm{J/kg/K}\,]$
c_A	Heat capacity of air	$[\,\mathrm{J/kg/K}\,]$
ρ	Density	$[\,\mathrm{kg/m^3}\,]$

Chapter 1

Introduction

This chapter will introduce the problem that is addressed in this project. Previous similar projects will be investigated, and the value of this research will be presented.

1.1 Background and Motivation

Traditionally the thermal diffusivity, otherwise referred to as the α-value, of timber is based simply on the EN 1995:1-2-2004, further referred to as the Eurocode, or similar standards. This research project will aim to obtain the thermal diffusivity of cross-laminated SA-Pine timber by further analysing data originally obtained by van der Westhuyzen *et al.* (2020) for their study of the samples' charring rate.

The thermal diffusivity of timber is an unobservable quality that cannot be measured directly. However, thermal diffusivity is directly related to thermal conductivity (κ) and can be calculated when thermal conductivity is known. The majority of this report will therefore discuss the methods used to obtain the thermal conductivity or κ-values, and the diffusivity will be calculated when interpreting the results.

Conductivity can be related to measurements of temperature and time through differential models. When heat conduction is calculated using finite element methods, the process is usually simplified to a linear problem (Fish, 2007). Due to the changes in thermal conductivity of timber with temperature, as can be seen in EN 1995:1-2-2004 (page 49) and Figure 3.22, the conductivity cannot be linearly modelled. The problem therefore lends itself to being analysed by inversion techniques. This approach will allow us to obtain information about the conductivity based on the combined information assumed prior to measuring, further referred to as the prior, and the measured data. Statistical inversion delivers a probability distribution, thereby providing a collection of conductivity estimates and their corresponding probabilities. The approach

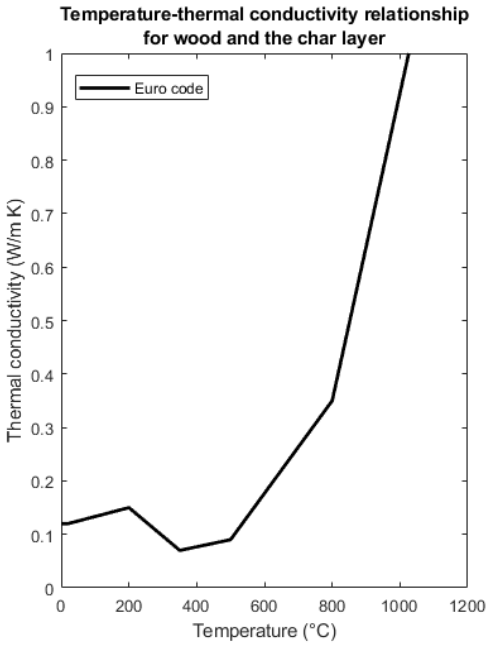


Figure 1.1: Standard temperature-thermal conductivity relationship for timber from (CEN, 2004)

taken in this report is unique due to the nature of data used, as the main purpose of this data's collection was to determine char rates and not thermal conductivity or diffusivity. The difference in research goals means that more assumptions had to be made, in contrast to a experiment that intended to determine the thermal conductivity and would have controlled those variables.

Currently, the fire rating of specific timber samples is based on fire tests conducted in a furnace. The furnace is kept at increasing temperatures corresponding with the Standard or ISO 834 fire curve as specified in ISO 834 ISO (1999). This process becomes very costly if it must be repeated every time timber is used for construction, as timber has seen increased usage for multiple story construction projects over the past decades. This increase is partially due to the sustainability of timber as a construction material: not only is it renewable but it also has a small carbon footprint (Salvadori, 2017). Obtaining and standardising thermal diffusivity values for different species of timber will assist in more accurate modelling. If modelling accuracy can be increased, the use of modelling to confirm fire tests or to replace small-scale fire tests will become more feasible.

1.2 Aim and objectives

During the course of the project, the student will aim to meet the following objectives:

1. Modify a time-dependant heat transfer Finite Element Model implemented in Matlab into an accurate and effective function;
2. Compare the model to the measured data;
3. Solve for the thermal diffusivity using Bayesian Inversion using the following methods:
   (a) Maximum a Posteriori
   (b) Markov-Chain Monte Carlo

1.3 Literature study

In their article *Simple Method to Determine the Diffusivity of Green Wood*, Frayssinhes *et al.* (2020) determine the global diffusivity of a Douglas fir green log using inverse identification methods. Their experiment involved submerging the log with K-type thermocouples attached at each end in a boiler filled with water at 60°C. The thermocouples were specifically placed to improve the accuracy of the diffusivity ratio calculation. The finite element model constructed for their calculations used linear interpolation with four-node quadrangle elements. An analytical model was also constructed using the heat conduction and diffusion equations. The thermal diffusivity was calculated by minimising the root mean square error between the slope of the measured data and the finite element model. This research intended to assist the peeling industry in making the pretreatment process more cost-effective. The method proved to be effective at determining the thermal diffusivity of green Douglas fir logs, as the κ-values obtained were comparable to those from literature. A crucial difference remains as the temperatures at which these experiments were conducted as well as the final usage of the data differ greatly.

Wagner *et al.* wrote an article in March 2021 where Bayesian model inversion was used with stochastic spectral embedding. Spectral embedding, or spectral likelihood expansion, was used to solve for the likelihood function in such a way that sampling is not necessary. Similarly to this report, the research of Wagner *et al.* (2021) involved the creation and use of a Bayesian model to solve a heat diffusion problem. The likelihood function used for this analysis was similar to the likelihood function explained later in the report. The focus of the article was to determine the accuracy of stochastic spectral likelihood embedding, stochastic spectral embedding and spectral likelihood functions to

calculate the thermal diffusivity. A Markov Chain Monte-Carlo analysis was done to compare to values from other methods. Although the Markov Chain Monte-Carlo method was not discussed in detail in the article, the usage of Bayesian modelling to create a likelihood function that the MCMC analysis could be done on is similar to the process followed throughout this project.

Chapter 2

Technical Foundation

In this chapter, detailed explanations of the algorithms and concepts applied throughout the project are presented. This chapter aims to enable the reader to better understand the implementation and reasoning for the use of specific methods.

2.1 Thermal properties of fire exposed timber

Specific thermal properties of timber must be thoroughly understood to allow for accurate modelling and interpretation of the results. After 100°C, the temperature exposure causes the timber structure to break down in such a way that it will not return to the original state after cooling. This breakdown is generally referred to as thermal degradation, which can be broken into four categories (White and Dietenberger, 2001) (Shi and Chew, 2021). These categories are based on the different reactions that occur, and are briefly summarised below:

- Above 100 °C and below 200 °C, the moisture in the timber is evaporated and other non-combustible gasses are released.
- Between 200 °C and 300 °C, carbon monoxide is released in significant quantities, and some of the timber components undergo a change in chemical composition due to the high temperatures.
- Temperatures above 300 °C and before 450 °C are significant due to the amount of flammable volatiles released and the break of carbon linkages at 370 °C.
- At temperatures higher than 450 °C, the remaining timber is charred and any further degradation is due to oxidation from carbon monoxide, carbon dioxide, and water.

Timber undergoes physical and chemical changes as temperature changes; it therefore makes sense that its thermal properties such as its specific heat capacity (c_p) and its density (ρ) of timber also varies with temperature. As the thermal conductivity and thermal diffusivity will only be assessed at specific temperatures, the specific heat and density at only those temperatures will be summarised below.

Temp (°C)	c_p $(J/kg/K)$	ρ (kg/m^3)	$c_p\rho$ $(kJ/kg/K)$
0	1530	479	732.8
60	1652	479	791.3
100	13600	479	6514.4
140	2090	408	852.7
200	2000	408	816.0
350	850	220	187.0
500	1200	142	170.4
800	1650	110	181.5
1200	1650	100	165.0

Table 2.1: c_p and ρ at different temperatures

All the thermal properties of various timber species at high temperatures were discussed in a research article by Shi and Chew (2021). In their article, they provide a general expression for the thermal conductivity of softwoods (Equation 2.1) and char (Equation 2.2) with all temperatures in Kelvin.

$$\kappa = 0.124 + 0.8432 \times 10^{-4}(\mathrm{T} - 293) \tag{2.1}$$

$$\kappa = 0.091 + 8.2 \times 10^{-4}\mathrm{T} \tag{2.2}$$

For this project, the temperature of at which charring occurs will be taken as 300 °C following the research of van der Westhuyzen *et al.*. Using the equations of Shi and Chew, the following κ-values (Equation 2.3) are obtained.

$$\kappa = \begin{bmatrix} 0.1223 \\ 0.1274 \\ 0.1308 \\ 0.1341 \\ 0.1392 \\ 0.1518 \\ 0.7250 \\ 0.9710 \\ 1.2990 \end{bmatrix} \begin{matrix} 0\ ^\circ\mathrm{C} \\ 60\ ^\circ\mathrm{C} \\ 100\ ^\circ\mathrm{C} \\ 140\ ^\circ\mathrm{C} \\ 200\ ^\circ\mathrm{C} \\ 350\ ^\circ\mathrm{C} \\ 500\ ^\circ\mathrm{C} \\ 800\ ^\circ\mathrm{C} \\ 1200\ ^\circ\mathrm{C} \end{matrix} \tag{2.3}$$

2.1.1 Heat conduction and diffusion

Heat is transferred within an element if a temperature gradient present, the rate that heat transfers is directly influenced by the thermal conductivity of the material. This project focuses on heat transferred via conduction, but heat can also be transferred in the form of thermal radiation or convection Fish (2007). The heat conduction equation or Fourier's Law of heat conduction (Fourier, 1878) can be simply expressed in Equation 2.4. Where heat flux is expressed as a partial differential equation dependant on the temperature and thickness of the element.

$$q = k\frac{\Delta T}{L} \tag{2.4}$$

More information can be added if the temperature only varies in one direction. The thermal conductivity can also be expressed as a negative since heat always flows from warm to cold. Fourier's Law can then be written as in Equation 2.5:

$$q = -k\frac{\partial T}{\partial x} \tag{2.5}$$

Derivation of the heat conduction equation over time leads to the heat diffusivity equation, Equation 2.6:

$$\frac{\partial T}{\partial t} = -\alpha\frac{\partial^2 T}{\partial x^2} \tag{2.6}$$

where

$$\alpha = \frac{\kappa}{c_p \rho} \tag{2.7}$$

As is evident in Equation 2.7, knowing the specific heat capacity and density of the timber at different temperatures is crucial. The values in Table 2.1 will be used in conjunction with the κ-values determined through simulation to calculate new thermal diffusivity values.

2.2 Finite Element Method

The finite element method (or finite element analysis) is used when the behaviour of a large element cannot be accurately depicted by a simple mathematical equation. Modelling the behaviour of large and complex systems is a near impossible task without breaking the problem into smaller understandable parts. The finite element method enables engineers and scientists to break a large problem or element into comprehend-able and definable parts (Zienkiewicz and Taylor, 2000). This allows approximations and assumptions to be made at a small scale without affecting the large scale significantly, such

as assuming linear behaviour between nodes. To understand simply why it works, you could simply look at your computer screen. The screen consists of pixels that are simplified to one colour, but the collection of pixels that is seen from the user perspective can show complex forms and pattens. Each pixel was simplified and approximated, but the end result is sufficiently accurate due to the sheer number of pixels. Assumptions made on a smaller scale have a lesser effect on the final answer than the same assumptions made on a large scale would have had. If the whole screen was one pixel and only the majority colour was assumed true, the whole screen would be white and that is clearly incorrect. It follows that the accuracy of a finite element model is highly dependant on the number of elements used and only improves with more elements. Unfortunately, the computational time needed to analyse a model with a thousand elements is significantly more than needed for a model with fewer elements. The fine balance between sufficient accuracy and short enough computational time should always be kept in mind.

2.2.1 Origin

The finite element method (FEM) used today is the sum of decades of research. In an article by Gupta and Meek (1996), the authors discuss the five main contributors to the finite element method. They report that the idea behind the finite element method was initially explored in the 1943 article by Courant. Courant acknowledges the complex nature of mathematical problems in his first paragraph by stating: "Mathematics is an indivisible organism uniting theoretical contemplation and active application." He goes on to discuss the variational method created by Lord Rayleigh and Walther Ritz and the shortcomings of their methods. He concludes that using few elements does not provide accurate local estimations, but instead the interpolation will provide approximations of the actual quantities that are sufficient for engineering problems. As his paper did not go into depth into the calculations, Courant cannot be accredited with the full development but rather with the starting point. Gupta and Meek (1996) further points out that in 1954, J. Argyris published an article where the details of his calculations are clear and matrix formulation can be seen. The finite element method as known to day were directly defined and clarified in a textbook by Zienkiewicz in 1957. This textbook is now in its fifth edition of print and has been continuously updated and was used for this project.

2.2.2 Boundary constraints

An important part of finite element models is the method in which the known information is embedded into the model. The main method is by defining

boundary conditions at known points. The two types of boundary conditions used in this project are Dirichlet boundaries and Newmann boundaries.

