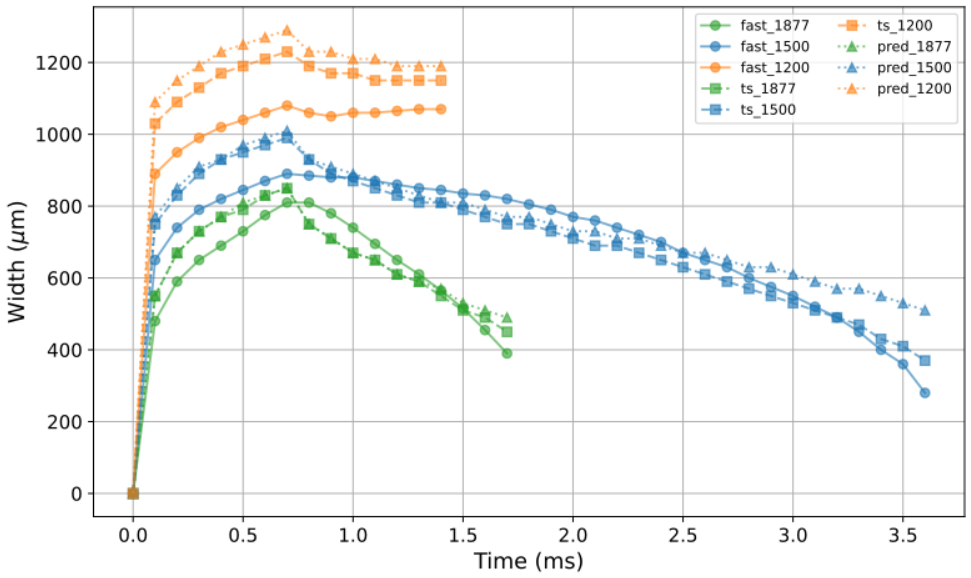
CHALMERS
UNIVERSITY OF TECHNOLOGY



Creation of a digital twin for simulation of melt pool geometry in metal additive manufacturing

Deriving a translation between two physical models in additive manufacturing by tuning material-and process parameters

Master's Thesis Degree project report in Complex Adaptive Systems

OSCAR KARLSSON

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Creation of a digital twin for simulation of melt pool geometry in metal additive manufacturing

Deriving a translation between two physical models in additive manufacturing by tuning material-and process parameters

OSCAR KARLSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025

Creation of a digital twin for simulation of melt pool geometry in metal additive manufacturing
Deriving a translation between two physical models in additive manufacturing by tuning material-and process parameters
OSCAR KARLSSON

Supervisor: Erik Granhed, Colibrium Additive
Examiner: Mattias Thuvander, Chalmers Department of Physics

Master's Thesis 2025
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 (0)31 772 1000

Cover: The width of three isotherms, $1200, 1500, 1877\,\mathrm{K}$, in the melt pool from electron beam melting in Ti-6Al-4V. The graphs show a comparison between a multi-physics model (solid line with circles), the tuned values in an analytical thermal model (solid line with squares) and the predicted values from a digital twin of the multi-physics model (dotted line with triangles).

Typeset in LaTeX
Printed by Chalmers digitaltryck
Gothenburg, Sweden 2025

Creation of a digital twin for simulation of melt pool geometry in metal additive manufacturing
Deriving a translation between two physical models in additive manufacturing by tuning material-and process parameters
OSCAR KARLSSON
Department of Physics
Chalmers University of Technology

Abstract

Additive manufacturing (AM) is a production technique that allows for the creation of complex geometries and components in near net shape by fusing the material layer by layer. It is a versatile but expensive method and due to this cost, simulating the process is of utmost interest. Since it requires complex multi-physics models at a very small scale, producing simulations of desired accuracy takes a lot of time and computational power. Simpler analytical models have shown promise as they run in a fraction of the time, but at the cost of accuracy. The present work explores the possibility to train an analytical model by adjusting material and process parameters to compensate for the lack of physics. The goal is a digital twin of the more complex model, which generates melt pool geometries comparable to the multi-physics model in a fraction of the time. This was accomplished by minimising a cost function based on the width and depth of a number of isotherms in a simulated melt pool in an electron beam melting AM process. The work demonstrates the feasibility of using a low resolution analytical model with modified material and process parameters as a digital twin to a multi-physics based simulation. Thus demonstrating the feasibility of bridging the gap between computational efficiency and model accuracy. It also establishes a foundation for more cost-effective and scalable simulations in AM, advancing the potential for innovation in the field.

Keywords: Additive manufacturing, electron beam melting, titanium-64, simulation, optimisation, FastLab, AnTem, transfer function, digital twin.

Acknowledgements

I want to express my sincerest gratitude to my supervisor Erik Granhed for making this work possible. Likewise, I would like to express my gratitude towards my examiner Mattias Thuvander. Finally, I would like to thank all the people at Colibrium Additive for always providing help and enjoyable discussions, creating an enjoyable work environment throughout the thesis.

Oscar Karlsson, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AM	Additive manufacturing
ANOVA	Analysis of variance
DYCORS	Dynamic coordinate search
EBM	Electron beam melting
EI	Expected improvement
FVM	Finite volume model
IP	Intellectual property
LCB	Lower confidence bound
MARS	Multi Adaptive Regression Splines
MSE	Mean squared error
PBF	Powder bed fusion
RS	Random search
SRBF	Stochastic radial basis functions

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

k	Index for isotherms
S	Index for spatial direction width or depth
t	Index for time steps
i	Index tracking iterations
j	Index tracking combinations of parameters
p	Max number of parameter combinations
d	Index to track width, depth and length

Sets

$\mathcal{I}$	Set of isotherms used in simulation
$\mathcal{T}_k$	Set of time steps for isotherm k

Parameters

T_0	Background temperature
T_L	Liquidation temperature
T_S	Solidification temperature
λ_S	Thermal conductivity for solid material
λ_P	Thermal conductivity for powdered material
ρ_s	Density of solid material
ρ_P	Density of powdered material

C_P	Specific heat capacity
P	Beam power
α	Material specific constant related to thermal diffusivity
σ	Focal width of electron beam
t	On time for electron beam

Variables

P_{scale}	Multiplicative scale parameter to adjust input power
ρ_{scale}	Multiplicative scale parameter to adjust input density
T_{scale}	Multiplicative scale parameter to adjust input background temperature
σ_{scale}	Multiplicative scale parameter to adjust input focal width
λ_{scale}	Multiplicative scale parameter to adjust input thermal conductivity
x_{ij}	Melt pool geometry with jth combination of parameter for ith iteration
b_j	Effect of parameter combination j on melt pool geometries
z_{ij}	Binary variable to indicate if specific parameter combination is applied or not
e_i	Residual error or noise for iteration i
a_m	Value of an analytical model in dimension m
s_m	Value of a multi-physics model in dimension m
$y_{S,k,t}$	Target value in FastLab for isotherm k
$q_{S,k,t}$	Current value for used analytical model and isotherm k

Contents

List of Figures

List of Tables

1
Introduction

Additive manufacturing (AM), also known as 3D-printing in some fields, is a production technique that allows for the creation of complex geometries and components by constructing them layer by layer [1]. Unlike more traditional methods, such as casting, where the limitations of the cast limit what you can create, AM is often able to avoid these problems by directly fusing the material into the finished shape [2]. Among the many methods used in AM, plastic printing is perhaps the most widely known, but one could also use metals. For metals there are several different approaches to fusing the material such as binder jetting [3], laser [4] and electron beam melting (EBM) [5]. EBM is a layer-wise powder based fusion (PBF) process that uses an electron beam to melt metallic powder [5], a process that allows for high-accuracy production of components with excellent mechanical properties [5]. This method also opens up for the possibility to work with what is known as ”hard-to-weld” materials. These are metallic alloys with desirable properties such as high cooling rates, but they are usually difficult to work with when using conventional methods [6]. Another advantage is that the components will be produced in a near-net shape, a state that requires little to no post-processing [7]. These beneficial properties of EBM also make it suitable for more common materials such as titanium. As a result, it is excellent for applications in aerospace and biomedical fields [2], [5], making it a versatile and effective construction method. Notable is that EBM is limited to electrically conducting materials, which means that it cannot process materials such as plastics [5], [8].

Additionally, unlike the more traditional methods, such as casting, PBF has a lot more degrees of freedom as the process can be controlled down to a much smaller scale. Because of this, being able to observe the process in-situ is of utmost interest. Naturally, being able to predict the process also becomes an interest, but this requires simulations across many levels of physics. In these simulations, approaches vary from comprehensive multi-physics models to simpler coarse-grained thermal conductivity models [2], [9]. One of these many levels of physics involves having an understanding of the solidification dynamics in order to be able to predict the microstructural outcomes [10]. This process is discussed in great detail by Stump et al. [11] who analyses the dynamic material properties, vaporisation, radiation and fluid flow and which of these and when can be included or neglected in finite volume models (FVM). The main takeaway is the impact of latent heat on the solidification conditions for which they implement a posterior correlation factor to allow for these simpler analytical models to predict the solidification process in place of a multi-physics model [11]. Bridging these different levels of physics across these scales can prove challenging due to different model assumptions and simplifications. Ensuring that simulation tools agree in overlapping domains is essential to maintain predictive accuracy while reducing computational cost. One of the many AM producers looking into

these possibilities is Colibrium Additive. To accurately depict their AM process they primarily make use of a multi-physics model, or more precisely a Lattice-Boltzmann model. However, because this sophisticated model is rather time consuming, they want to look into this approach of bridging the physics with a more basic analytical thermal model, as it could perform the simulations in a fraction of the time. Using an approach by Malmberg and Wallenås [12] and continued by Forslund et al. [13], simulations show promise. To further support this implementation of an analytical model, an idea tested by Stump et al. [11] has shown that it is possible to train an algorithm to slightly alter the material-and-process parameters in the model to compensate for the lack of physics, resulting in simulations close to the more sophisticated multi-physics model in the computation time of an analytical model.
This thesis therefore aims to take this idea further and establish a mapping between a known multi-physics model and a more computationally efficient known analytical thermal model. It will be done by calibrating the material-and-process parameters in the latter using simulations from the former as training and validation. These calibrated parameters will then be used to find a transfer function between the models, resulting in a digital twin of the Lattice-Boltzmann model that is capable of producing high-quality results in the computational time of an analytical model. Thus, demonstrating the feasibility of bridging the gap between computational efficiency and model accuracy.

1.1 Limitations

Due to the main focus of the project being to optimise a simulation model, the theory behind the AM and EBM process and the physics behind the models will not be in focus. Especially not for the Lattice-Boltzmann model whose main purpose is to train the program and which will be treated as the equivalent to physical experiments. However, enough theory will be studied to create an understanding of the model. The same applies to the analytical model, which will be treated as something close to a black box that performs the simulation. The model will not be changed as it needs to be kept as is for the company and it will be the only method of simulating the process. No other artificial neural networks or other similar methods will be implemented. The data will also be limited to what Colibrium Additive can generate with their existing models and measurements. Other limitations were put in place on the parameters in the models which are discussed in greater detail in Chapter 3.1, the main takeaway is that there will be no altering of the time for which the electron beam is powered on, as this would in a sense alter the experiments being looked at. Further, the exact form of the final transfer function between the models will not be disclosed due to company policy, although its capabilities will be demonstrated through examples.

1.2 Outline of the work

As the project has several different parts, the outline of the work will be presented to hopefully simplify the reading. Chapter 2 explains the background to the problem, the two models used and the data. This is to give some insight into how they work and why some decisions were made. Chapter 3 presents the methodology of the work, such as the

cost function, parameter limitations and algorithms, primarily explaining how the results were achieved. Chapter 4 then presents the results in the form of numbers and graphs, showing comparisons between the models and the transfer function. This is followed by a discussion on the social, ethical and ecological aspects of what has been examined in this thesis in Chapter 5. Then Chapter 6 discusses deeper reasoning and complications encountered during the project along with a discussion of the results. Lastly, Chapter 7 summarises what has been done, the problems and potential future improvements, as well as a conclusion to the work.

2
Background

With the goal of creating a mapping between two models, the first model used was a Lattice-Boltzmann model that is able to produce simulations that agree well with physical experiments. The second model was a less accurate but significantly faster analytical thermal model. To use the first model, an in-house program known as FastLab, used by Colibrium Additive to simulate their EBM process, was utilised. It will be discussed in more detail in Section 2.1. All data used in this thesis was generated using FastLab and was treated as the ground truth for training and testing due to it agreeing well with reality. This data will be discussed in Section 2.2. To utilise the second model, another in-house program known as AnTem was used. It is based on the analytical approach by Malmberg and Wallenås [12], pursued by Forslund et al. [13] and is also used regularly by Colibrium Additive. AnTem is further discussed in Section 2.3.

2.1 FastLab

FastLab is an in-house program based on the Lattice-Boltzmann model to simulate the EBM-process. Through testing and verification with physical experiments, it is believed to accurately depict reality. Due to its high accuracy and the small scale it works in, it includes a lot of physics under a small time and length scale and as a consequence, is rather time consuming. Simulating $1\,\mathrm{ms}$ of the process takes roughly one hour of real time, meaning simulating the melting of a single point can take half a day. Hence, other methods are often desirable when working at larger scales. This is for the most part not a major problem as it is primarily used for simulating singular melt pools. FastLab was as such not used directly in the work of this thesis. Instead, since this work exclusively looks at point melting, it was used to create training data and validation data.

2.2 Data

The data used for the optimisation was produced using FastLab and was treated as reality due to its accuracy and the complicated and timely process of using actual experiments as reference data. The data were generated for Ti-6Al-4V (Ti64) and consisted of four process parameters; power P, dwell time for the beam t_d, spot size σ and background temperature T_0. But also three material parameters; specific heat capacity C_P, density ρ and thermal conductivity λ. Values of the material parameters along with the solidification temperature T_S and the liquidus temperature T_L of Ti64, can be seen in Table 2.1. However, the models do not look at ρ and C_p as individual parameters, but instead considers the product of them, the volumetric heat capacity ρC_p. Neither do they consider

the thermal conductivity, but instead the thermal diffusivity defined as $\frac{\lambda}{\rho c_p}$. This simplifies the process as the volumetric heat capacity can be treated as a single value, and as such only two parameters need to be altered for all three to be affected.

Parameter	Value	Unit
T_L	1877	K
T_S	1933	K
λ	12.43	$\mathrm{W{\cdot}m^{-1}{\cdot}K^{-1}}$
ρ	4430	$\mathrm{kg{\cdot}m^{-3}}$
C_p	567	$\mathrm{J{\cdot}\ kg^{-1}{\cdot}K}$

Table 2.1: Predefined parameter values from FastLab simulations.

The data were formatted as tables consisting of ten isotherms, that is, $1000, 1100, 1200, ...,$ 1800 and $1877\,\mathrm{K}$. The last is the assumed solidification temperature under the given conditions. It also contained the width and depth [μm] of each isotherm in the substrate, all in steps of milliseconds from the moment the electron beam was powered on. Data were generated for various combinations of values for the four process parameters. Most in ranges used in actual production, but also with outliers of more extreme values. For example, a beam power and background temperature combination too low to properly melt the material. These outliers were included to ensure that the program worked properly not only for parameters relevant for Ti64, but also under conditions used for other materials to open up for other applications in the future. Another important detail of the data was that it exclusively involved point melting, meaning that we only had to look at a single symmetric point in a known spot each time, simplifying the process. Including several points or more complex geometries have been discussed as a future implementation, but are not on the company agenda as of today.
Moreover, the data were generated in two forms, one set of higher resolution and one of lower resolution. The difference was a factor of eight in both the width and depth of the melt pool. The high-resolution data did, however, have the risk of encountering what internally at the company is known as the *"0.7 Problem"* which can cause a large amount of the simulation to become invalid due to an old implementation to enhance visual clarity. This problem is explained in greater detail in Section 2.2.1 and its side effects are discussed with examples later in Section 6.2.1. The data were still usable and in a good enough condition to produce results, despite this, the low-resolution data were tested. Mainly because the real effect of the resolution was not very well known. But also because it is not affected by this 0.7 problem. Consequently, resulting in a substantial increase in data points. However, these low-resolution data were somewhat experimental and, at times, had problems that could render entire isotherms unusable, which inevitably led to the decision of not using it.

2.2.1 The 0.7 problem

There are some known problems that affected the work, one of them being what internally at Colibrium Additive is known as the *0.7 Problem*. It affects the FastLab data, which in this case are both training data and validation data. The problem is as follows. Initially,

for the Lattice-Boltzmann simulations used to generate the data in this project, all voxels in FastLab are assigned a negative value. Once it reaches the liquidus temperature $T_L \approx 1933\,\mathrm{K}$, the boundary temperature above which the material is completely liquefied, the sign changes to positive, making tracking the perimeter of the melt pool an easy task. However, when the temperature decreases again and reaches the solidification temperature $T_S \approx 1877\,\mathrm{K}$, the boundary below which the material is completely solid, its temperature is hardcoded to $0.7 \cdot T_S$ and remains so throughout. This makes it impossible to investigate the temperature decline and consequently the solidification process, causing a considerable loss of data due to the width and depth stagnating with constant temperature. The reason behind this implementation is that it is a left-over from an older project where this method greatly enhanced the visual clarity of what was to be investigated, something that as of today is no longer necessary. But as the implementation lies in the source code, which is handled by a different branch of the company, there have been no possibility of reverting this implementation back to how it is supposed to be. This also means that there is no possibility of retrieving the lost values as they are lost somewhere in the simulation process, which in some cases resulted in the need of removing a major portion of the data points in order for the program not to train on a non-realistic behaviour. The side effects of this problem along with examples of it is discussed further in Section 6.2.1.

2.3 AnTem

The analytical model used in this thesis is based on the approach described by Malmberg and Wallenås [12] and continued by Forslund et al. [13]. They derived a solution to the heat equation using a half-space subject to a moving Gaussian heat flux with piecewise constant parameters. Their approach was to neglect all the mechanisms but the heat conduction, which is assumed to be dominant. This allows the whole process to be described by the heat equation as in (2.1).

$$\rho c_p \frac{\partial T}{\partial t} = \lambda \nabla^2 T + Q \tag{2.1}$$

Where ρ is density, c_p is heat capacity, T is temperature, t is time, λ is thermal conductivity and Q is the gaussian heat source. Q is in this case defined according to (2.2).

$$Q = \frac{P}{2\pi\sigma} \exp\left(-\frac{(x-x')^2 + (y-y')^2}{2\sigma^2}\right) \tag{2.2}$$

Here, P is the beam power, σ the beam spot size, $(x-x')^2 + (y-y')^2$ is the distance between the centre of the beam and the point of interest. Then it is possible, by assuming a semi-infinite domain with $z = 0$, to solve the heat equation with Fourier analysis according to (2.3) [13].

$$T(x,y,z,t) = \int_0^1 T^* \frac{1}{1+s^2} \exp\left(-\frac{(x-Vs^2)^2 + y^2}{1+s^2} - \frac{z^2}{s^2}\right) ds. \tag{2.3}$$

Where $V = v_x\sigma/2\sqrt{2}\kappa^2$, where v_x is the constant speed of the beam and $\kappa^2 = \lambda/\rho C_p$ is the thermal diffusivity. This convolution integral can then be evaluated using a numerical

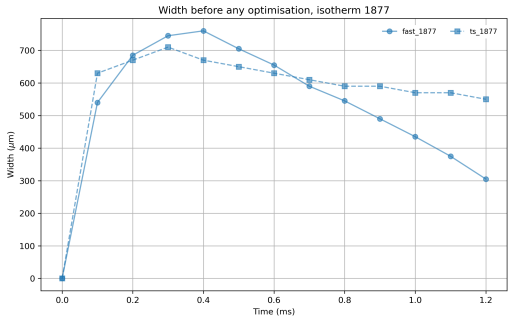
quadrature scheme, as in (2.4). This possibility of solving it analytically is also a consequence of the semi-infinite domain as there will not be any boundary conditions except at $z = 0$.