Named after Johann P.G.L Dirichlet (Cheng and Cheng, 2005), Dirichlet boundaries force the solution function to be equal to the prescribed value at the boundary. Newmann boundaries prescribe that the derivative of the solution function be equal to the predetermined value. In the context of this problem, the Dirichlet boundaries prescribe the temperatures at the boundaries and the Newmann boundaries prescribe the heat flux at the boundaries.

2.3 Bayes' theorem of inverse problems

The method of statistical inversion is dependant on a fundamental understanding of the Bayes' theorem of inverse problems. This understanding was obtained through studying Chapter 3 of statistical and Computational Inverse problems by Kaipio and Somersalo (2005), further referred to as Kaipio. There are four principles of statistical inversion that are essential to the thorough understanding of these models.

Firstly is the principle that any variable in the model needs to be modelled as a random variable. This randomness is based on the extent of information that is available. To ensure that the extent of knowledge is accurately portrayed in the model, the extent of knowledge will be coded into the probability distributions assigned to the different variables. Finally, it needs to be understood that the solution of a statistical inversion is a posterior probability distribution. A generalised equation of Bayes' theorem can be seen in 2.8 taken from Kaipio.

$$\pi^*(x) = \pi(x|y_{\text{observed}}) = \frac{\pi_{\text{pr}}(x)\pi(y_{\text{observed}}|x)}{\pi(y_{\text{observed}})} \tag{2.8}$$

Further application of Bayes' theorem of inverse problems to the project at hand is discussed in Section 3.3

2.4 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a method of integration that will be used to determine the mean of the κ-values at specific temperatures. Markov Chain Monte Carlo is a method that was created by combining the concept of Monte Carlo sampling and a Markov Chain. To fully understand MCMC, its underlying methods must be investigated further. For a better understanding of this concept, *Introducing Markov chain Monte Carlo* by Gilks *et al.*, Kaipio, and various websites (Brownlee, 2019), (Wiecki, 2015) were consulted.

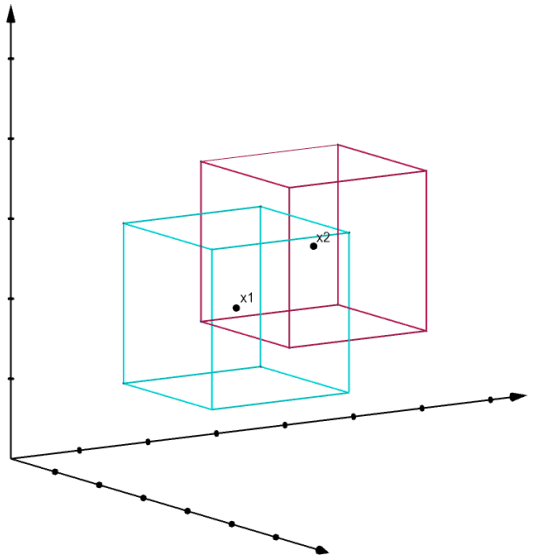


Figure 2.1: Two-dimensional example of Markov Chain application (TODO)

2.4.1 Markov Chains

The core principle of a Markov chain is that the next value (x_{n+1}) in a sequence is dependent on the current value(x_n). A step size that indicates the range within the next point falls in. Values are then randomly generated but restricted to be within this range. This concept can be visualised as follows: the accepted point (x_1) is in the centre of a cube. The next possible random point is randomly generated but still within the cube(our search range). After this next number is selected, the cube moves such that the new point(x_2) is now the centre, and so it continues. See Figure 2.1 for clarification. The above example simplifies the concept, but this understanding can now be expanded. If every coordinate direction in the aforementioned simple example is seen as a single entry in the x vector, then the example has three independent values. More or fewer values can be used depending on the problem. Another level of complication can be added if it is taken into account that every point in the cube is no longer equally likely. A distribution within the cube can be chosen, for example simply a normal distribution The shape of the cube then warps into a stranger shape with points closer to the center being more likely choices and the edges being less likely.

The purpose of a Markov Chain is for the chain to converge to a distribution and be independent on the very initial estimation. In principle, it should then reach a near stationary distribution. Since Markov Chains are not used if we know the answer, a way to determine when values are no longer affected by the initial estimate is needed Gilks *et al.* (1996). The simple proposed solution is the concept of burn-in. The concept of conventional burn-in for usage in

Markov Chains are disputed as the Markov Chain itself is created in such a way that values are only directly dependent on the value immediately before them Meyn and Tweedie (1993). Burn-in in the Markov Chain sense can simply be referred to as the removal of the initial samples of low probability to increase the accuracy of the average taken after all the iterations Cook (2016).

2.4.2 Monte Carlo Integration

Monte Carlo integration is used to evaluate a probability distribution that cannot be solved simply. The evaluation is done by drawing a collection of random values from the distribution. These values are then used as the sample, and a sample mean is taken. The arithmetic sample mean can be used to approximate the population mean per the law of large numbers (Gilks *et al.*, 1996).

2.4.3 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is one of the available simulation methods based on the MCMC principles. For this project, the Metropolis-Hastings algorithm was chosen above the Gibbs-sampler as the Gibbs sampler can be relatively slow Murphy (2012).

All of the random samples generated by the Monte Carlo integration can not be immediately accepted. Here, the acceptance criterion comes into play. There are multiple options for how a posterior is deemed acceptable; these are elaborated on in the book *Monte Carlo Statistical Methods* by Robert and Casella. The most general acceptance criterion is set out in Equation 2.9 and comes from Kaipio.

$$\begin{aligned} &\text{if} \quad \frac{\pi(x_2)}{\pi(x_1)} > 1 \quad \text{Accept automatically} \\ &\text{or} \quad \frac{\pi(x_2)}{\pi(x_1)} > \text{rand} \quad \text{Accept with probability } < 1.0 \\ &\text{else reject and sample new } x_2 \end{aligned} \tag{2.9}$$

2.5 Maximum a Posteriori

To ensure a thorough investigation of the thermal diffusivity of timber the posteriori function will also be optimised to obtain the value with the highest probability, maximum a posteriori, in addition to the mean. As the implementation will be done using built-in optimisation functions in Matlab the technical aspects will briefly explain how that function works. The function used is `fminsearch.m`. In the name of the function, it can be seen that the

function will be minimised. This seems counter-intuitive since the intention is to find the maximum of the function. Luckily the minimum is simply a negative maximum.

According to the `fminsearch` Matlab documentation (`https://ch.mathworks.com/help/matlab/ref/fminsearch.html`) , the algorithm is based on the Nelder-Mead algorithm (Lagarias *et al.*, 1998). The Nelder-Mead algorithm is an unconstrained heuristic direct search method. The heuristic aspect of the algorithm means that it searches the function space in a guided random manner.

Nelder-Mead uses a simplex, a triangle in the dimension needed, to find the minimum. In two-dimension, the simplex is simply a triangle, where each point has a x and y coordinate and is notated as. A starting value needs to be provided to the algorithm, this will then be one of the points in the first simplex. All the points on the simplex are ordered from best to worst, where the smallest value ($f(x)$) is best if minimising. In the two-dimensional example, the centroid of the two best points (x_l, x_s) are calculated, from there using the value of the function at the simplexes and the centroid transformations can be performed to find even smaller values. The transformations are applied to change the worst point (x_h) until it is no longer the worst.

The transformations are performed in a specific order. The first transformation is reflection. Point x_h is mirrored around the line that passes through the centroid and x_l and x_s to create x_r. The idea behind this is that the values are lower in that direction so the algorithm attempts to take the biggest stride possible to a lower value. If x_r is better than x_r the triangle is expanded and x_e is even further in the assumed minimising direction. If x_r is not better than x_h, the reflected point is rejected and the original x_3 is contracted closer to x_1 and x_2. This contracted point (x_c) should ideally have a smaller value than x_h, if not a shrink contraction needs to be performed. In a shrink contraction, x_h and x_s are redefined and pulled closer to x_l with hopes of finding the minimum near the best point.

The result is a simplex that moves over a surface in search of the lowest point and as it reaches that point it shrinks until all the points are practically at the same place. When this happens or the points are close enough to each other as defined when initializing the code the algorithm stops. Below in Figure 2.5, the different transformations can be seen.

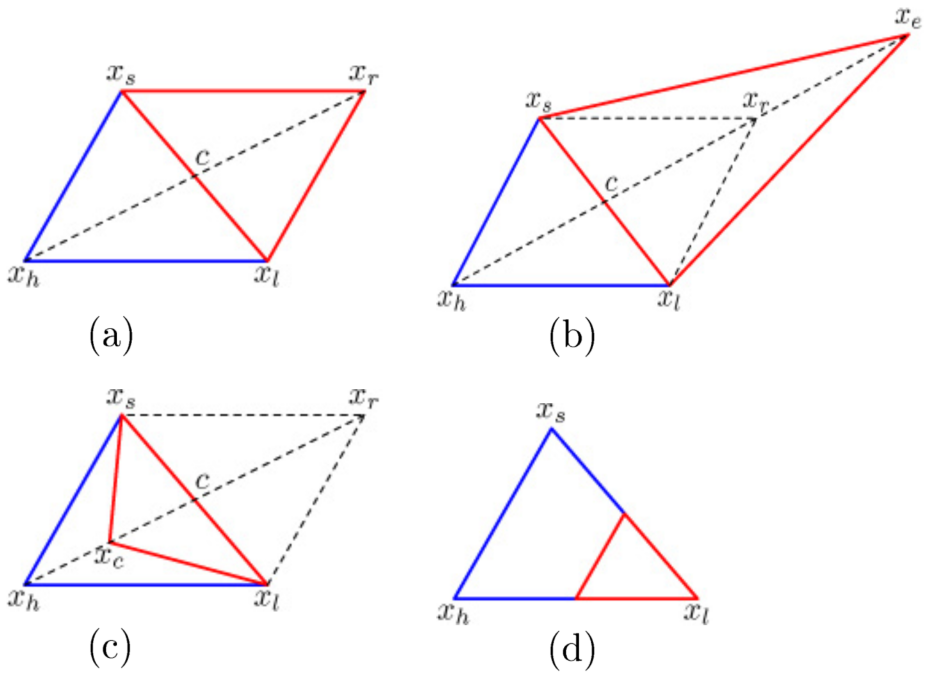


Figure 2.2: Visual representation of Nedler-Mead transformations a) Reflection, b) Expansion, c) Contraction, d) Shrink contraction

Chapter 3

Implementation

This chapter will elaborate on the test data used as well as the process that was followed to achieve the results in Chapter 4.

3.1 Existing data

The data used was acquired by van der Westhuyzen *et al.* (2020) for an article assessing the charring rate of both SA-Pine and Eucalyptus. For the purpose of this project, only the data obtained from the SA-Pine test was considered and analysed.

3.1.1 Summary of test

The test was conducted on a sample panel of 100 mm by 0.9 m ×0.9 m cross-laminated SA-pine. This sample was then divided into nine cubes of 100 mm ×100 mm ×100 mm. Each cube was fitted with seven Type K-thermocouples placed at consecutive 16.5 mm drilled holes, as can be seen in Figure 3.1. The test panel was tested in a furnace and was exposed to the standard ISO 834 Fire curve (Figure 3.2 and Equation3.1) on one side and room temperature on the other. The panel was exposed to the fire curve for 50 minutes, at which stage near complete de-lamination was observed and the test ended.

$$T = 20 + 345 \log_{10}(8t_{min} + 1) \tag{3.1}$$

In Figure 3.1.1 the measured temperatures at different depths in the sample can be seen. These same depths will be used in the model and throughout the project to analyse the accuracy of the modelling that was done.