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \tag{2.4}$$

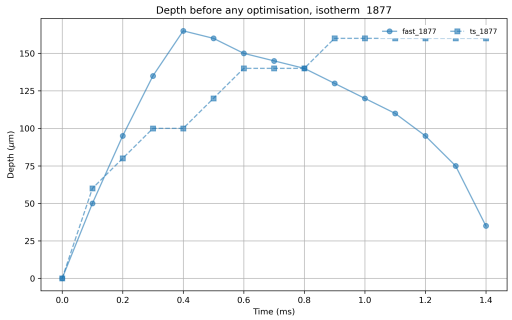
This specific model is of particular interest for PBF processes like that of EBM due to the possibility of using these parameters to control the conduction of heat caused by the electron beam scanning the material with energy concentrated in a spot-like behaviour. Further, the authors translate the problem into a dimensionless form to help generalise the heat equation. Making it applicable on various materials and input parameters, allowing for an easier comparison between them due to the reduction of independent variables [12], [13]. Finally, the implementation of the quadrature scheme to compute the numerical integrals simplifies the calculations further as having to solve an analytical heat equation in each step can prove challenging and time consuming [12], [13]. This implementation also creates one of the most appealing features of the model, namely the possibility of performing most calculations in isolation, as they will be independent of each other. Thus, combining the dimensionless translation of the heat equation and the numerical calculations using quadrature schemes by [12], [13] with the simplification of the solidification conditions by [11], it is possible to accurately simulate the EBM process on metals using a PBF process without using a multi-physics model like the aforementioned Lattice-Boltzmann model. This approach is the foundation of the in-house program AnTem used for the analytical approach to the simulations.

AnTem is, as such, based on the analytical model by [12], [13] to simulate an EBM-process. It approaches the problem by making an analytical solution to the temperature history of the process. The main benefit of this approach is the time efficiency as it only takes seconds or minutes to perform the corresponding simulation that takes FastLab hours. However, AnTem is assumed to be less accurate due to its lack of physics. Nevertheless, with them producing fairly similar results, the idea sparked that one might be able to slightly alter the input parameters of AnTem in order for it to produce results more comparable to FastLab. The main argument for altering the parameters is that the programs are based on entirely different physical models, resulting in the same input most likely not producing the same output, and correction will be needed to compensate for the lack of physics. Another major assumption that laid the foundations of this work was that since AnTem in practice is the analytical solution of the heat equation, it should behave very similar to FastLab at the border of the melt pool due to FastLab barely making use of any other physics there. But because of its simplicity, what it will fail to include are properties like fluid dynamics in the form of liquefied material being pushed out by the electron beam, as well as keyholing. A phenomenon in welding that creates a pocket of vaporised metal in the melt pool [14] that will affect the material around the point being studied. As for a comparison and visual explanation of the difference between FastLab and AnTem we can look at the width and depth of a single isotherm, in this case $1877\,\mathrm{K}$, in the melt pool produced by these two programs. As seen in Figure 2.1, there is a clear difference in the size and shape of both the width 2.1a and depth 2.1b but also in step size. That means that FastLab can have values with much higher precision than AnTem. Therefore, a perfect alignment will most likely not be possible, but with the correct ad-

justments, there is a possibility for them to follow each other fairly closely and provide you with an estimate how the other behaves.



(a) Width comparison of FastLab (solid line) and AnTem (dotted line) using the initial values of FastLab with no adjustments.



(b) Depth comparison of FastLab (solid line) and AnTem (dotted line) using the initial values of FastLab with no adjustments.

Figure 2.1: A comparison between FastLab (solid line) and AnTem (dotted line) using the original parameters retrieved from FastLab to illustrate the difference in the melt pool geometrics.

The idea had previously been tested to implement an optimisation algorithm that changes the input parameters to try and match the max width, depth and length between a multi-physics model and an analytical model for a line melt [11]. Iteratively finding analytical solutions closer and closer to that of the multi-physics model, a process that is described in more detail in Section 3.1.1. However, the idea was not thoroughly implemented and instead seen as a future project by Colibrium Additive. Moreover, having the models produce the exact same results was never considered due to the difference in resolution making it impossible in most cases. An important detail is that improving on the analytical model by adding more physics to it would not be feasible without losing its analytical properties, and with it, a lot of the positive aspects of the model. One of these major perks that likely would be lost is the ability to alter values freely in many steps of the simulation. This feature stems from the fact that most calculations are made in isolation and do not depend on each other. Allowing for a great deal of freedom that opens up for the possibility to easily simulate more complex behaviour by combining smaller, simpler simulations.

Even if this idea was not implemented into the work of the company, it laid the foundation for this project as it was hypothesised that one could improve on an idea that showed promise but with new and improved data. After which one was to ensure it worked for vastly different simulations, record how the input parameters needed to be altered for different kinds of parameter combinations to agree with FastLab. Eventually, translating these alterations into a transfer function between the models. Resulting in a digital twin of FastLab, based on AnTem, that can produce results comparable to a multi-physics model in the time it takes to run an analytical model. Notable is the aforementioned difference in resolution between the models, meaning that the digital twin will likely differ from reality. It will, however, provide you with a strong indication of what will happen and if something is likely to fail when testing the EBM-process in the laboratory. Saving

both time and material. However, due to these potential flaws, FastLab is likely to be the preferred model to simulate the process as a final step before stepping foot in the laboratory. But instead of confirming the feasibility of an idea in days, you could do it in seconds or minutes with one or several AnTem simulations.

3

Methodology

The methodology for finding the transfer function will be a process of tuning AnTem for different parameter combinations. Each set of parameters will be optimised by tuning the process-and-material parameters in AnTem until the output of the width and depth of each of the isotherms of the melt pool agrees with the original data from FastLab. The idea behind the approach is due to the lack of physics one could treat the calculations in AnTem as a black box where the "how" is not interesting, only the final geometry of the melt pool. Thus, altering the parameter values slightly and consequently altering the physics to some extent is not seen as a problem as the final output becomes more realistic. This can then be repeated across multiple parameter sets, allowing for a mapping to be constructed from FastLab's inputs to AnTem's tuned values. These mapped values can then serve as training data for the desired transfer function. This process is what will be explained in this chapter, how the optimisation algorithm was derived, how it worked, and how it was used to produce a transfer function between FastLab and AnTem.

3.1 Optimisation of AnTem

The first step of optimising AnTem was performed as follows. By initially using the same parameter values as FastLab, along with defining the spatial and temporal domain and respective step size to match that in FastLab, AnTem was able to produce similar data to FastLab. The parameters that were investigated during the project were the background temperature $T_0\,[\mathrm{K}]$, the thermal conductivity $\lambda\,[\mathrm{W}\cdot\mathrm{m}^{-1}\cdot\mathrm{K}^{-1}]$, the density ρ $[\mathrm{kg}\cdot\mathrm{m}^{-3}]$ and the focal width of the electron beam $\sigma\,[\mathrm{mm}]$ along with the beam power P [W]. The last two and T_0 being unique values for each simulation retrieved from FastLab. A fourth parameter unique for each simulation was the time for which the beam was powered on t_d, but it was treated more like a constant than a variable. Further, the heat capacity C_P was seen as a constant due the volumetric heat capacity and thermal diffusivity being of interest, resulting in only two out of ρ, C_p and λ needing alteration for all three to be affected. Consequently bringing the total parameters of interest down from seven to five, those being $[P, \sigma, T_0, \rho, \lambda]$. The goal then was to alter these parameters in a way to make AnTem produce the same isotherm geometry in the melt pool as FastLab. The fact that they ultimately used different material properties was neglected due to them being based on different types of physics and that the final geometry was what was of interest. The span in which they were altered was moreover considered small enough to further support this claim. In order to achieve the later goal of finding a relationship between the two simulation programs, a set of multiplicative scale parameters was introduced $[P_{scale}, \sigma_{scale}, T_{scale}, \rho_{scale}, \lambda_{scale}]$, each limited to different ranges. These were

implemented to control and observe how each parameter was altered. P_{scale} was limited to $[0.3, 1.0]$ since a power higher than the input was not deemed reasonable. But a lot of energy can be lost. Experiments by Colibrium Additive show that as little as $50\,\%$ can be absorbed, the rest is reflected, radiates, or evaporates away. For $\sigma_{scale} \in [0.5, 1.5]$, a larger range supported by the fact that the actual value of the focal width is not known with a high certainty. For $T_{scale} \in [0.8, 1.2]$, a value larger than 1 was argued for due to the lack of physics in AnTem compared to FastLab, meaning we lose energy, partly in the form of latent heat which might need to be compensated for [11]. The density parameter was barely allowed to change $\rho_{scale} \in [0.99, 1.01]$, this was mainly due to it barely changing at all when testing, meaning we could restrict its range to simplify the optimisation. Finally the thermal conductivity scale parameter $\lambda_{scale} \in [0.8, 1.5]$, a range of values that support its property of being a dynamic parameter in both FastLab and reality that changes somewhat quickly with temperature [15]. Initially, the value was 1.0 for T_0 and ρ and 1.4 for λ, the reasoning behind these values was that they typically landed close to these when testing. As for P and σ, these varied a bit more, which prompted a more exploratory search method that involved rerunning the optimisation with several different initial values in their respective range of values. This exploratory method was mainly used to help the algorithm take larger steps, which otherwise took a lot of time when this many variables were involved. One could also argue that you could keep ρ fixed due its small range to simplify the optimisation, but the effect on the results was noticeable enough to want it to be able to vary slightly.

3.1.1 Cost function

As a measure of goodness of fit, a cost function was developed for the algorithm to know what impact the changed parameters had on the melt pool, what parts of the pool were affected and by how much. Stump et al. [11] showed that this was possible using similar data for the more common method of line melting. Their approach was to fit the maximum value of the width, depth, and length of the melt between an analytical model and a multi-physics model according to (3.1). Here, s_d is the target value in a multi-physics model for the maximum of dimension d (width, depth, length) and a_d is the maximum value in the analytical model for dimension d at steady state [11].

$$\text{Cost} = \sqrt{\frac{\sum_{d \in width, depth, length} (1 - \frac{a_d}{s_d})^2}{3}} \tag{3.1}$$

Stump et al. [11] limited their training data to the maximum value of width, depth and length at steady sate. With data from FastLab we can expand on this idea. Starting from (3.1), we replace the three-dimensional index d with the two-dimensional S (width and depth) as we now look at a point melt. We add a new index k to indicate what isotherm is being looked at and the time step index t to include the whole melt process, leading to (3.2). Consequently $q_{S,k,t}$ represents the AnTem value for a given dimension, isotherm and time step, the same goes for the FastLab value $y_{S,k,t}$. Hence, the cost is calculated individually for each isotherm k and individually for width and depth. Additionally, w_k are isotherm specific weights that balance out the impact of them having different amounts of data points. This was however not enough as some isotherms were harder to fit and thus required a higher weight which was added into this weight to help the optimisation.