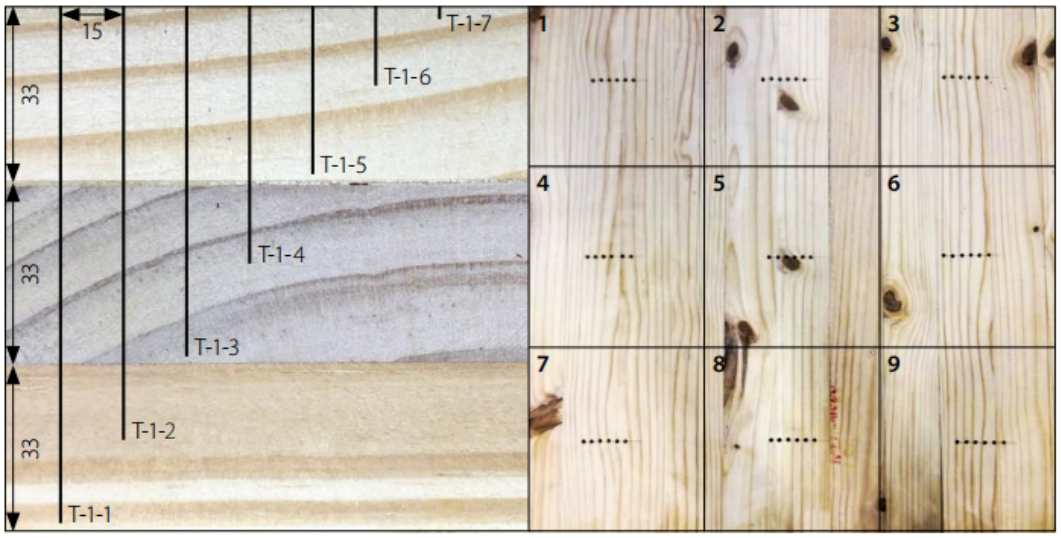


Figure 3.1: Thermocouple layout in test conducted by van der Westhuyzen *et al.* (2020) cross-section (left) and overall layout (right)

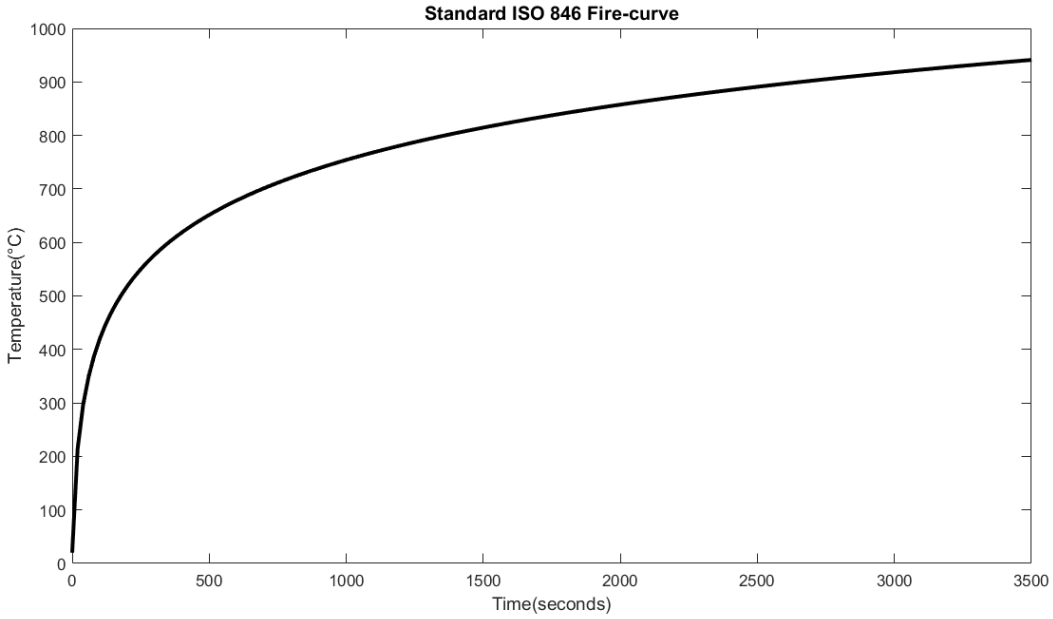


Figure 3.2: Standard ISO fire curve

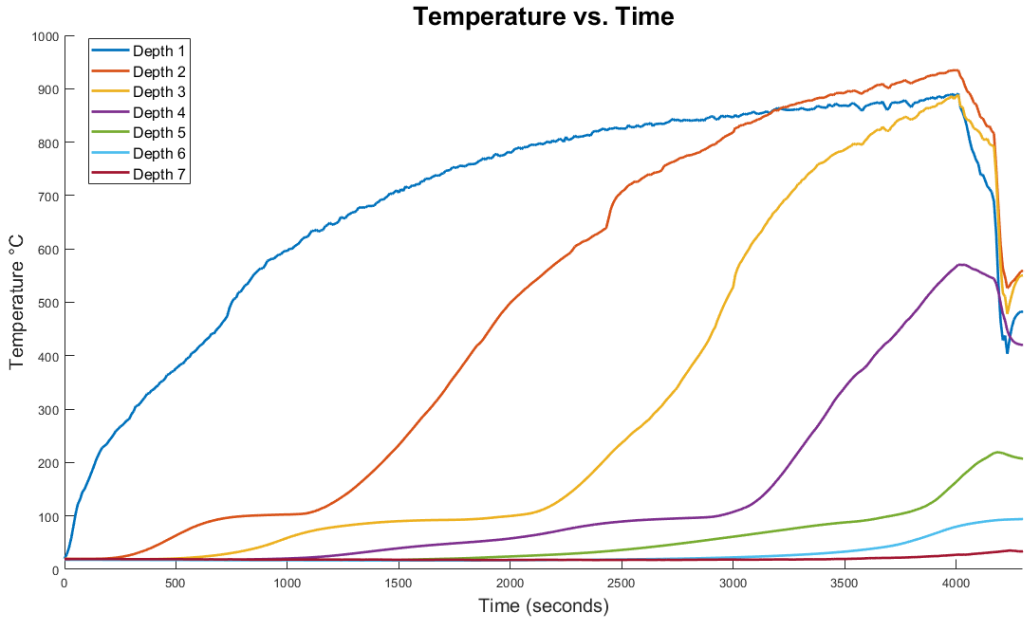


Figure 3.3: Average of measured temperatures at the different depths

3.1.2 Potential inaccuracies

As with most tests, everything is not always perfect. The potential inaccuracies are discussed below. In the data, it was observed that two of the thermocouples malfunctioned during testing and could not be considered This malfunction required that two of the depth measurements were no longer the average between nine samples but instead the average between eight. Another inaccuracy that could potentially influence the accuracy of the final result is the accuracy of the depth of the holes in which the thermocouples were placed.

There is also debate about the significance of the contribution of the timber burning to the temperature inside the furnace. For the purposes of this project, it will be assumed that the timber burning does not contribute to the temperature inside the furnace.

3.2 Finite Element Modelling

A one-dimensional finite element model that simulates what we expect to obtain from the fire tests based on the simplified κ-values provided in EN 1995:1-2-2004 (CEN, 2004) is modified into a function. This function should provide the temperature of the modelled element based on a specified location and thermal conductivity. The derivation and adaptation of the model are expanded on below.

3.2.1 Existing Model

For this project, an existing finite element model of time-dependant conductive heat transfer implemented in Matlab by Prof. N de Koker was modified for usage in the Bayes' theorem 2.8. This model is used to determine the likelihood function. The current model uses the κ-values as well as the specific heat specified in EN 1995:1-2-2004.

The model discretises the wooden element into 32 first order one-dimensional elements. This number of elements was sufficient to obtain converged results. The model is a one-dimensional finite element model that takes time differentiation into account.

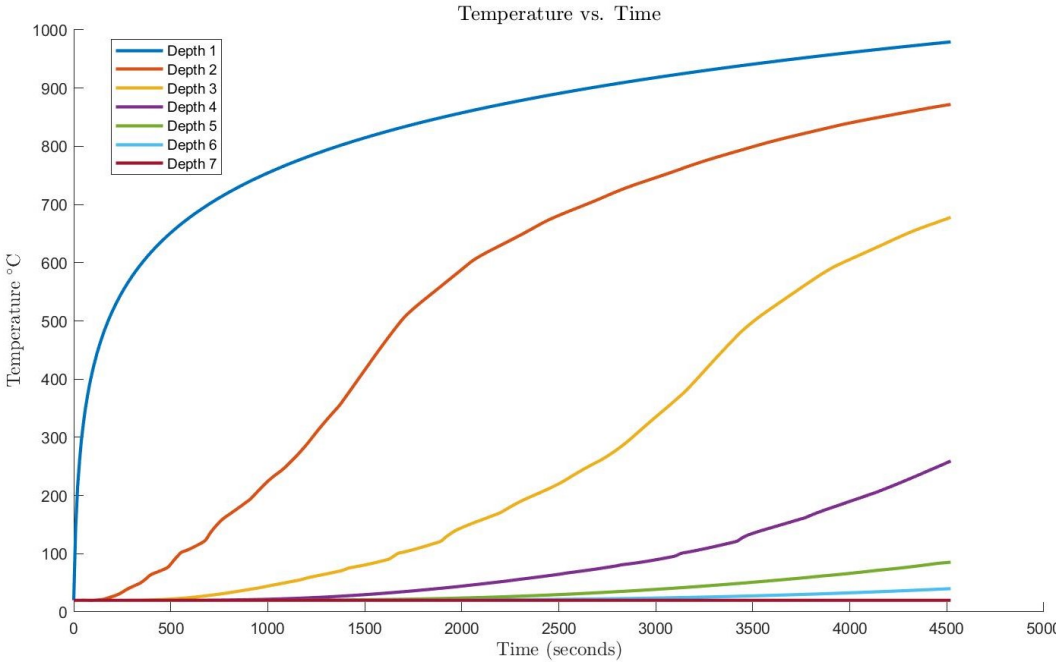


Figure 3.4: Output of finite element model using κ-values as indicated in EN 1995:1-2-2004

3.2.2 Derivation

Assumptions were made to simplify the model, they were as follows

1. The air on the side of fire follows the temperature of the fire curve.
2. The air on the cold side remains at 20°C.

Stationary heat conduction

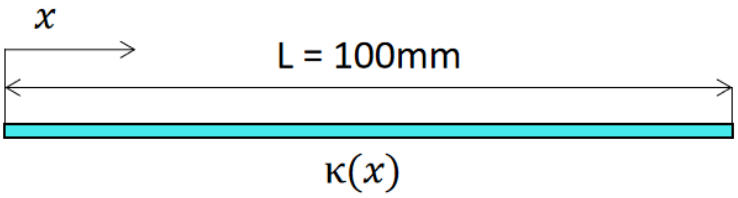


Figure 3.5: Visualisation of model in one-dimension

Based on Hughes (1987) and class notes from Informatics 314, the derivation started as a one-dimensional stationary heat conduction problem with the energy balance equation and the heat conduction equation.

$$q_{,x} - f = 0 \ ...(1) \qquad q = -\kappa u_{,x} \ ...(2) \tag{3.2}$$

Integrating Equation 3.2 (1) over the length of the element(shown in Figure 3.2.2) and introducing a weighting function $w(x)$ we obtain 3.3. Since the derivative of $w(0)$ is known and $q_{,x}$ is unknown. The first term in 3.3 is integrated by parts. After the integration by parts and substituting q with 3.2 (2), Equation 3.4 is created.

$$\int_{x=0}^{L} w q_{,x} dx - \int_{x=0}^{L} w f dx = 0 \tag{3.3}$$

$$\int_{x=0}^{L} w k u_{,x} dx + \int_{x=0}^{L} w f dx - w q|_0^L = 0 \tag{3.4}$$

In Equation 3.4 the u and w need to be defined. Assuming $u \approx u^h$ and $w \approx w^h$ and using the basis function (N^A), Equation 3.5 is obtained.

$$\begin{aligned} u_e^h &= \sum_B N^B d^B \quad ; \quad w_e^h = \sum_A N^A c^A \\ u_{e,x}^h &= \sum_B N_{,x}^B d^B \quad ; \quad w_{e,x}^h = \sum_A N_{,x}^A c^A \end{aligned} \tag{3.5}$$

Substituting the u and w functions back, we obtain the Galerkin weak form shown in Equation 3.6. In Equation3.6 the A_N refers to the nodes that have Newmann boundaries ,explained in 2.2). The variables c^A and d^B are independent of x and can therefore be taken out of the integral. The sum over A and B are also taken out of the integral. After the summing is applied a matrix of all the possible combinations between A and B can be used to replace the sum.

The resulting matrices are shown in Equation 3.7. When written in matrix form the summing is implied, if matrix form is not written then the expression referrers to the terms that will still be summed.

$$\begin{aligned} \sum_e \int_{\Omega_e} w^h_{e,x} k u^h_{e,x} dx + \sum_e \int_{\Omega_e} w^h_e f dx - w(L)q_L + w(0)q_0 &= 0 \\ \int_{\Omega_e} \sum_A \sum_B N^A_{,x} c^A k N^B_{,x} d^B dx + \int_{\Omega_e} \sum_A N^A c^A f dx - \sum_{A \in A_N} c^A q^A &= 0 \end{aligned} \tag{3.6}$$

$$\mathbf{c}_e^T \mathbf{K}_e^{AB} \mathbf{d}_e + \mathbf{c}_e^T \mathbf{F}_e^f - \mathbf{c}_e^T \mathbf{F}_e^q \tag{3.7}$$

where

$$\begin{aligned} K_e^{AB} &= \int_{\Omega_e} N^A_{,x} N^B_{,x} k dx \\ F_e^{Af} &= \int_{\Omega_e} N^A_{,x} f dx \\ F_e^{Aq} &= \begin{cases} q(x_N) & \text{for } x \in \Gamma_N \\ 0 & \text{for other} \end{cases} \end{aligned}$$

Boundaries

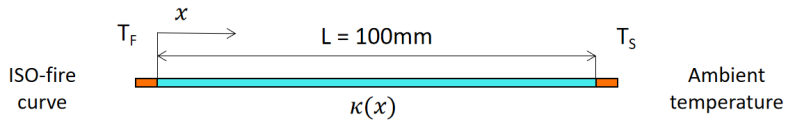


Figure 3.6: Visualisation of one-dimensional model with air elements and external conditions added

One of the additional difficulties in modelling is that the known boundary temperatures are measured in the air of the furnace and assumed for the air outside the furnace. Initially, a long element of air was modelled. This element is then overlaid with the timber element to effectively add an air element it each end of the timber element as depicted in 3.2.2 In the air elements, both heat convection and heat radiation need to be taken into account. The equations for heat convection and radiation can be seen in Equation 3.8 and 3.9.

$$\begin{aligned} q_{\text{adv}} = \nu \rho_A c_A \Delta T &= h \Delta T \\ \nu &= \text{velocity m/s} \\ \rho_A &= \text{density of air} \\ c_\rho &= \text{heat capacity of air} \end{aligned} \tag{3.8}$$

$$
\begin{aligned}
q_{\text{rad}} &= \varepsilon\sigma\phi\left(T_f^4 - T_S^4\right) \\
\varepsilon &= \text{emissivity} \\
\sigma &= \text{Stefan-Boltzmann constant} \\
&\quad 5.670e^{-8}[W/(m^2K^4)] \\
\phi &= \text{view factor; } 1 \quad \text{here}
\end{aligned}
\tag{3.9}
$$

Given that T_S and T_F are known a new equivalent heat flux value can be calculated as in Equation 3.10.

$$
q_{\text{con}}^{\text{equiv}} = \kappa^{\text{equiv}}\frac{\Delta T}{\Delta L} = q_{\text{rad}} + q_{\text{adv}} \tag{3.10}
$$

Due to the clear relationship between heat flux and thermal diffusivity the equivalent diffusivity(κ^{equiv}) can be calculated as shown below in Equation 3.11.

$$
\begin{aligned}
\kappa^{\text{equiv}} &= \frac{[q_{\text{rad}} + q_{\text{adv}}]}{\Delta T} \\
&= \frac{\epsilon\sigma\phi\Delta\left(T^4\right)}{\Delta T/\Delta L} + h\Delta L \\
&= \frac{\epsilon\sigma\Delta\left(T^4\right)}{\Delta T} + h
\end{aligned}
\tag{3.11}
$$

One-dimensional diffusion

The concept of heat diffusion is thoroughly explained in Section 2.1.1 below the mathematical application is explained and steps that were taken to obtain the final model is shown.

$$
q_{,x} - f = \frac{\partial Q}{\partial t} = c_p\rho\frac{\partial u}{\partial t} \quad ...(1) \qquad q = -\kappa\frac{\partial u}{\partial x} \quad ...(2) \tag{3.12}
$$

Substituting Equation 3.12(2) into 3.12 and taking the derivative as indicated ($q_{,x}$) gives Equation 3.13. As previously disccussed heat conduction (α) is heat diffusion (κ) divided by specific heat (c_p).

$$
\begin{aligned}
\therefore -\kappa\frac{\partial^2 u}{\partial x^2} - f = c_p\rho\frac{\partial u}{\partial t} \rightarrow f = 0: \\
\frac{\partial^2 u}{\partial x^2} = -\frac{c_p\rho}{\kappa}\frac{\partial u}{\partial t} \\
\text{or} \\
\frac{\partial u}{\partial t} = -\alpha\frac{\partial^2 u}{\partial x^2}
\end{aligned}
\tag{3.13}
$$

Let $c_p\rho = \lambda$

$$\therefore -\kappa u_{,xx} - \lambda u_{,t} = f \tag{3.14}$$

Then:

$$\int_0^L wq_{,x}\mathrm{d}x - \int_0^L w\lambda u_{,t}\mathrm{d}x - \int_0^L wf\mathrm{d}x = 0 \tag{3.15}$$

Similar to what was done in 3.5 a special approximation is made to obtain Equation 3.16.

$$u \approx u^h \rightarrow u_e^h = \sum_A N^A d^A \quad ; \quad w_e^h = \sum_A N^A L^A \tag{3.16}$$

From the above equations the below matrix formulation could be assembled:

$$\mathbf{c}^T\boldsymbol{\kappa}\mathbf{d} + \mathbf{c}^T\mathbf{M}\dot{\mathbf{d}} = \mathbf{c}^T\mathbf{F} \tag{3.17}$$

Where solving for $\mathbf{d}$ and $\dot{\mathbf{d}}$ is the main goal. By setting $\mathbf{d}_0$ equal to the initial boundary conditions and setting $\mathbf{V}_0 = 0$ and integrating over time according to v-form time integration in Chapter 8 in *The Finite Element Method* by Hughes shown in detail in Appendix C.

Finally by combining the boundary conditions, stationary heat conduction and the one-dimensional heat diffusion models the below formulation in Equation 3.18.

$$\mathbf{K}\cdot\mathbf{d} + \mathbf{M}\cdot\dot{\mathbf{d}} = \mathbf{F}' - \mathbf{F}^{Ke} - \mathbf{F}^{Me} = \mathbf{F} \tag{3.18}$$

This can be solved as shown below in Equation 3.19.

$$\begin{aligned}
\tilde{\mathbf{d}}_{n+1} &= \mathbf{d}_n + (1-\alpha)\Delta t\mathbf{v}_n \\
(\mathbf{M} + \alpha\Delta t\mathbf{K})\mathbf{v}_{n+1} &= \mathbf{F}_{n+1} - \mathbf{K}\tilde{\mathbf{d}}_{n+1} \\
\rightarrow \mathbf{v}_{n+1} &= (\mathbf{M} + \alpha\Delta t\mathbf{K})^{-1}(\mathbf{F}_{n+1} - \mathbf{K}\tilde{\mathbf{d}}_{n+1}) \\
\rightarrow \mathbf{d}_{n+1} &= \tilde{\mathbf{d}}_{n+1} + \alpha\Delta t\mathbf{v}_{n+1} \\
\mathbf{v} &= \dot{\mathbf{d}}
\end{aligned} \tag{3.19}$$

All of these derivations were the basis on which the existing model was built. Understanding of this model was essential to ensure that the adaptation was done correctly.

3.2.3 Adapted Model

The model was changed into a function that takes κ-values and provides a new temperature distribution over the elements for the different κ-values. The code for the adapted model function, `model_kinput.m` can be seen in Appendix A.

The focus of the adaptation was the `main.m` script in the original program. Within the `main.m` function all of the global variables used were initially defined. The defining of variables was moved into a new function `instantiate_all.m`, this function defined all the global variables used within the original `main.m` as well as the measured temperature data. Here the thickness of the timber element was defined as 99 mm and was divided into 30 elements with 2 nodes per element. Two additional elements were also added to incorporate the air elements as explained in Section 3.2.2. The global variable `kpine` was left uninstantiated, instead it will be fed to the new adapted model (`model_kinput.m`). The timesteps of the model was also set to 10 seconds to coincide with the time-steps in the measurements Importantly the `instantiate_all.m` function was called only once and outside of `model_kinput.m`. This was done to ensure that the model remains the same size throughout. Inside `model_kinput.m` the global `kpine` variable was set equal to the `kinput` variable fed into the model. This adaptation allowed the model to be called within the posterior distribution calculation with different κ-values each time.

3.3 Inversion method

The basis of the stochastic analysis is the adapted Bayesian equation 3.20 below. Each aspect of this equation below will be further explored in the sections below.

$$\pi^*(\mathbf{x}|T) \propto \exp\left(-\frac{(\boldsymbol{\mu}-\mathbf{x})^2}{2\sigma_\mu^2}\right) \cdot \exp\left(-\frac{(\mathbf{T}-M(x))^2}{2\sigma_{\text{temp}}^2}\right) \tag{3.20}$$

3.3.1 Prior probability

$$\pi(x) \propto \exp\left(-\frac{(\mu-x)^2}{2\sigma_\mu^2}\right) \tag{3.21}$$

The prior probability function (Equation 3.21) is based on the κ-values assumed prior to any simulation or analysis. The σ_μ in this equation was assumed to be equal to 0.13 W/m·K, In this case, the prior values are indicated as μ and refer to the vector of κ-values(3.22) at specific temperatures as indicated. The temperatures are not exactly similar to the temperatures used in Euro code, this is due to an attempt at modelling the effect of evaporation at 100 °C.

$$\mu = \begin{bmatrix} 0.12 \\ 0.12 \\ 0.12 \\ 0.12 \\ 0.15 \\ 0.07 \\ 0.09 \\ 0.35 \\ 1.5 \end{bmatrix} \begin{matrix} 0\ ^\circ\text{C} \\ 60\ ^\circ\text{C} \\ 100\ ^\circ\text{C} \\ 140\ ^\circ\text{C} \\ 200\ ^\circ\text{C} \\ 350\ ^\circ\text{C} \\ 500\ ^\circ\text{C} \\ 800\ ^\circ\text{C} \\ 1200\ ^\circ\text{C} \end{matrix} \tag{3.22}$$

The x (in Equation 3.21) refers to a vector of randomised κ-values that correspond with the same temperatures as the values in the μ vector. The first iteration of randomised κ-values are generated by creating a random perturbation of the μ vector. By multiplying the μ vector with $((0.5 + \text{rand}) \cdot 1.5)$ the first values of x are guaranteed to be within an acceptable range of the prior values. The process of obtaining the x vector after the first iteration is discussed later in section 3.3.3.

Initially, the program was written to generate completely random new values for the first iteration of x. This later proved to not only be unnecessary but also made the process less accurate as there was a larger burn-in period before the values were anywhere near the actual solution. To increase the accuracy and reduce the number of times the program needed to run to produce a sufficient number of accurate samples, the program was changed to the current method. The prior function in this case was relatively easy to generate and incorporate into the program as a well-defined list of prior values exists.

3.3.2 Likelihood probability function

$$\pi(T) \propto \exp\left(-\frac{(T - M(x))^2}{2\sigma_{\text{temp}}^2}\right) \tag{3.23}$$

The likelihood probability was more complex to implement, as this required utilisation of the function created from the finite element model as discussed in section 3.2. This function will output the probability of the modelled values $M(x)$ given the measured temperature values (T). As can be seen in Equation 3.23, the $M(x)$ vector is written as a function. The function indicated here takes the new randomised x vector and then runs the model to provide a new temperature distribution over time at various nodes. The output of the finite element model was reduced such that only the nodes at the same depths as the thermocouples are provided to the likelihood function. For the likelihood function, the σ_{temp} value was assumed to be 15°C.

3.3.3 Markov Chain Monte-Carlo integration

The two main parts of the MCMC integration (as mentioned in section 2) are: how a value is deemed acceptable (Monte Carlo), and how the next random sample is selected after a previous sample is accepted (Markov Chain). As explained in Section 2.4.1 a step size is a crucial part of determining the next random sample. For this project, a step size of 0.05 W/m·K was chosen.

Compared to the example shown in Figure 2.1, the actual problem is quite more complex. For this project, a log-normal distribution was chosen instead of a uniform distribution. If every coordinate direction in the aforementioned simple example is seen as a single entry in the x vector, then the example has three κ-values. The problem also technically becomes nine-dimensional, as there are nine separate independent κ-values that must be chained. An important part of the `takesteps.m` (Appendix A) was to ensure the steps are only taken in a logical direction and that they would not move into an illogical or improbable range. Negatives were specifically forbidden, as a negative thermal diffusivity would mean that heat flows in a direction opposite to the thermal gradient. Provision was made in the function to prevent the algorithm to run away from what we know to be the general area.

The acceptance criterion of the new x-vector is based on the proximity of the new posterior value to the reference posterior value and the previous posterior. A potential problem encountered with the standard acceptance probability method (shown in Section 2.4.2) is that the magnitudes of the values were not considered. This problem lead to many false positives, as incorrect values were accepted even though they were not acceptable. The acceptance criterion was based on Equation 3.24. The i value was chosen to be 50 arbitrarily, initially $i = 1$ but as can be seen in Figure 3.3.3 this acceptance was still to lenient.

$$\exp\left(\frac{\pi^*(x_{n+1}) - \pi^*(x_n)}{\pi^*(x_{\text{ref}})i}\right) \text{with} \quad i = 0.02 \tag{3.24}$$

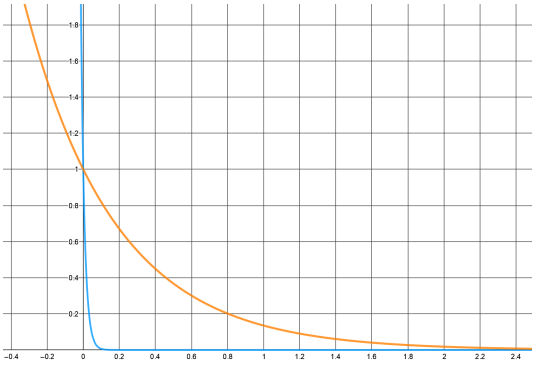


Figure 3.7: Graph explaining the difference in acceptance rates (Generated at https://www.geogebra.org/graphing/g7kyzwce)

To allow for sufficient number of samples the program was run until 20 000 samples were generated. The program was run four times with 5000 iterations each time. Due to computational limitations more iterations could not be run. Due to burn-in, as explained in Section 2.4.1, the first 100 samples of each run was removed from analysis. This resulted in 19 600 acceptable and useful samples that were analysed.