Further, as it was noted that fitting the larger and smaller isotherms often led to the ones in the middle adjusting themselves, the outer ones consequently received slightly larger weights to encourage this behaviour. w_S is a weight specific for width and depth, mainly because the depth was more difficult to optimise. Leading to all calculations being on the same form for both width and depth, and for each isotherm. The weights are multiplied with the sum of the differences between the current AnTem $q_{S,k,t}$ value and the target FastLab value $y_{S,t,k}$ to a power of four for all time steps in the current isotherm, then normalised against the sum of the average of these to a power of four. Finally, sum the cost of the width to the cost of the depth for all isotherms in order to get the total cost of the problem. Additionally, for the cases when the denominator was 0, which only occurred when both points were 0, since there cannot be any negative values, the cost was set to 0, as the points coincided.

$$\text{cost}_{S,k} = \begin{cases} 0, & \text{if } q_{S,k,t} = y_{S,k,t} = 0 \\ w_S w_k \left(\dfrac{\sum_{t \in \mathcal{T}_k} \left(\left(q_{S,k,t} - y_{S,k,t} \right)^4 \right)}{\sum_{t \in \mathcal{T}_k} \left(\left(\frac{q_{S,k,t} + y_{S,k,t}}{2} \right)^4 \right)} \right), & \text{else} \end{cases} \tag{3.2a}$$

$$\text{Total cost} = \sum_{k \in \mathcal{I}} \text{cost}_{depth,\,k} + \sum_{k \in \mathcal{I}} \text{cost}_{width,\,k} \tag{3.2b}$$

3.1.2 Optimisation methods

The different cost functions were tested using several different optimisation algorithms in Python. To start, more common ones such as NLopt [16] and scikit-learn [17] were used. These did unfortunately struggle a lot to fit AnTem after FastLab, taking hundreds of iterations to even get close. Likely getting stuck in spurious minima [18]. Eventually, the Python library pySOT [19] was tested. It is an optimisation toolbox made for computationally expensive problems, using a surrogate model to approximate the problem [19]. This significantly sped up the process, allowing for sufficient results to be reached in tens of iterations instead of hundreds.

When all relevant parameters, optimisation methods and the cost function were decided, the optimisation could, in simplified terms, be performed according to Algorithm 1. More detailed descriptions of the program functions can be found in Appendix A. The method was as follows. Initialise the FastLab values and handle any potential problems. Initialise AnTem, set up all simulation settings and generate data in the same form as the FastLab data, remove points from AnTem that were problematic in FastLab to match the dimensions of the data, as there is no point in including data points with no counterpart to fit against. Compute an initial cost between the models, send it to pySOT along with the scale parameters and then let it search for better solutions. When a promising point was found, recompile the AnTem simulation with the new point and repeat the cost calculation and the pySOT search for better values. When the difference was small enough or enough iterations had passed, return the best found solution and recompile AnTem with these values.

Algorithm 1 Optimisation process of Simulation Parameters

```
Define constants and initial values, example n = number of isotherms
function GET_GET_FASTLAB_DATA(Initial values)
    Initialise FastLab settings with given values
    function READ_FASTLAB_DATA(Settings)
        Read melt pool geometries and save to Data frames
        Process problematic data
        Return melt pool geometries
    end function
end function
Simulated data = Initial run of AnTem(P, σ, T_0, ρ, λ, c_p, t_d, isotherms)
Calculate initial cost
function MAIN(Dataset)
    for i = 1 to floor(n/2) do current isotherms = isotherms[0:i] + isotherms[-i:]
        function OPTIMISE COST FUNCTION(current isotherms, Simulated data, Scale
Parameters, max_iterations)
            Initialise pySOT[19]
            Optimise cost function with pySOT
            Save best result
        end function
    end for
    Rerun AnTem and calculate cost function with best found parameters
end function
```

3.2 Transfer function between the models

By producing enough scale parameters, a transfer function can be calculated by letting a Python toolbox such as scikit-learn [17] perform a polynomial fit on the scale parameters and the initial values. Using hyper-parameter tuning with cross-validation over $1e^6$ iterations and a tolerance of $1e^{-4}$ for three different models, comparing them and using the results with the smallest error, a higher order polynomial can be derived. This expression can then be used to instantly calculate the final scale parameters for AnTem, given some input parameters that were to be used in FastLab.

4
Results

Below follows the results of a sensitivity analysis, the results for the different optimisation steps and the results of the derived transfer function.

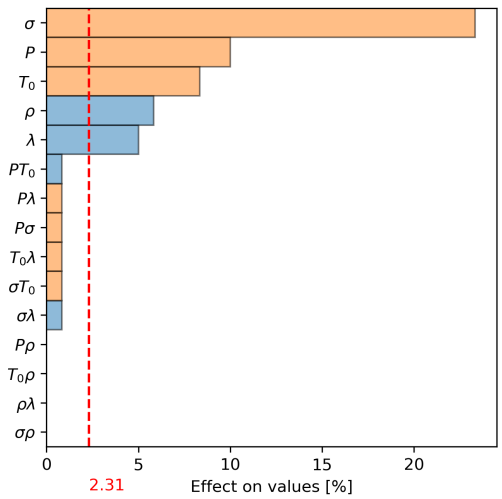
4.1 Sensitivity analysis

Before starting the optimisation, a sensitivity analysis was conducted to investigate if the variables were of importance, how much they affected the results, and if any could be deemed to have a low enough importance to be considered constants. To do this, a regression analysis and an analysis of variance (ANOVA) were performed by reformulating the ANOVA as multiple regression problems, as described in [20]. In practice, this acted as a sensitivity analysis that was simple to implement on this kind of problem by making use of functions in the Statsmodels Python library [21]. This method allows for this kind of assessment by modelling the relationship between the parameters and the final result of a simulation by treating each factor or combination of factors as a quantitive variable. These regression equations take the form of (4.1) [20].

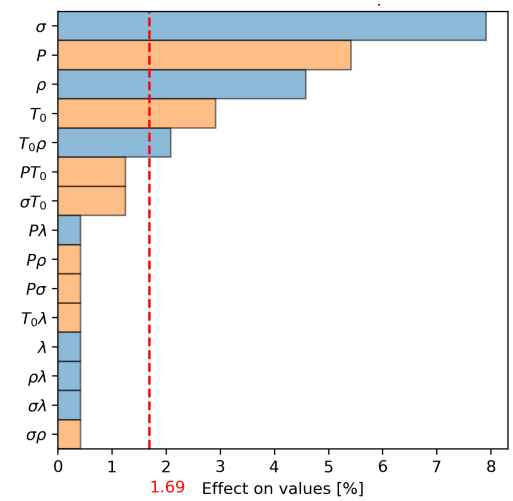
$$x_{ij} = b_1 z_{i1} + b_2 z_{i2} + \cdots + b_p z_{ip} + e_i \tag{4.1}$$

Where $x_{i,j}$ are the melt pool geometries with the *j*th combination of parameters in the *i*th iteration of the same calculation. b_j is the effect of parameter combination *j* on the geometries. Incorporated into b_j is also a form of step length which decides how large or small the increase or decrease is when varying the parameters, a common value is between one and a few percent. Additionally, $z_{i,j}$ is a binary variable that specifies if that specific parameter combination is applied or not (1 if it is, 0 otherwise) and e_i is a residual error from noise or other error sources.
An example of this analysis with a $95\,\%$ confidence interval and changing the parameter values by $10\,\%$ can be seen in Figure 4.1. The red dotted line represents the bound for the confidence interval. The x-axis indicates the effect this $10\,\%$ change had on the width or depth, an orange colour represents an increase and a blue colour a decrease. The variables examined with the regression analysis were the beam power and spot size, density, initial background temperature, and thermal conductivity. This process was done in hope of some variables, which at a start was thought to be of significance to the optimisation, instead to be fixed due to a low significance, simplifying the process by reducing the number of variables. Even if the dwell time for the beam had been decided from the start to be kept fixed, it was evaluated with the analysis. Its impact was magnitudes larger than all other parameters, further supporting the decision of keeping it fixed as it could come to dominate the optimisation process. This would likely happen through the other

parameters being ignored due to their, in comparison, negligible impact. Additionally, because the specific heat capacity was treated as a constant, it was excluded from the analysis. In panel 4.1a, we see, among other things, that an increased spot size leads to a wider pool and from panel 4.1b we see that it also leads to a shallower melt pool, which is a reasonable behaviour. An important detail is that these results vary with the input parameters as different combinations of parameters have different sensitivity to different adjustments. Meaning that the results might look different when rerunning the calculation if not all values are exactly the same. However, the results appeared roughly the same for different combinations of parameters, with σ having the highest impact, followed by P. T_0 and the material parameters often had a slightly smaller, but still significant impact. These conclusions also further supported the decision to manually change σ and P to find the best possible values, as they had the largest impact. In the end, all parameters have enough significance to be included and none can be kept constant. Additionally, what is of interest is not the exact effect each parameter has, but the general knowledge if they are of importance or not. Worth noting is that the analysis was not done by hand, but by rewriting a Python script made by the company for similar implementations to better fit this problem. The results were also corroborated with earlier results and the commercial statistics software package MINITAB [22].



(a) Regression analysis showing the impact of a $10\,\%$ increase of the parameters on the width of the melt pool.



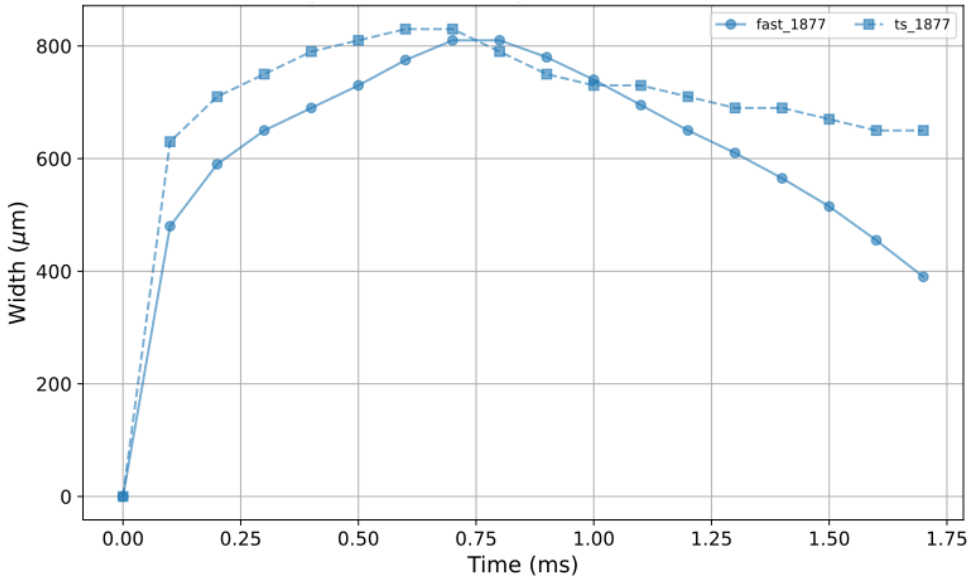
(b) Regression analysis showing the impact of a $10\,\%$ increase of the parameters on the depth of the melt pool.

Figure 4.1: Figures showing the parameters' different impact on the melt pool geometries based on a regression analysis. The red dotted line indicates the boundary for a $95\,\%$ confidence interval.

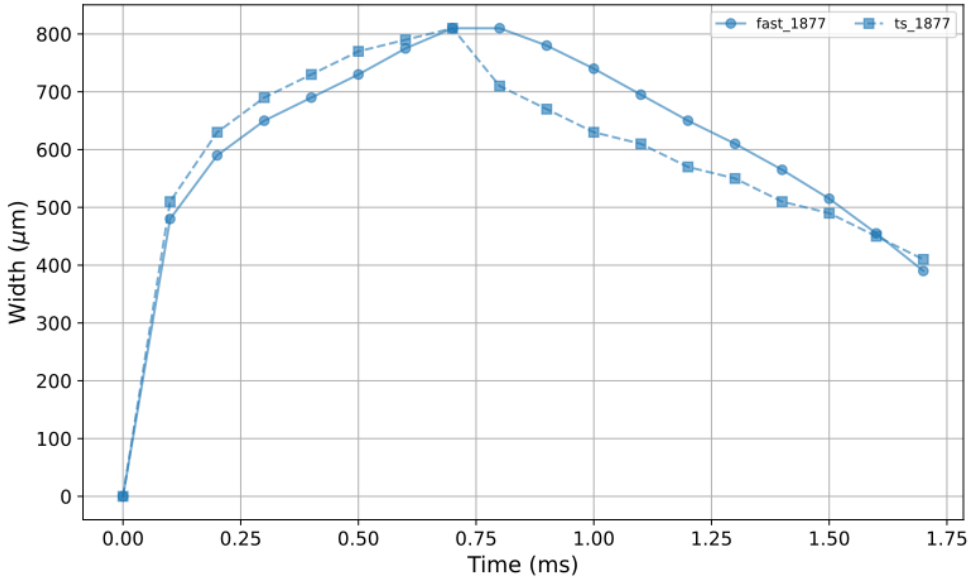
4.2 Optimisation of individual dataset

For the initial optimisation, the parameter set $[P, t_d, \sigma, T_0] = [360, 0.7, 750, 1073]$ was used. Initially, the optimisation was performed on a single isotherm to keep it simple and to ensure that the program worked as intended. The isotherm chosen for this was $1877\,\mathrm{K}$

as the solidification temperature is usually the most important and the most important to accurately simulate. A before and after comparison between FastLab and AnTem for the width can be seen in Figure 4.2 and the depth in Figure 4.3. The total cost before optimising was 6.869 compared to 0.596 after.

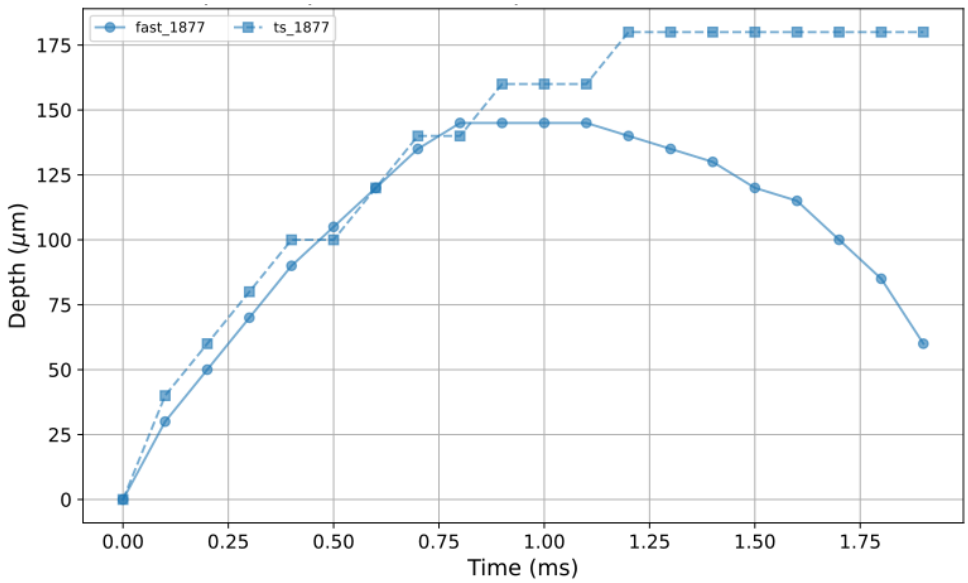


(a) Width comparison before optimisation on the $1877\,\mathrm{K}$ isotherm.

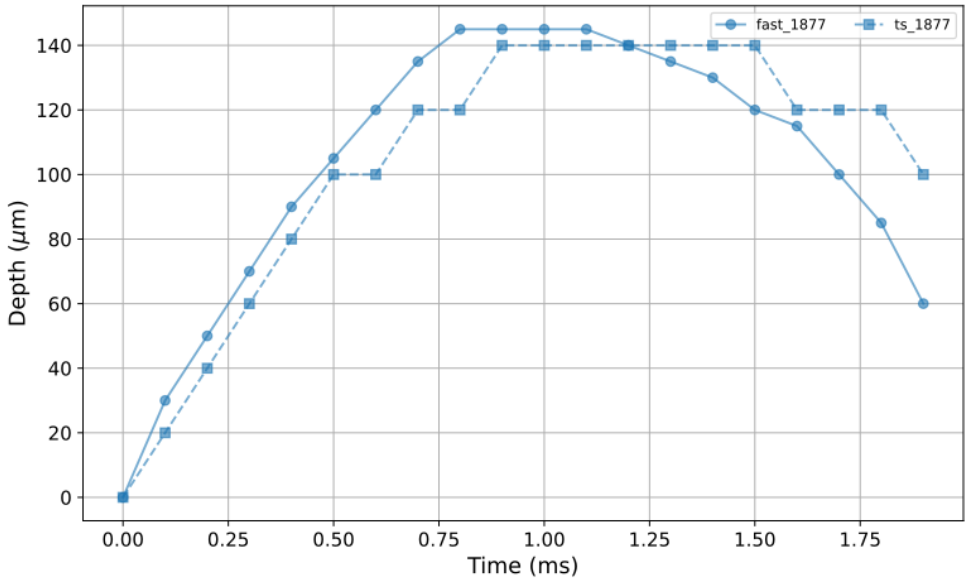


(b) Width comparison after optimisation on the $1877\,\mathrm{K}$ isotherm.

Figure 4.2: Width comparison between FastLab (solid) and AnTem (dotted) before and after optimising a single isotherm.



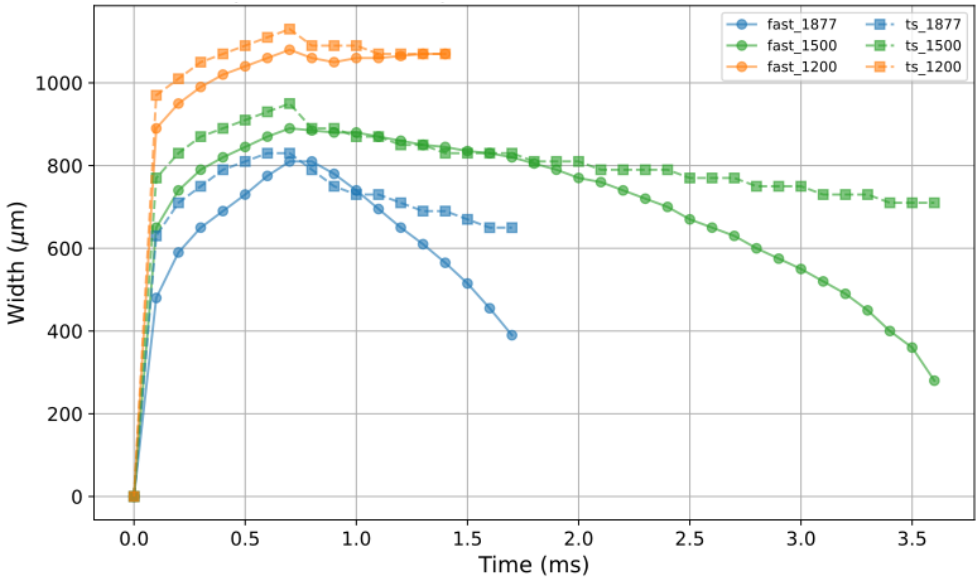
(a) Depth comparison before optimisation on the 1877 K isotherm.