3.4 MAP

The maximum a posteriori of the posteriori function was also determined to allow for further comparison of results. All the values and constants except for the κ-values were kept the same as in the MCMC integration to allow the results to be compared on the same basis. The optimisation to determine the maximum value of the posteriori function was initially done utilising the built-in Matlab optimisation algorithm `femsearchmin.m`. Since the built-in optimisation intends to minimise the function the probability was multiplied with -1 such that the minimum found is truly the maximum. After initial analysis, the maximum was found at negative κ-values, as previously stated that is illogical and impractical. Further research on various adaptations of the built-in function lead to finding the `femsearchminbnd.m` adaptation by D'Errico (2021). This was used to limit the search area to only positive κ-values. Further, the algorithm was set to only iterate a thousand times and stop if the points of the simplex are within 0.009 of each other. The start point for the algorithm was chosen to be the κ-values from the EURO code.

A problem that arose was that the function could get stuck at a local maximum. This can be seen in Figure 4.1 on the histogram for 140°C

Chapter 4

Results

The results of the MCMC analysis as well as the MAP optimization is visualised in the section below.

4.1 Resulting κ-values

The thermal conductivity (κ-values) at key temperatures integral to this project, the resulting values are shown in Table 4.1. As can be seen in Figure 4.1 there was quite a drastic difference in the thermal diffusivity at 1200 °C and between 0°C and 200°C. The large difference at 1200°C is due to how the model is created, at that temperature the majority of the heat is transferred through thermal radiation. The model did not model thermal radiation separately but instead incorporated the radiation into a equivalent conduction. Between 0°C and 200°C the modelling of the evaporation was moderately successful as a spike can be seen at 100 °C.

For each temperature there will be a different distribution of samples. In

°C	EURO	MAP	MCMC
0	0.12	0.062	0.12
60	0.12	0.189	0.278
100	0.12	0.204	0.247
140	0.12	0.096	0.242
200	0.15	0.2355	0.243
350	0.07	0.117	0.096
500	0.09	0.039	0.120
800	0.35	0.265	0.234
1200	1.5	7.943	7.467

Table 4.1: Posterior thermal conductivity in W/m·K

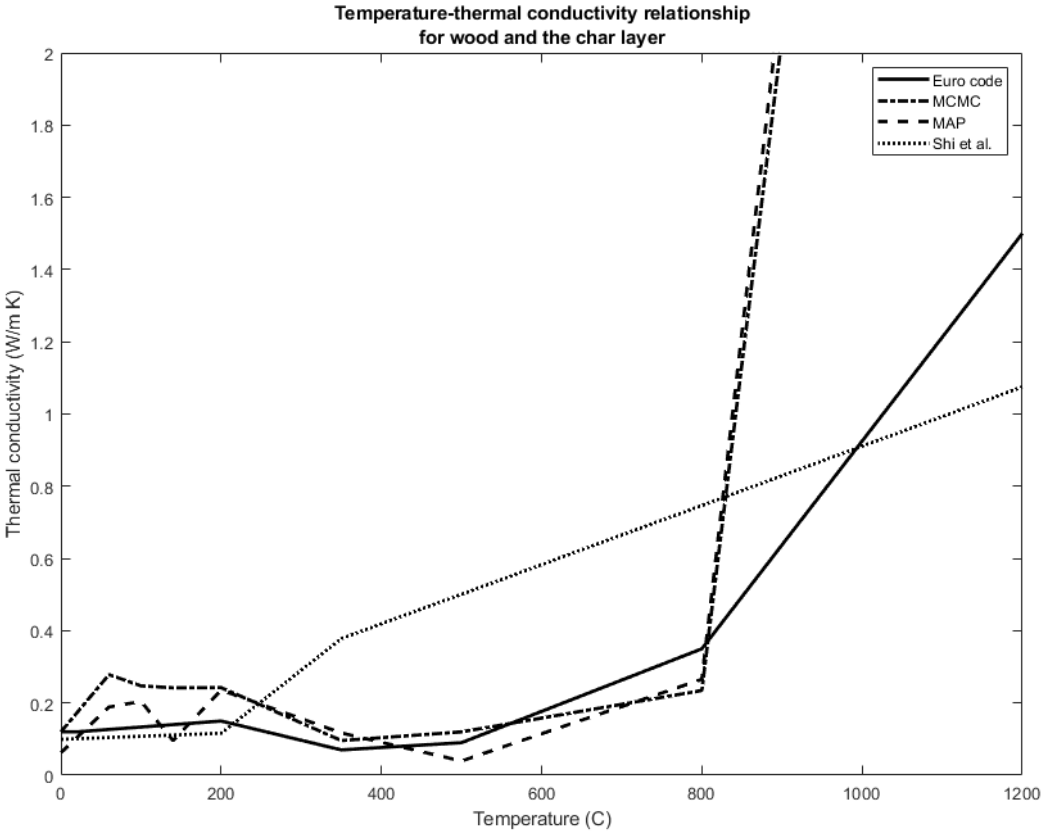


Figure 4.1: Resulting κ values compared to Euro-code standard values

Figure 4.1 The distribution of all the samples were plotted in a histogram with a kernel density estimation overlay. The resulting κ-values from the MCMC algorithm is plotted in red and the Maximum a posteriori is plotted in blue. From the blue lines it is clear that the Nelder-Mead optimization was not run a sufficient amount of times as many of the values are not at the maximum but on the way to the maximum. At 140°C it can also be seen that the algorithm got stuck at a local maximum.

4.1.1 Evaluation of error

Despite the uncertainties connected to the problem, a quantifiable measure of error is still required to allow unbiased assessment of the success of the algorithm and results. The obtained κ-values will be used to rerun the model and obtain the new modelled temperature for each depth. For each depth the error will be quantified using the norm of the relative error at the seven different depths, noted as i. The relative error will be calculated according to Equation 4.1. The new modelled output errors are compared to the EURO-code κ-

value output errors to assess if the methods used produced a more accurate representation of the data. The error at different depths are summarised in Table 4.1.1 and the lowest error at that depth is highlighted in green. At a depth of 0mm none of the errors are highlighted as this temperature is independent of the κ-values. This error measurement can be used to asses the error in the model or assumptions made to obtain the model. It is important to note that this error does not only quantify the error in the thermal conductivity but any errors due to model assumptions are also encompassed.

$$\epsilon(i) = \frac{\|\hat{x}(i) - x(i)\|_2}{\|x(i)\|_2} \qquad i = 1, ..., 7 \tag{4.1}$$

Depth(mm)	EURO	MAP	MCMC
0	0.2886	0.2886	0.2886
16.5	0.2166	0.1943	0.2050
33	0.3008	0.1905	0.2001
49.5	0.5693	0.3482	0.3227
66	0.5467	0.3186	0.2673
82.5	0.5012	0.3394	0.3583
99	0.2192	0.2190	0.2185

Table 4.2: Summary of error between models and data

4.1.2 Thermal diffusivity

The main intent of this project was to determine the thermal diffusivity of SA-Pine. In Table 4.1.2 the resulting thermal diffusivity is shown. The thermal diffusivity was calculated using the information in Table 2.1 and Equation 2.7.

Temp °C	EURO	MAP	MCMC
0	$1.64 \cdot 10^{-4}$	$8.46 \cdot 10^{-5}$	$1.64 \cdot 10^{-4}$
60	$1.64 \cdot 10^{-4}$	$2.58 \cdot 10^{-4}$	$3.79 \cdot 10^{-4}$
100	$1.64 \cdot 10^{-4}$	$2.78 \cdot 10^{-4}$	$3.37 \cdot 10^{-4}$
140	$1.64 \cdot 10^{-4}$	$1.31 \cdot 10^{-4}$	$3.30 \cdot 10^{-4}$
200	$2.05 \cdot 10^{-4}$	$3.21 \cdot 10^{-4}$	$3.32 \cdot 10^{-4}$
350	$9.55 \cdot 10^{-5}$	$1.60 \cdot 10^{-4}$	$1.31 \cdot 10^{-4}$
500	$1.23 \cdot 10^{-4}$	$5.32 \cdot 10^{-5}$	$1.64 \cdot 10^{-4}$
800	$4.78 \cdot 10^{-4}$	$3.62 \cdot 10^{-4}$	$3.19 \cdot 10^{-4}$
1200	$2.05 \cdot 10^{-3}$	$1.08 \cdot 10^{-2}$	$1.02 \cdot 10^{-2}$

Table 4.3: Resulting thermal diffusivity(α)in m/s^2

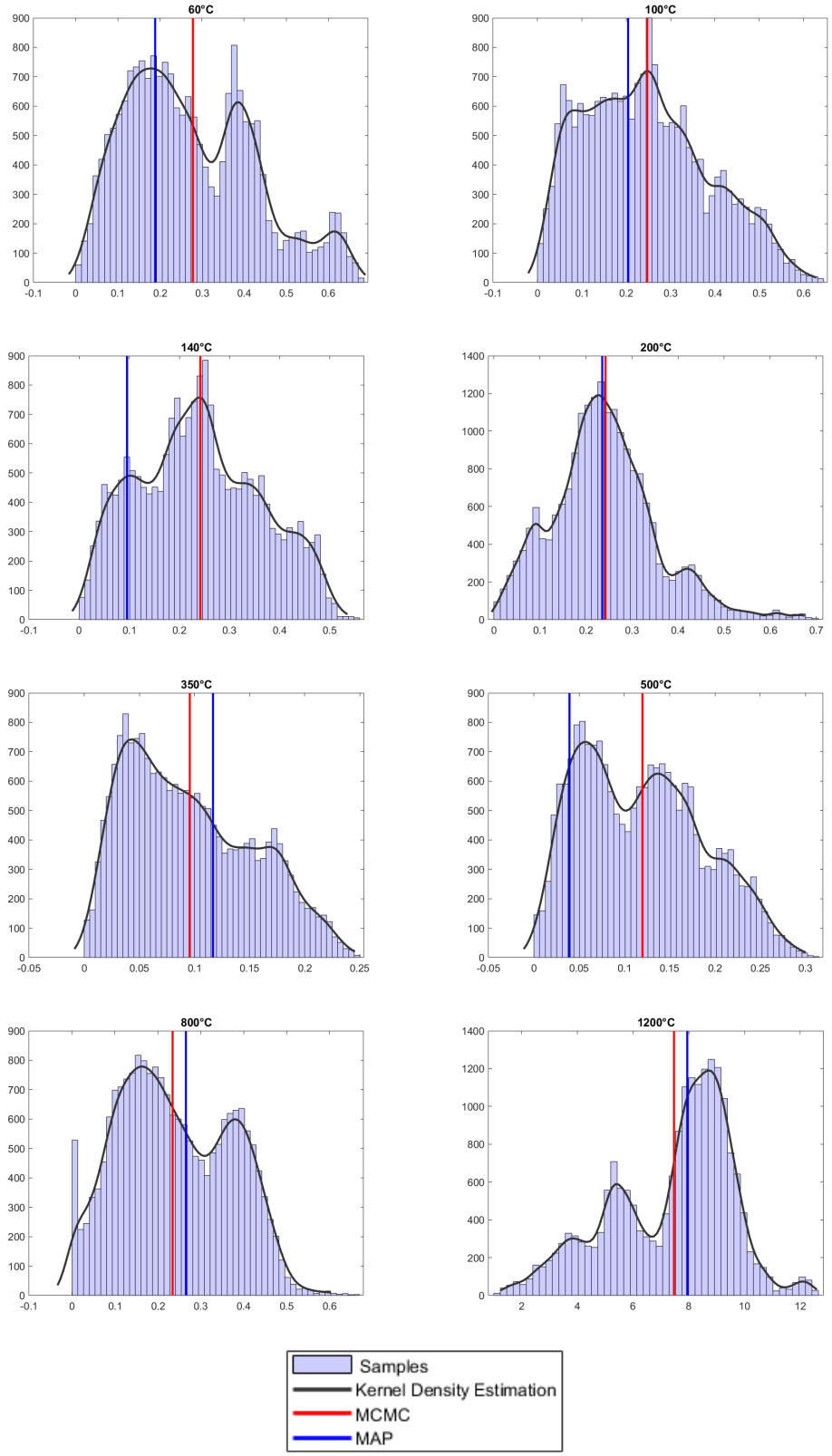


Figure 4.2: Distribution of samples at different temperatures with MCMC and MAP results indicated

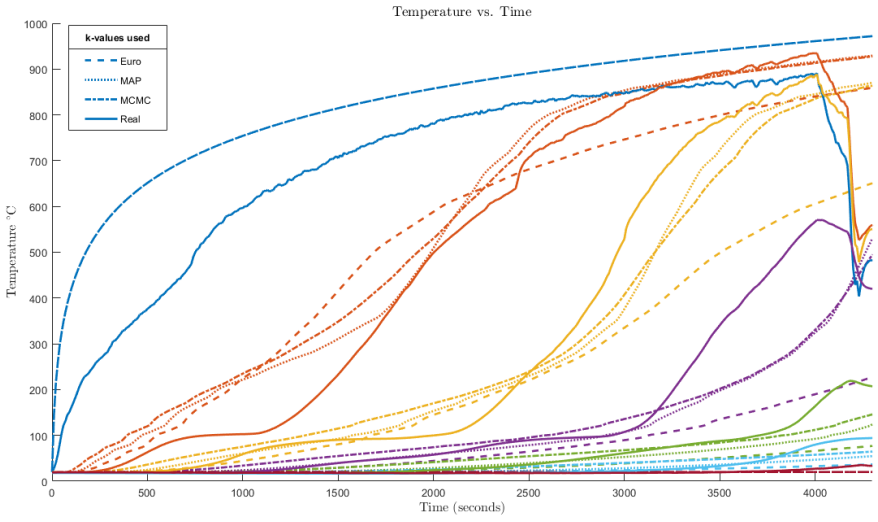


Figure 4.3: Graph of measured data compared to model output

Chapter 5

Discussion

In Figure 4.1.2 the difference in output from running the model with the new κ-values, Euro code κ and the measured data can be seen. The improved accuracy is clear from in Table 4.1.1 as the error for the model using MCMC values is consistently smaller than the error incurred by the same model using the Euro-code thermal conductivity (κ) values. There are instances where the error from the model using the MAP κ-values is smaller than the model using the MCMC κ-values. This should not discredit the usage and value of the Markov Chain Monte-Carlo algorithm as the increased accuracy from the MAP values is only marginal. It is also important to remember that the MAP optimization did not truly find the maximum at every point and that due to insufficient iterations the results from the MAP optimization was at some temperatures closer to the results from the MCMC algorithm than to the true maximum.