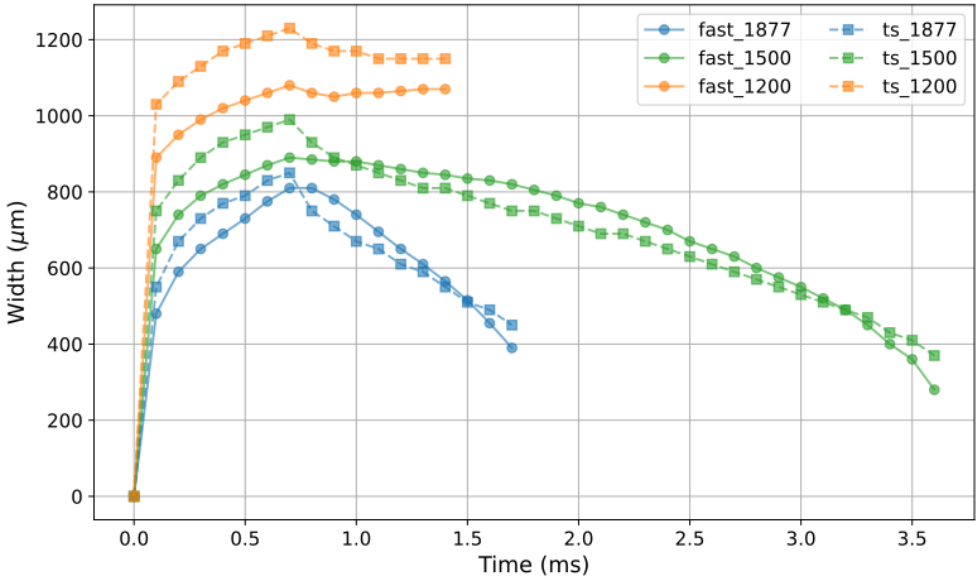
(b) Depth comparison after optimisation on the 1877 K isotherm.

Figure 4.3: Depth comparison between FastLab (solid) and AnTem (dotted) before and after optimising a single isotherm.

Next was to include more isotherms, starting with $[1877, 1500, 1200]$ to include both ends of the range as well as the middle. The before and after comparison for the width can be seen in Figure 4.4 and depth in Figure 4.5. The cost before optimising was 7.048 while after optimisation it was 0.0001.

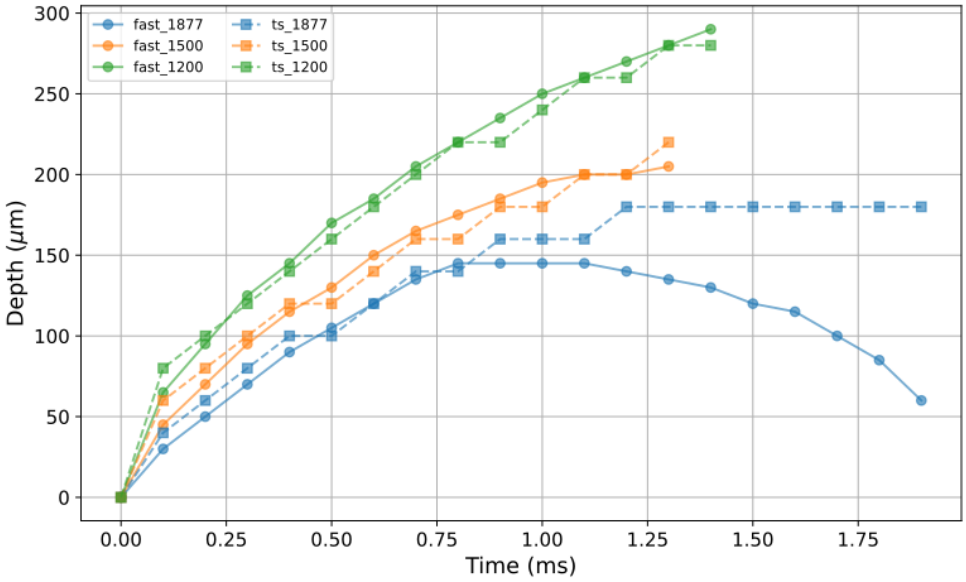


(a) Width comparison before optimisation on isotherms $1877, 1500$ and $1200\,\mathrm{K}$.

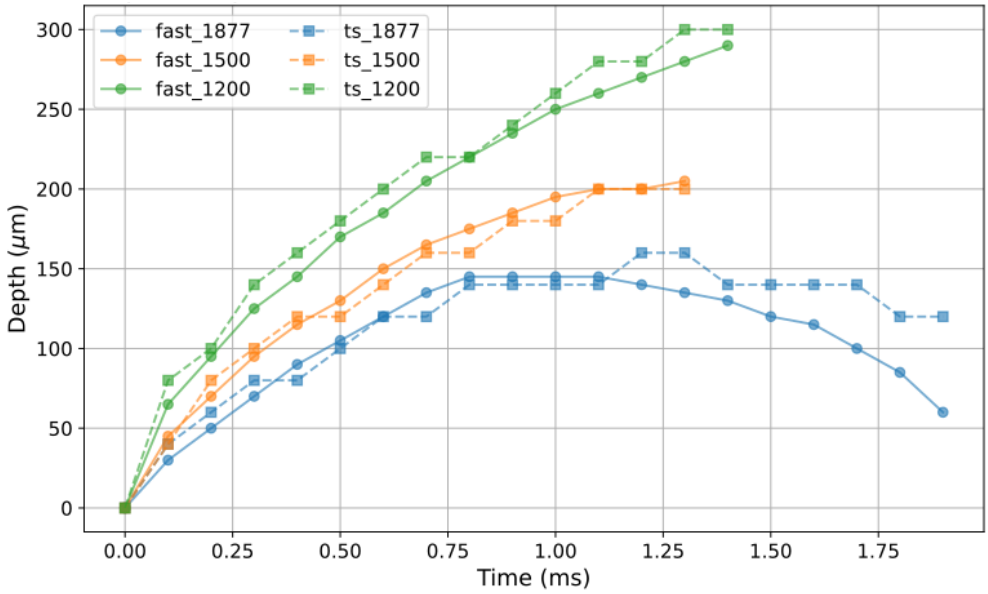


(b) Width comparison after optimisation on isotherms $1877, 1500$ and $1200\,\mathrm{K}$.

Figure 4.4: Width comparison between FastLab (solid) and AnTem (dotted) before and after optimising three isotherms.



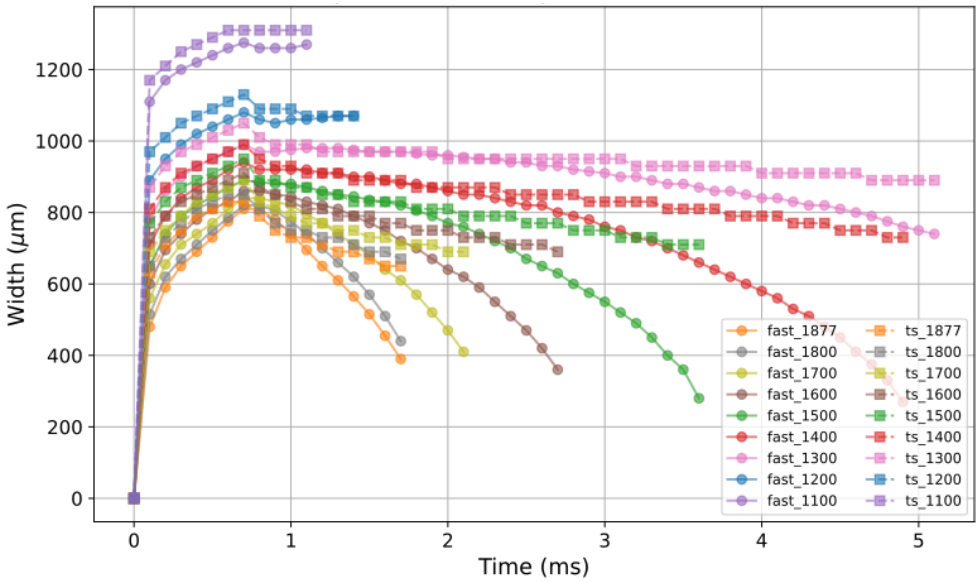
(a) Depth comparison before optimisation on isotherms 1877, 1500 and 1200 K.



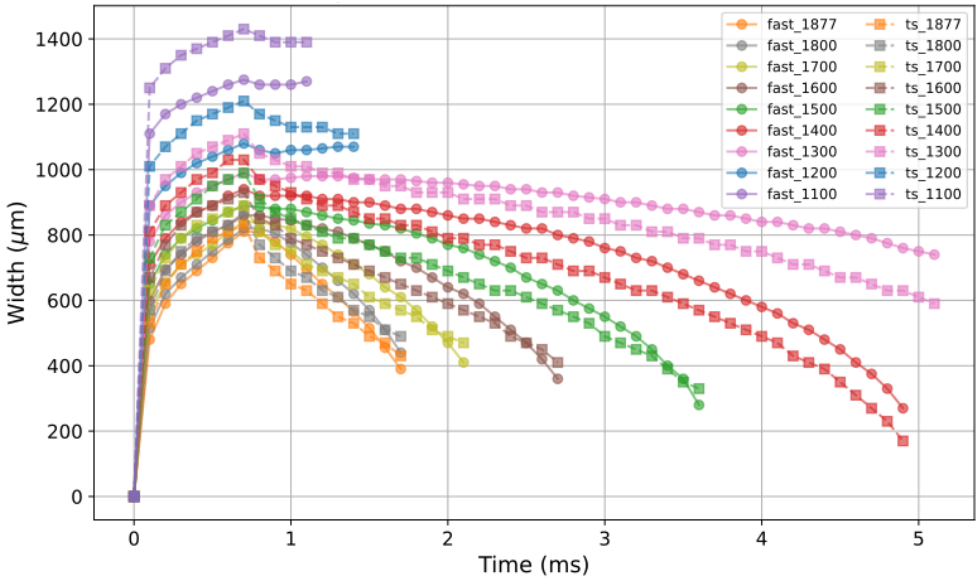
(b) Depth comparison after optimisation on isotherms 1877, 1500 and 1200 K.

Figure 4.5: Depth comparison between FastLab (solid) and AnTem (dotted) before and after optimising three isotherms.

Lastly, all 10 isotherms were included in the optimisation, those being 1000, 1100, 1200,..., 1800 and 1877 K. The before and after for the width can be seen in Figure 4.6 and the depth in Figure 4.7. The initial cost with no optimisation was 15.46 and the final cost after optimising was 0.001. Note that only nine isotherms are included in the graphs, this comes from the 1000 K being 0 throughout the whole simulation as a consequence of the background temperature being above 1000 K for this parameter combination.

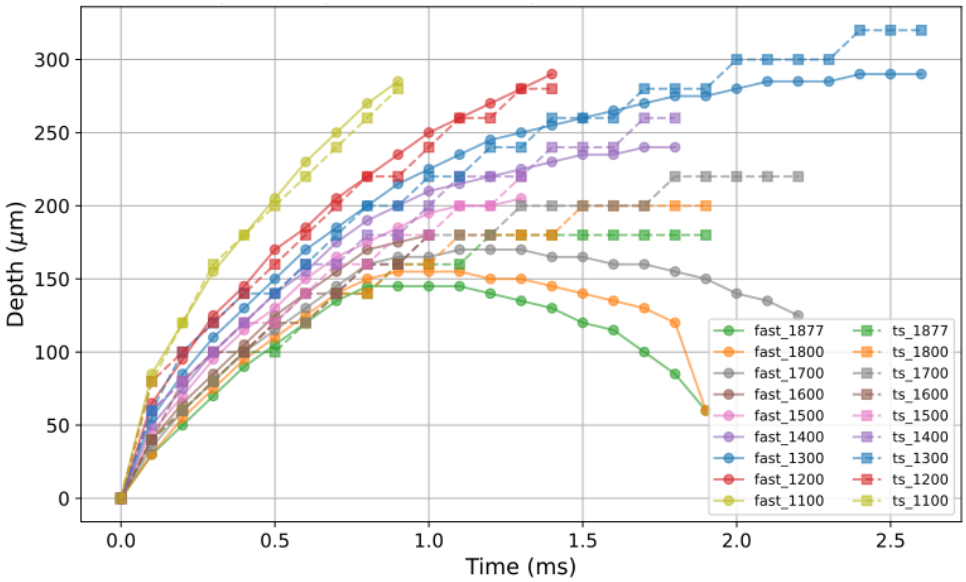


(a) Width comparison before optimisation for all isotherms.

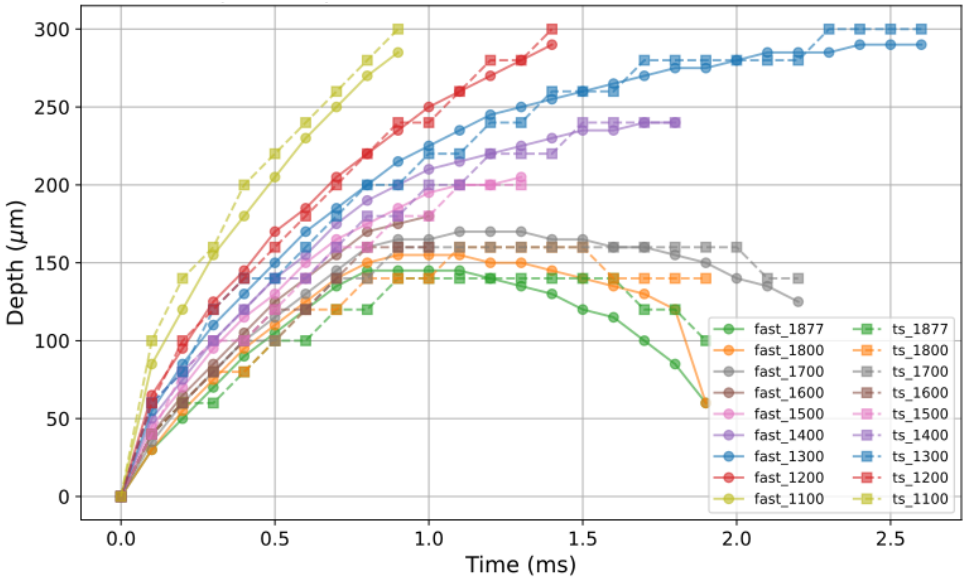


(b) Width comparison after optimisation for all isotherms.

Figure 4.6: Width comparison between FastLab (solid) and AnTem (dotted) before and after optimising all isotherms.



(a) Depth comparison before optimisation on all isotherms.



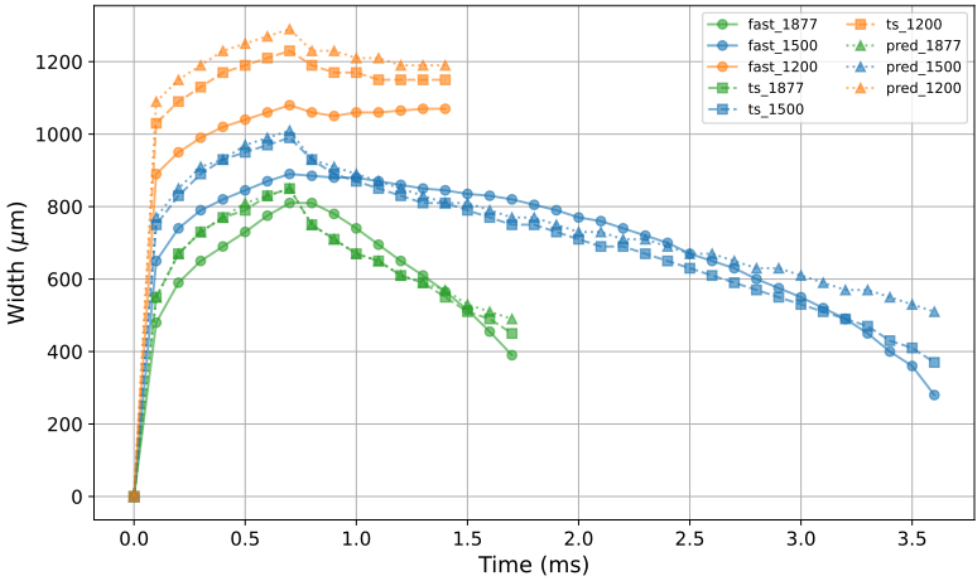
(b) Depth comparison after optimisation on all isotherms.

Figure 4.7: Depth comparison between FastLab (solid) and AnTem (dotted) before and after optimising all isotherms.

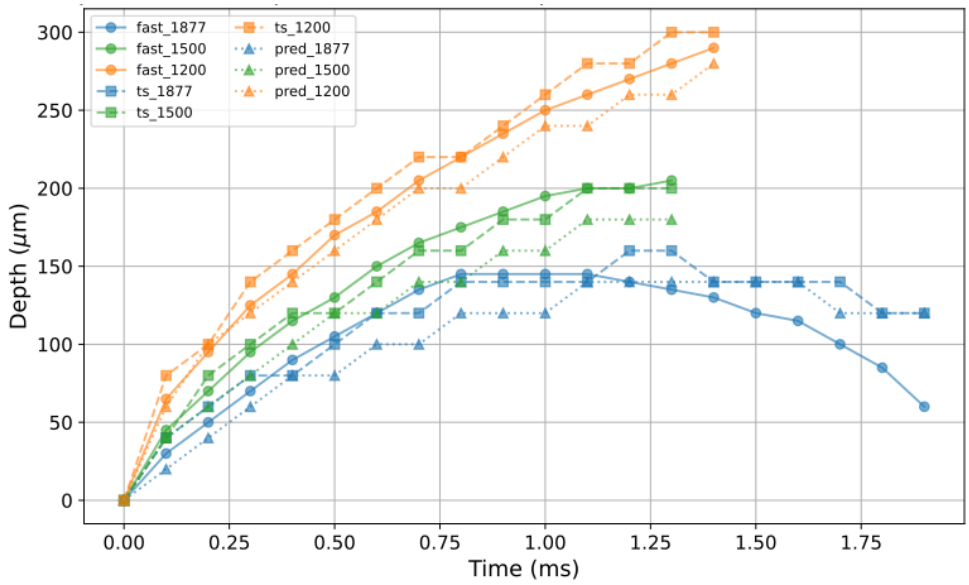
Later, other parameter combinations were tested to ensure that the program could handle more extreme conditions, for example a beam power or background temperature too low for the material to barely melt or not melt at all. During this testing it was noticed that if you had the high and low isotherms, or the borders of the problem, accurately optimised, the intermediate ones tend to self-adjust with next to no decrease in quality for the results. Due to this, it was decided to utilise this behaviour and that the next, larger step, of the project would be conducted with only three isotherms, which were chosen to be 1200, 1500 and 1800 K. The reasoning behind choice of the isotherms was that 1200 K, being the third lowest, had a high chance of being 0 throughout, but also a high chance of some activity, unlike 1100 and 1000 K which were permanently 0 in a lot of the simulations, resulting in a good balance. 1800 K was used as the 0.7 problem was prominent for it, but not as prominent as for 1877 K. It was also close enough to the solidification temperature to take those characteristics into consideration. Finally, 1500 K was a somewhat arbitrary choice of an isotherm in the middle, only to ensure that nothing went wrong in-between the edges. Being able to reduce the number of isotherms from 10 to 3 with little to no loss in result quality sped up the process considerably. It was also noticed that the program could optimise against the lower resolution data about as well as the high resolution. Even if the lower resolution is believed to be further from reality. The optimisation was also run on it to see if the increased amount of data points could make up for this flaw. A combination of the datasets was also tested, where instead of truncating the high resolution data, it was instead continued by the low resolution data, an implementation that showed promise for the future when more data is available.

4.3 Transfer function

Finally when the program was deemed to produce satisfactory results for various parameter combinations, it was run on 180 datasets, all having different combinations of parameters in order to produce enough data for an as accurate transfer function as possible. Among these sets, the power ranged from 90 to 960 W, the background temperature ranged from 20 to 1000 K, the beam on time in the range of 0.05 to 20 ms and the focal width in the range 300 to 900 mm. For each set, the initial values and its final scale parameters were saved. The exact transfer function can not be disclosed due to company policy, but what can be shown are the final scale parameter values for individual datasets and what the predicted AnTem simulation looks like in comparison to FastLab. For validation, datasets not included in these 180 were inserted into the transfer function, one of them being $[P, t_d, \sigma, T_0] = [360, 0.7, 750, 1073]$, the same dataset used to test the program on an individual dataset. This resulted in the predicted values $[P_{scale}, \rho_{scale}, \sigma_{scale}, T_{scale}, \lambda_{scale}] = [0.721, 0.999, 1.035, 1.206, 1.064]$ with a final cost of 0.588. A comparison between FastLab (solid line and circles), the results of optimising AnTem (solid line and squares) and the values produced with the transfer function (dotted line and triangles) can be seen in Figure 4.8, where 4.8a depicts the width and 4.8b the depth. Overall, the simulations agree well, with the exception of the width for the isotherm 1200 K. This decrease in quality likely stems from the regression used for the transfer function having to find some middle ground for all datasets, leading to slightly worse results in some specific cases.