The κ-values obtained from the MAP and MCMC analysis give a more accurate model output than those from the Euro-code. The increased accuracy that the new κ-values provide, serves as a proof of concept that MCMC analysis can be used to determine more accurate fire ratings and specifically fire resistance.

Fire resistance is measured in minutes, this indicates the amount of time from the start of the fire until a specific condition is met. The main conditions taken into account are accounted for by the REI marker system (CETRIS, 2021). Where 'R' indicates the load-bearing capacity of the structural element, the associated time indicates when the element can no longer carry the design load. 'E' refers to the integrity of the element and the time indicates how long after ignition the fire penetrates the element a 'I', indicates insulation ability, a limit is set to the temperature of the non-heated side. Time corresponding with that rating refers to when that temperature is exceeded on the non-heated side.

With further development of analytical determination of thermal conductivity and applied finite element models, the data obtained from fire test could

be used in a model to determine the REI markers of different size elements.

The accuracy of the Markov Chain Monte-Carlo analysis can be increased by modelling the standard deviation, of both the temperatures and the κ values, as random values and solving for them as well. This would make the analysis more heuristic and decrease the dependence of the results on assumptions made by the researcher. Additionally the program could've been run for a longer time to ensure more samples. and thereby a more accurate approximation of the population mean. Unfortunately the main limitation in this project was computational time.

The MAP analysis could be fine-tuned yet that would be pointless as it is already known that the probability of the thermal conductivity is not symmetrically distributed and that the mean is a more accurate description of the data than the Maximum a posteriori.

In retrospect there should have been more cleaning of the measured data before analysis was done. Inaccuracies arose due to the first 90 sec of the measurements being taken before the furnace was turned on and the thermocouples still measuring after the fire died down. These measurement discrepancies between model and measurement should have been better taken into account and excluded form modelling.

This research can be expanded and applied to the measured data available for Eucalyptus. Applying the same algorithm to different data sets and obtaining accurate modelling from both data sets would be confirmation that the algorithm is accurate enough to be further explored for usage in practice.

Chapter 6

Summary and Conclusion

The usage of Markov Chain Monte-Carlo integration to solve for the thermal diffusivity of cross-laminated SA Pine was successful. The adaptation of the finite element model derived from the heat conduction and diffusion equation enabled the creation of a posterior distribution. The posterior distribution modelled after Bayes theory of inverse problems could be traversed and explored using the Markov Chain Monte-Carlo and the Maximum a Posteriori could be found using Nelder-Mead optimization. The resulting κ and α values produced a more accurate model of temperature over time. There is a lot of potential for further optimization and fine-tuning of these algorithms and models. With further development this concept could lead to simplified methods of calculating the fire rating of specifically SA-Pine as well as other timber samples. More accurate and accessible fire ratings will help pave the way to improved fire safety.

Appendix A

Relevant code segments

Below are the relevant code segments taken from the Matlab code that was used to generate the final samples used for analysis.

Prior function

Determines the probability of x assuming a (NORMAL) probability over the prior values.

```
function q_val = prior_pdf(x_values,sigmaMU)

 global mu_values
 q_val = -0.5*((x_values - mu_values)*(x_values -
    mu_values)')/(sigmaMU^2);

end
```

Likelihood Function

```
function likelihood_pi = likelihood_func(kinput,sigmaT)

% Global variables: physical constants
global_const;
% Global variables: material properties
global_prop
% Global variables: time integration values
global_time;
% Global variables: mesh geometry
global_mesh;
% Global variables of measured temperatures
```

```
global_measuredtemp;

udepths = model_kinput(kinput);
depths_measured = [depth1;depth2;depth3;depth4;depth5;depth6;depth7];
%Where depths measure is the actual measured temperature data from
    the experiment and udepths is the temperature at the same points
    generated by the model using the new k-values.

tempmat = depths_measured - udepths;
likelihood_pi = tempmat(:)'*tempmat(:)/(-2*sigmaT^2);

end
```

Adapted model function

Model adapted to a function as explained in Section 3.2.3. Based on source code from Prof. N de Koker.

```
function udepths = model_kinput(kinput)

global_time;
global_mesh;
global_const;
global_prop;

kpine = kinput;

% Print current number of elements and current sample k-values
kpine(2,:)
fprintf("%d \n",nels);

% Calculate boundary conditions
[isofc, isofcdot] = isofirecurve(tvec);
%isofc(tvec>1000) = 20; % check cooling effect
%isofcdot(tvec>1000) = 0;
atmtemp = tvec*0 + 20; % Celsius
atmtdot = tvec*0;

% Set up initial conditions
unode(:,1) = xnp*0 + atmtemp(1);
unode(1,1) = isofc(1);
vnode(:,1) = xnp*0 + atmtdot(1);
vnode(1,1) = isofcdot(1);%time derivatives
```

```
% Iterate: calculate temperature profile
for i = 2:length(tvec)
    % Set up Dirichlet
    ddir = zeros(nnp,1);
    vdir = zeros(nnp,1);
    ddir(1) = isofc(i);
    vdir(1) = isofcdot(i);
    ddir(nnp) = atmtemp(i);
    vdir(nnp) = atmtdot(i);

    % Assemble global matrices
    Kmat0 = glomatK(unode(:,i-1), ddir);
    Mmat0 = glomatM(unode(:,i-1), ddir);
    Fvec0 = glovecF(unode(:,i-1), ddir);

    % Account for Dirichlet
    Fdir = Kmat0*ddir + Mmat0*vdir;
    Fvec0 = Fvec0 - Fdir;
    Kmat = Kmat0(2:end-1, 2:end-1);
    Mmat = Mmat0(2:end-1, 2:end-1);
    Fvec = Fvec0(2:end-1);

    % Solve for d and ddot in this step
    [unode(2:end-1,i), vnode(2:end-1,i)] = dvnext(
        unode(2:end-1,i-1), vnode(2:end-1,i-1), Kmat, Mmat, Fvec );
    unode(:,i) = unode(:,i) + ddir;
    vnode(:,i) = vnode(:,i) + vdir;

    clear Kmat0 Kmat Fvec0 Fvec Mmat0 Mmat;

end
% Isolate nodes at depth of thermocouples
count = [2 7 12 17 22 27 32];

udepth1 = unode(count(1),2:end);
udepth2 = unode(count(2),2:end);
udepth3 = unode(count(3),2:end);
udepth4 = unode(count(4),2:end);
udepth5 = unode(count(5),2:end);
udepth6 = unode(count(6),2:end);
udepth7 = unode(count(7),2:end);

udepths = [udepth1;udepth2;udepth3;udepth4;udepth5;udepth6;udepth7];

end
```

Function to take next step

This function takes the current x-vector and generates a new probable x-vector.

```
function xvalue2 = takexsteps(xvalue1)
   global temps mu_values stepsize sigmastepMU sigmastepT

   locsigma = stepsize*mu_values;
   locxvalue = xvalue1;

   lnMu = log(locxvalue.^2 ./ sqrt(locsigma.^2+locxvalue.^2));
   lnSigma = sqrt(log(locsigma.^2./xvalue1.^2 + 1));

   xvalue2 = max(0, lognrnd(lnMu, lnSigma));
   xvalue2(1) = xvalue1(1);

   xvalue2(xvalue2<locsigma/20) =
       (mu_values(xvalue2<locsigma/20)+xvalue2(xvalue2<locsigma/20))/2;

end
```

MAP determination

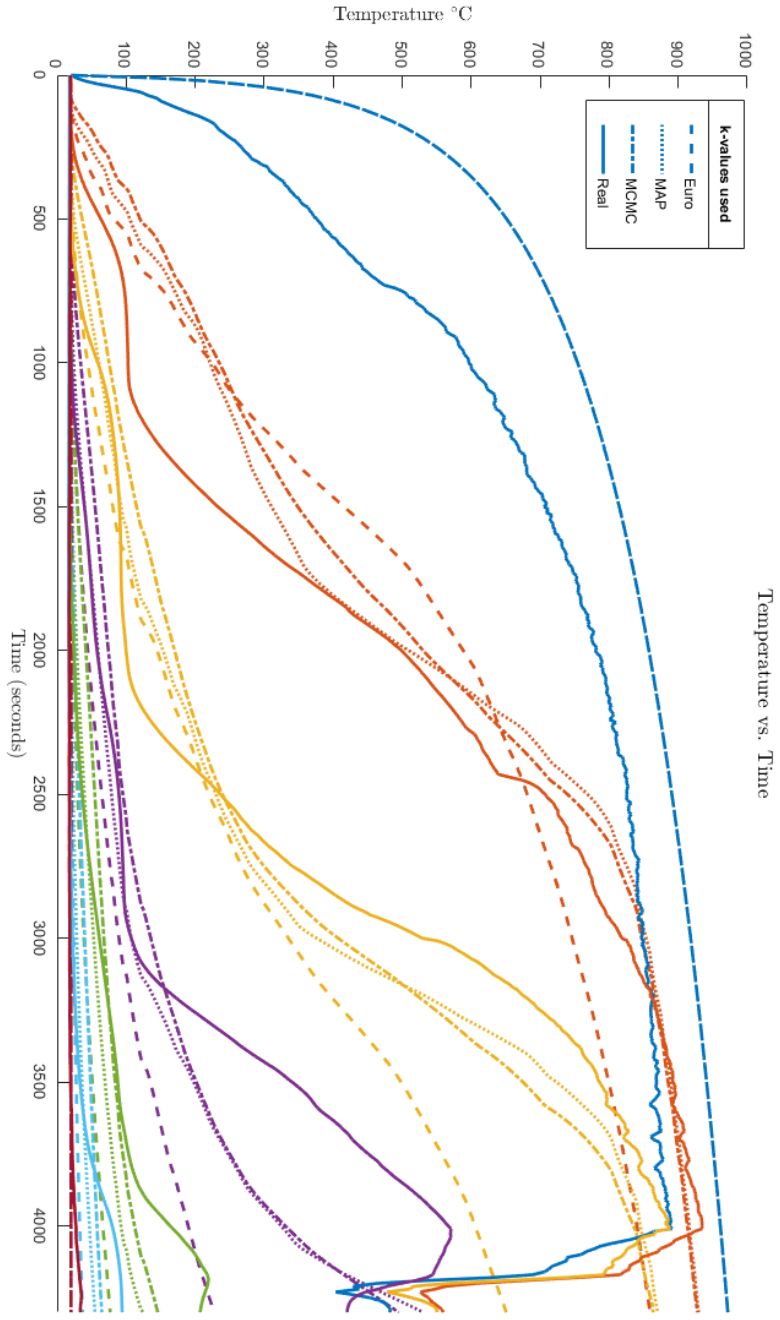
```
global temps stepsize mu_values
temps = [0,60,100,140, 200,350,500,800,1200];
mu_values = [0.12, 0.12, 0.12, 0.12, 0.15, 0.07, 0.09, 0.35, 1.5];
x_values = mu_values;
stepsize = 0.05;
sigmaT1 = 15;
sigmaMU1 = 0.13;
instantiate_all();

fminopts = optimset('fminsearch');
fminopts.TolX = 0.0009;
fminopts.MaxFunEvals = 1000;
fminopts.MaxIter = 1000;
lowBound= [0.000, 0.000, 0.000, 0.000, 0.000, 0.0000, 0.0000, 0.00,
    0.1];
upBound = [0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 0.8, 20];

fun = @(x_values) -1*posteriori_func(x_values,sigmaT1,sigmaMU1);
[mu_map,fval] = fminsearchbnd(fun,
    x_values,lowBound,upBound,fminopts);
```

Appendix B

Detailed results graph


Temperature °C
Time (seconds)
k-values used
Euro
MAP
MCMC
Real
Temperature vs. Time

Appendix C

Derivation of Implemented FEM Model

$$q_{,x} - f = 0 \ ...(1) \qquad\qquad q = -\kappa u_{,x} \ ...(2) \tag{C.1}$$

Integrating Equation C.1 (1) over the length of the element and introducing a weighting function $w(x)$ we obtain C.2. Since the derivative of $w(0)$ is known and $q_{,x}$ is unknown. The first term in C.2 is integrated by parts. After the integration by parts and substituting q with C.1 (2), Equation C.3 is created.

$$\int_{x=0}^{L} w q_{,x} dx - \int_{x=0}^{L} w f dx = 0 \tag{C.2}$$

$$\int_{x=0}^{L} w k u_{,x} dx + \int_{x=0}^{L} w f dx - w q|_0^L = 0 \tag{C.3}$$

In Equation C.3 the u and w need to be defined. Assuming $u \approx u^h$ and $w \approx w^h$ and using the basis function (N^A), Equation C.4 is obtained.

$$\begin{aligned} u_e^h &= \sum_B N^B d^B \quad ; \quad w_e^h = \sum_A N^A c^A \\ u_{e,x}^h &= \sum_B N_{,x}^B d^B \quad ; \quad w_{e,x}^h = \sum_A N_{,x}^A c^A \end{aligned} \tag{C.4}$$

Substituting the u and w functions back, we obtain the Galerkin weak form shown in Equation C.5.

$$\begin{aligned} \sum_e \int_{\Omega_e} w_{e,x}^h k u_{e,x}^h dx + \sum_e \int_{\Omega_e} w_e^h f dx - w(L) q_L + w(0) q_0 &= 0 \\ \int_{\Omega_e} \sum_A \sum_B N_{,x}^A c^A k N_{,x}^B d^B dx + \int_{\Omega_e} \sum_A N^A c^A f dx - \sum_{A \in A_N} c^A q^A &= 0 \end{aligned} \tag{C.5}$$

$$\mathbf{c}_e^T \mathbf{K}_e^{AB} \mathbf{d}_e + \mathbf{c}_e^T \mathbf{F}_e^f - \mathbf{c}_e^T \mathbf{F}_e^q \tag{C.6}$$

where

$$\begin{aligned}
K_e^{AB} &= \int_{\Omega_e} N_{,x}^A N_{,x}^B k dx \\
F_e^{Af} &= \int_{\Omega_e} N_{,x}^A f dx \\
F_e^{Aq} &= \begin{cases} q(x_N) & \text{for } x \in \Gamma_N \\ 0 & \text{for other} \end{cases}
\end{aligned} \tag{C.7}$$

$$\begin{aligned}
q_{\text{adv}} &= \nu \rho c_p \Delta T \quad = \quad h \Delta T \\
\nu &= \text{velocity m/s} \\
\rho &= \text{density of air} \\
c_\rho &= \text{heat capacity of air}
\end{aligned}$$

$$\begin{aligned}
q_{\text{rad}} &= \varepsilon \sigma \phi \left(T_f^4 - T_S^4 \right) \\
\varepsilon &= \text{emissivity} \\
\sigma &= \text{Stefan-Boltzmann} \\
&\quad 5.670e^{-}8[W/(m^2 K^4)] \\
\phi &= \text{view factor; 1} \quad \text{here}
\end{aligned} \tag{C.8}$$

Given that T_S and T_F are known a new equivalent heat flux value can be calculated as in Equation 3.10.

$$q_{\text{con}}^{\text{equiv}} = \kappa^{\text{equiv}} \frac{\Delta T}{\Delta L} = q_{\text{rad}} + q_{\text{adv}} \tag{C.9}$$

Due to the clear relationship between heat flux and thermal diffusivity the equivalent diffusivity(κ^{equiv}) can be calculated as shown below in Equation 3.11.

$$\begin{aligned}
\kappa^{\text{equiv}} &= \frac{[q_{\text{rad}} + q_{\text{adv}}]}{\Delta T} \\
&= \frac{\epsilon \sigma \phi \Delta \left(T^4 \right)}{\Delta T / \Delta L} + h \Delta L \\
&= \frac{\epsilon \sigma \Delta \left(T^4 \right)}{\Delta T} + h
\end{aligned} \tag{C.10}$$

Diffusion Equation:

$$q_{,x} - f = \frac{\partial Q}{\partial t} = c_\rho \frac{\partial u}{\partial t} \quad ...(1) \qquad q = -\kappa \frac{\partial u}{\partial x} \quad ...(2) \tag{C.11}$$

Substituting Equation C.11(2) into C.11 and taking the derivative as indicated $(q_{,x})$ gives Equation C.12. As previously disccussed heat conduction (α) is heat diffusion (κ) divided by specific heat (c_p).

$$\begin{gathered}\therefore -\kappa\frac{\partial^2 u}{\partial x^2} - f = c_\rho\frac{\partial u}{\partial t} \rightarrow f = 0: \\ \frac{\partial^2 u}{\partial x^2} = -\frac{c_\rho}{\kappa}\frac{\partial u}{\partial t} \\ \text{or} \\ \frac{\partial u}{\partial t} = -\alpha\frac{\partial^2 u}{\partial x^2}\end{gathered} \tag{C.12}$$

Let $c_p = \lambda$

$$\therefore -\kappa u_{,xx} - \lambda u_{,t} = f \tag{C.13}$$

Then:

$$\int_0^L w q_{,x}\mathrm{d}x - \int_0^L w\lambda u_{,t}\mathrm{d}x - \int_0^L w f\mathrm{d}x = 0 \tag{C.14}$$

Similar to what was done in C.4 a special approximation is made to obtain Equation C.15.

$$u \approx u^h \rightarrow u_e^h = \sum_A N^A d^A \quad ; \quad w_e^h = \sum_A N^A L^A \tag{C.15}$$

$$\rightarrow \sum_e \int_{\Omega^e} w_{e,x}^h \kappa u_{e,x}^h \mathrm{d}x_e + \sum_e \int_{\Omega^e} w_e^h \lambda u_{e,t}^h \mathrm{d}x_e + \sum_e \int_{\Omega^e} w_e^h f \mathrm{d}x_e - \sum_{e\in\epsilon_A} w q_N = 0 \tag{C.16}$$

$$*_1 : \int_{\Omega^e} \sum_A \sum_B N_{,x}^A c^A \kappa N_{,x}^B d^B \mathrm{d}x = \sum_A \sum_B \left[\int N_{,x}^A N_{,x}^B \kappa \mathrm{d}x\right] d^B = \mathbf{c}_e^T \boldsymbol{\kappa}_e \mathbf{d}_e$$

$$*_2 : \int_{\Omega_e} \sum_A \sum_B N^A c^A \lambda N^B \dot{d}^B \mathrm{d}x = \sum\sum c^A \left[\int N^A N^B \lambda \mathrm{d}x\right] d^B = \mathbf{c}_e^T \mathbf{M}_e \dot{\mathbf{d}}_e$$

$$*_3 : \int_{\Omega^e} \sum_A N^A c^A f \mathrm{d}x = \sum c \int_B^A N f \mathrm{d}x = \mathbf{c}_e^T \mathbf{F}_e^B$$

$$*_4 : \sum_{A\in\mathcal{A}} \kappa^A q_N^A = \mathbf{c}_e^T \mathbf{F}_e^q$$

From all the above equations the below matrix formulation could be assembled:

$$\mathbf{c}^T\mathbf{K}\mathbf{d} + \mathbf{c}^T\mathbf{M}\dot{\mathbf{d}} = \mathbf{c}^T\mathbf{F} \tag{C.17}$$

Set $\mathbf{d}_0$ equal to the known conditions at time 0.

And $\mathbf{v}_0 = \mathbf{0}$

Also set α as a parameter between 0 and 1. Δt is the time step.

Following the v-form Time integration in Hughes Chapter 8 , page 460, define $\mathbf{d}_{n+1}$ as in Equation C.18.

$$\mathbf{d}_{n+1} = \mathbf{d}_n + (1-\alpha)\Delta\mathbf{v}_n + \alpha\Delta t\mathbf{v}_{n+1} \tag{C.18}$$

$$(\mathbf{M} + \alpha\Delta t\boldsymbol{\kappa})\mathbf{v}_{n+1} = \mathbf{F}_{n+1} - \boldsymbol{\kappa}\tilde{\mathbf{d}}_{n+1} \tag{C.19}$$

If $\mathbf{M}$ and $\boldsymbol{\kappa}$ independent of time and temperature,it is sufficient to say:

$$\mathbf{v}_{n+1} = (\mathbf{M} + \alpha\Delta t\boldsymbol{\kappa})^{-1}(\mathbf{F}_{n+1} - \boldsymbol{\kappa}\tilde{\mathbf{d}}_{n+1}) \tag{C.20}$$

If κ and λ is independent of position:

$$\begin{aligned}
\kappa_e^{AB} &= \int_0^\ell N_{,x}^A N_{,x}^B \kappa \mathrm{d}x \\
&= \kappa \int_0^\ell N_{,x}^A N_{,x}^B \mathrm{d}x \\
&= \kappa \int_{-1}^{+1} N_{,\xi}^A N_{,\xi}^B \xi_{,x}^2 \left|J\right| d\xi \\
&= \kappa \frac{2}{\ell} \int_{-1}^{+1} N_{,\xi}^A N_{,\xi}^B \mathrm{d}\xi \\
&= \pm\frac{\kappa}{\ell} = \begin{cases} +\frac{1}{2} & \text{if } A = B \\ -\frac{1}{2} & \text{if } A \neq B \end{cases}
\end{aligned} \tag{C.21}$$

$$\therefore \boldsymbol{\kappa}_e = \frac{\kappa}{\ell}\begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \tag{C.22}$$

$$M_e^{AB} = \int_0^\ell N^A N^B \lambda \mathrm{d}x = \lambda \int_0^\ell N^A N^B \mathrm{d}x = \frac{\lambda\ell}{2}\int_{-1}^{+1} N^A N^B \mathrm{d}x \tag{C.23}$$

$$N^1N^1 = \frac{1}{4}(1-\xi)^2; N^2N^2 = \frac{1}{4}(1+\xi)^2; N^1N^2 = \frac{1}{4}(1-\xi^2) \tag{C.24}$$

$$N^1 = \frac{1-\xi}{2}; N^2 = \frac{1+\xi}{2} \tag{C.25}$$

Then

$$\begin{aligned}\int N^1N^1 &= \frac{1}{4}\left[\xi+\frac{1}{3}\xi^3\right]_{-1}^{+1} = \frac{2}{3}\\ \int N^2N^2 &= \frac{1}{4}\left[\xi+\frac{1}{3}\xi^3\right]_{-1}^{+1} = \frac{2}{3}\\ \int N^1N^2 &= \frac{1}{4}\left[\xi-\frac{1}{3}\xi^3\right]_{-1}^{+1} = \frac{1}{3}\end{aligned} \tag{C.26}$$

$$\therefore \mathbf{M}_e = \frac{\lambda\ell}{6}\begin{bmatrix}2 & 1\\ 1 & 2\end{bmatrix} \tag{C.27}$$

$$\mathbf{F}_e^{fA} = \int_0^\ell Nf\mathrm{d}x = f\frac{\ell}{2}\int_{-1}^{+1} N\mathrm{d}x = f\frac{\ell}{2} \therefore \mathbf{F}_e^f = \frac{f\ell}{2}\begin{Bmatrix}1\\1\end{Bmatrix} \tag{C.28}$$

More generally matrix K_e and M_e are:

$$\begin{aligned}K_e^{AB} &= \int_0^\ell N_{,x}^A N_{,x}^B \kappa \mathrm{d}x \text{ with } \kappa = N^1\kappa^1 + N^2\kappa^2\\ &= \left(\frac{\ell}{2}\right)\left(\frac{\pm 1}{4}\right)\left(\frac{4}{\ell^e}\right)\int_{-1}^{+1}\left[\frac{1-\xi}{2}\kappa^{(1)} + \frac{1+\xi}{2}\kappa^{(2)}\right]\mathrm{d}\xi\end{aligned} \tag{C.29}$$

and

$$M_e^{AB} = \int_0^\ell N^A N^B \lambda \mathrm{d}x \quad \text{with} \quad \lambda = N^1\kappa^{(1)} + N^2\kappa^{(2)} \tag{C.30}$$

If $A = 1$ and $B = 1$ then M_e is calculated as shown below in Equation C.31.

$$\begin{aligned}M_e^{11} &= \left(\frac{\ell}{2}\right)\int_{-1}^{+1}\frac{1}{4}(1-\xi)^2\left[\frac{1-\xi}{2}\cdot\lambda^{(1)} + \frac{1+\xi}{2}\cdot\lambda^{(2)}\right]\mathrm{d}\xi\\ &= \frac{\ell}{16}\int_{-1}^{+1}\left[(1-\xi)^2(1-\xi)\lambda^{(1)} + (1-\xi)^2(1+\xi)\lambda^{(2)}\right]\mathrm{d}\xi\\ &= \frac{\ell}{16}\left[\lambda^{(1)}\int_{-1}^{+1}(1-\xi)^2(1-\xi)\mathrm{d}\xi + \lambda^{(2)}\int_{-1}^{+1}(1-\xi)^2(1+\xi)\mathrm{d}\xi\right]\\ &= \frac{\ell}{16}\left[\lambda^{(1)}\cdot 4 + \lambda^{(2)}\cdot\frac{4}{3}\right]\\ &= \frac{\ell}{4}\left[\lambda^{(1)} + \frac{\lambda^{(2)}}{3}\right]\end{aligned} \tag{C.31}$$

When $A = 2$ and $B = 2$:

$$\begin{aligned} M_e^{22} &= \frac{\ell}{16}\int_{-1}^{+1}\left[(1+\xi)^2(1-\xi)\lambda^{(1)} + (1+\xi)^2(1+\xi)\lambda^{(2)}\right]\mathrm{d}\xi \\ &= \frac{\ell}{4}\left[\frac{\lambda^{(1)}}{3} + \lambda^{(2)}\right] \end{aligned} \tag{C.32}$$

When $A = 1$ and $B = 2$ **or** $A = 2$ and $B = 1$:

$$\begin{aligned} M_e^{12} &= \frac{\ell}{16}\int_{-1}^{+1}\left[(1-\xi^2)(1-\xi)\lambda^{(1)} + (1-\xi^2)(1+\xi)\lambda^{(2)}\right]\mathrm{d}\xi \\ &= \frac{\ell}{4}\left[\frac{\lambda^{(1)}}{3} + \frac{\lambda^{(2)}}{3}\right] \\ &= \frac{\ell}{12}\left[\lambda^{(1)} + \lambda^{(2)}\right] \end{aligned} \tag{C.33}$$

The matrix can then be assembled from the above Equations C.31,C.32 and **??**:

$$\begin{aligned} \mathbf{M}_e &= \frac{\ell}{4}\left(\lambda^{(1)}\begin{bmatrix}1 & \frac{1}{3}\\ \frac{1}{3} & \frac{1}{3}\end{bmatrix} + \lambda^{(2)}\begin{bmatrix}\frac{1}{3} & \frac{1}{3}\\ \frac{1}{3} & 1\end{bmatrix}\right) \\ &= \frac{\ell}{12}\left(\lambda^{(1)}\begin{bmatrix}3 & 1\\ 1 & 1\end{bmatrix} + \lambda^{(2)}\begin{bmatrix}1 & 1\\ 1 & 3\end{bmatrix}\right) \end{aligned} \tag{C.34}$$

The same can be done to assemble the K matrix as shown below.

$$\begin{aligned} K_e^{AB} &= \left(\frac{\pm 1}{2\ell}\right)\int_{-1}^{+1}\left[\frac{1-\xi}{2}\kappa^{(1)} + \frac{1+\xi}{2}\kappa^{(2)}\right]\mathrm{d}\xi \\ &= \left(\frac{\pm 1}{2\ell}\right)\left[\kappa^{(1)} + \kappa^{(2)}\right] \end{aligned} \tag{C.35}$$

$$\mathbf{K}_e = \frac{1}{2\ell}\left(\kappa^{(1)}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} + \kappa^{(2)}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\right) \tag{C.36}$$

There after the $\mathbf{F}_e$ matrix

$$\begin{aligned} F_e^A &= \int_0^\ell N^A f\,\mathrm{d}x \\ &= \frac{\ell}{2}\int_{-1}^{+1} N^A f\,\mathrm{d}\xi \end{aligned} \tag{C.37}$$

$$\begin{aligned} F_e^1 &= \frac{\ell}{2}\int_{-1}^{+1}\left(\frac{1-\xi}{2}\right)\cdot\left[\frac{1-\xi}{2}\cdot f^{(1)} + \frac{1+\xi}{2}\cdot f^{(2)}\right] \\ &= \frac{\ell}{8}\int_{-1}^{+1}\left[(1-\xi)^2\cdot f^{(1)} + \left(1-\xi^2\right)\cdot f^{(2)}\right]\mathrm{d}\xi \end{aligned} \tag{C.38}$$

$$F_e^2 = \frac{\ell}{8}\int_{-1}^{+1}\left[(1-\xi^2)\cdot f^{(1)} + (1+\xi)^2\cdot f^{(2)}\right]\mathrm{d}\xi \tag{C.39}$$

$$\begin{aligned}\therefore \mathbf{F}_e &= \frac{\ell}{8}\left(f^{(1)}\begin{Bmatrix}\frac{8}{3}\\ \frac{4}{3}\end{Bmatrix} + f^{(2)}\begin{Bmatrix}\frac{4}{3}\\ \frac{8}{3}\end{Bmatrix}\right) + \sum_{A\in\mathcal{A}_N} q^{(A)}\begin{Bmatrix}-\delta^{A1}\\ +\delta^{A2}\end{Bmatrix}\\ &= \frac{\ell}{6}\left(f^{(1)}\begin{Bmatrix}2\\ 1\end{Bmatrix} + f^{(2)}\begin{Bmatrix}1\\ 2\end{Bmatrix}\right) + \sum_{A\in\mathcal{A}_N} q^{(A)}\begin{Bmatrix}-\delta^{A1}\\ +\delta^{A2}\end{Bmatrix}\end{aligned} \tag{C.40}$$

After assembly but before Dirichlet boundaries are applied

$$\bar{\mathbf{c}}^T\bar{\mathbf{K}}\bar{\mathbf{d}} + \bar{\mathbf{c}}^T\bar{\mathbf{M}}\dot{\bar{\mathbf{d}}} = \bar{\mathbf{c}}^T\bar{\mathbf{F}} \tag{C.41}$$

When the Dirichlet boundaries below (Equation C.42) are applied to the matrices shown in Equation C.41 , they can be simplified to Equation C.43.

$$\begin{aligned} c_1 = 0 \quad &\text{and} \quad c^{np} = 0\\ d_1 = u_{\text{fire(t)}} \quad &\text{and} \quad d^{np} = u_{\text{air}}\end{aligned} \tag{C.42}$$

Then:

$$\begin{aligned}&\mathbf{c}^T\cdot\mathbf{K}\cdot\mathbf{d} + \mathbf{c}^T\cdot\mathbf{M}\cdot\dot{\mathbf{d}} = \mathbf{c}^T\cdot\mathbf{F}' - \mathbf{c}^T\\ &\left(\{K'\}d' + \{K^{np?}\}d^{np}\right) - \mathbf{c}^T\left(\{M'\}\dot{d'} + \{M^{np?}\}\dot{d}^{np}\right)\end{aligned} \tag{C.43}$$

And so:

$$\mathbf{K}\cdot\mathbf{d} + \mathbf{M}\cdot\dot{\mathbf{d}} = \mathbf{F}' - \mathbf{F}^{Ke} - \mathbf{F}^{Me} = \mathbf{F} \tag{C.44}$$

Solve as

$$\begin{aligned}\tilde{\mathbf{d}}_{n+1} &= \mathbf{d}_n + (1-\alpha)\Delta t\mathbf{v}_n\\ (\mathbf{M}+\alpha\Delta t\mathbf{K})\mathbf{v}_{n+1} &= \mathbf{F}_{n+1} - \mathbf{K}\tilde{\mathbf{d}}_{n+1}\\ \rightarrow \mathbf{v}_{n+1} &= (\mathbf{M}+\alpha\Delta t\mathbf{K})^{-1}(\mathbf{F}_{n+1} - \mathbf{K}\tilde{\mathbf{d}}_{n+1})\\ \rightarrow \mathbf{d}_{n+1} &= \tilde{\mathbf{d}}_{n+1} + \alpha\Delta t\mathbf{v}_{n+1}\\ \mathbf{v} &= \dot{\mathbf{d}}\end{aligned} \tag{C.45}$$

List of References

Brownlee, J. (2019 Nov [Online]). A gentle introduction to markov chain monte carlo for probability. Machine Learning Mastery, Blog.
Available at: https://machinelearningmastery.com/markov-chain-monte-carlo-for-probability/

CEN (2004). *BS EN 1995-1-2:2004. Eurocode 5: Design of timber structures - General - Structural fire design.* BSI, London.

CETRIS (2021). Application of CETRIS boards in fire protection pursuant to EN-standards. Information booklet. Obtained from www.cetris.cz.

Cheng, A. and Cheng, D. (2005 03). Heritage and early history of the boundary element method. *Engineering Analysis with Boundary Elements*, vol. 29, pp. 268–302.

Cook, J.D. (2016 Jan [Online]). MCMC burn-in. John D. Cook Consulting.
Available at: https://www.johndcook.com/blog/2016/01/25/mcmc-burn-in/comment-page-1/?unapproved=1094214&moderation-hash=912d585bd365341b032df1d4f1bddf3b#comment-1094214

Courant, R. (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, pp. 1–23. ISSN 1088-9485.
Available at: https://doi.org/10.1090/S0002-9904-1943-07818-4

D'Errico, J. (2021). fminsearchbnd, fminsearchcon. MATLAB Central File Exchange.
Available at: https://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon

Fish, J.J. (2007). *A first course in finite elements*. John Wiley, Chichester. ISBN 9780470035801.

Fourier, J. (1878). *The analytical theory of heat.* Cambridge University Press.

Frayssinhes, R., Girardon, S., Marcon, B., Denaud, L. and Collet, R. (2020). A simple method to determine the diffusivity of green wood. *BioRes*, vol. 3, no. 15, pp. 6539–6549.

Gilks, W.R., Richarson, S. and Spiegelhalter, D.J. (1996). *Markov chain Monte Carlo in practice*. London, 1st edn.

Gupta, K. and Meek, J. (1996). A brief history of the beginning of the finite element method. *International Journal for Numerical Methods in Engineering*, vol. 39, pp. 3761–3774.

Hughes, T. (1987). *The Finite Element Method: Linear Statistic and dynamic finite element analysis*. Englewood Cliffs, New Jersey.

ISO (1999). *ISO 834 Fire-resistance test - Elements of building construction*. International Organization for Standardization, Geneva.

Kaipio, J. and Somersalo, E. (2005). *Statistical and Computational Inverse Problems*. Verlag New York.

Lagarias, J., Reeds, J., Wright, M. and Wright, P. (1998 Dec). Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, vol. 9, pp. 112–147.

Meyn, S. and Tweedie, R. (1993 Jan). *Markov Chains and Stochastic Stability*, vol. 92. Springer Verlag.

Murphy, K.P. (2012). *Machine learning : A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts. ISBN 9780262018029.

Robert, C.P. and Casella, G. (2004). *Monte Carlo Statistical Methods*, vol. 42. 2nd edn. Springer New York. ISBN 9781441919397.

Salvadori, V. (2017 12). *The Development of a Tall Wood Building*. Master's thesis, Polytechnic University of Milan and TU Wien.

Shi, L. and Chew, M.Y.L. (2021 Aug). A review of thermal properties of timber and char at elevated temperatures. *Indoor and Built Environment*, pp. 1–16. `https://doi.org/10.1177/1420326X211035557`.
Available at: `https://doi.org/10.1177/1420326X211035557`

van der Westhuyzen, S., Walls, R. and de Koker, N. (2020). Fire tests of south african cross-laminated timber wall panels: Fire ratings, charring rates, and delamination. *Journal of the South African Institution of Civil Engineering*, vol. 62, no. 1, pp. 33–41. ISSN 23098775.

Wagner, P.-R., Marelli, S. and Sudret, B. (2021). Bayesian model inversion using stochastic spectral embedding. *Journal of Computational Physics*, vol. 436, p. 110141. ISSN 0021-9991.
Available at: `https://www.sciencedirect.com/science/article/pii/S0021999121000334`

White, R. and Dietenberger, M. (2001). Wood products: Thermal degradation and fire. In: Buschow, K.J., Cahn, R.W., Flemings, M.C., Ilschner, B., Kramer, E.J., Mahajan, S. and VeyssiÃ¨re, P. (eds.), *Encyclopedia of Materials: Science and Technology*, pp. 9712–9716. Elsevier, Oxford. ISBN 978-0-08-043152-9.
Available at: `https://www.sciencedirect.com/science/article/pii/B0080431526017630`

Wiecki, T. (2015 Nov [Online]). MCMC sampling for dummies. While My MCMC Gently Samples, Blog.
Available at: `https://twiecki.io/blog/2015/11/10/mcmc-sampling/`

Zienkiewicz, O. and Taylor, R. (2000). *The Finite Element Method*, vol. Volume 1: The Basis. 5th edn. Butterworth-Heinemann. ISBN 0 7506 5049 4.