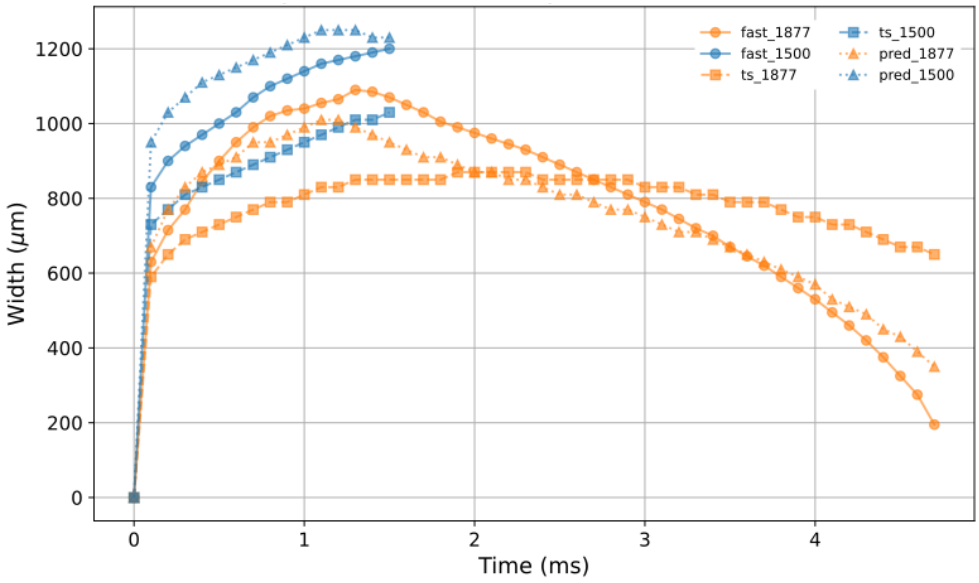
(a) Width comparison between FastLab (solid and circles), AnTem after optimisation (solid and squares) and AnTem using predicted values from the transfer function (dotted line with triangles). In this scenario isotherms $1877, 1500$ and $1200\,\mathrm{K}$ was used.



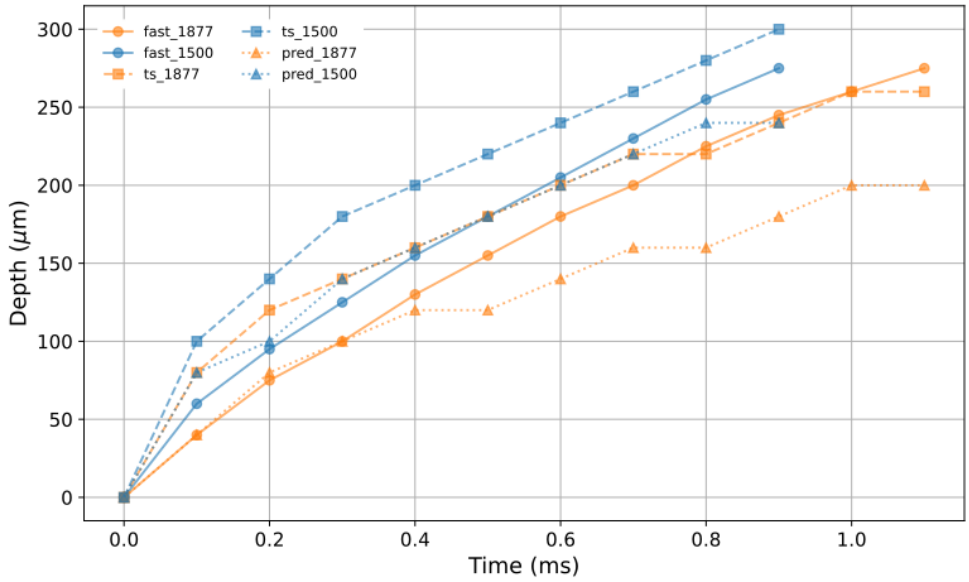
(b) Depth comparison between FastLab (solid and circles), AnTem after optimisation (solid and squares) and AnTem using predicted values from the transfer function (dotted line with triangles). In this scenario isotherms $1877, 1500$ and $1200\,\mathrm{K}$ was used.

Figure 4.8: Figures showing the comparison between FastLab, AnTem after optimising on this specific dataset and AnTem using the predicted scale parameters derived from the transfer function for the parameters $[P, t_d, \sigma, T_0] = [360, 0.7, 750, 1073]$. The prediction works well for the width but slightly worse for the depth. This decrease in quality is likely due to the regression process, which, in finding a single expression for all datasets, causes some to become less accurate in order to find balance and produce acceptable results for all isotherms.

Another of these sets of parameters used to test the transfer function was $[P, t_d, \sigma, T_0] = [480, 1.25, 750, 1273]$. The comparison between FastLab (solid line with circles), optimised values in AnTem (solid line with squares), and predicted values from the transfer function (dotted line with triangles) can be seen in Figure 4.9. This resulted in predicted scale parameter values of $[P_{scale}, \rho_{scale}, \sigma_{scale}, T_{scale}, \lambda_{scale}] = [0.743, 0.999, 1.039, 1.194, 1.162]$ with a cost of 0.517. For comparison, the values of the scale parameters calculated through the optimisation process were $[P_{scale}, \rho_{scale}, \sigma_{scale}, T_{scale}, \lambda_{scale}] = [0.994, 0.999, 1.017, 0.734, 1.318]$ with a cost of 0.545. In this case, the overall performance of the predicted values was better than the optimised values, showing that there is room for improvement in the optimisation. There is also a rather large difference in the scale parameters, especially for the P and T_0, showing how vastly different parameter combinations can yield similar results. As seen in Figure 4.9a, the predicted values are slightly too large for the width of the $1500\,\mathrm{K}$ isotherm and a bit too small for the depth of the $1877\,\mathrm{K}$ isotherm as seen in Figure 4.9b, which might indicate that it settled for a middle ground as close as possible to both. An important detail to note is how the background temperature is greater than $1200\,\mathrm{K}$, which causes the $1200\,\mathrm{K}$ isotherm to be permanently 0 because the temperature cannot reach a sufficiently low value, a scenario that was not too uncommon for the lower isotherms.



(a) Width comparison between FastLab (solid and circles), AnTem after optimisation (solid and squares) and AnTem using predicted values from the transfer function (dotted line with triangles). In this scenario isotherms $1877, 1500$ and $1200\,\mathrm{K}$ was used.



(b) Depth comparison between FastLab (solid and circles), AnTem after optimisation (solid and squares) and AnTem using predicted values from the transfer function (dotted line with triangles). In this scenario isotherms $1877, 1500$ and $1200\,\mathrm{K}$ was used.

Figure 4.9: Figures showing the comparison between FastLab, AnTem after optimising on this specific dataset and AnTem using the predicted scale parameters derived from the transfer function for the parameters $[P, t_d, \sigma, T_0] = [480, 1.25, 750, 1273]$. The prediction works well for the width but slightly worse for the depth. This decrease in quality is likely due to the regression process, which, in finding a single expression for all datasets, causes some to become less accurate in order to find balance and produce acceptable results for all isotherms.

5

Societal, ethical, and ecological aspect

As this work only involves simulations there is no direct impact of the work on either society nor the climate. The only exception being the energy consumed by the computers used to run the simulations. Likewise, the ethical considerations are minimal from a software development standpoint. Instead, what needs to be considered is the impact of AM, especially as it becomes more accessible due to improved simulation tools.
One of the major perks of AM is the minimal material waste. Since the product is built layer by layer, with minimal or no support structure, the material waste is next to none compared to other techniques [23]. This consequently also removes the need to create moulds used when casting and other material and time consuming intermediate steps. AM also allows for the possibility of restoring older products, reducing the need for new production, further reducing material usage [23]. On top of this, the material used for support structures can be reused, a positive aspect due to the metallic powder used in EBM requiring a larger amount of energy and argon in production [23]. Despite this, the pros outweigh the cons from an environmental viewpoint, making EBM one of the better alternatives in the field in this aspect [23]. Further, the environmental effect can then be reduced even more by accurate simulations, reducing the need of practical tests and consequently the use of materials and energy.
The implications of the versatility of AM does however reach beyond the factory. Partly because of the possibilities of titanium implants in biomedical engineering. But also the risks in aerospace [2]. With a larger subsection of the aerospace business being military, the risks are non-negligible since it could further the development of weaponry and other military technologies, which causes it to potentially have a rather large ethical impact. On the other hand, with its aforementioned possibilities in the medical field [2], thanks to titanium being one of the more common materials used for techniques like EBM, there are also clear positive opportunities in society.
It is also worth mentioning the high entry cost and operation cost that comes with EBM machines, causing this type of AM to be somewhat exclusive to corporations and institutions [2]. Instead, making the technology more cost efficient and available to the public could possibly reduce production costs for companies and reduce the need for them to rely on suppliers for products.

6
Discussion

This chapter discusses major aspects of the project, including challenges, results, and suggestions for future improvements.

6.1 Cost function

The cost function went through many versions, for all versions see Appendix B. The thought process behind the final function was as follows: normalising the difference against their average gave the width and depth a more even impact on the cost instead of using something like the maximum width or depth, which was noticed to be more imbalanced due to the size difference between points. The fine tuning of the balance could instead be done with weights, as the general shape of the curves was typically consistent. Additionally, a power of four was used instead of the more common power of two to put more weight on larger errors. Reducing the frequency and the size of the larger errors and solving the main issue with the power two, which was that it had a tendency to yield very good results for either width or depth while neglecting the other. A solution to this problem could have been to use different cost functions for width and depth and instead optimise them individually, this would however likely require you to perform the entire optimisation twice. Once only looking at width, and once only looking at depth, computing a transfer function for each. While this may yield better individual costs, it raises questions about the validity of results when the other dimension is essentially ignored during each optimisation. It is an approach and an analysis that is worth looking into in future development as an attempt to further improve the results or if only one dimension is of interest at a time. In this work however, it was deemed to be of greater interest to have both dimensions agree at the same time, even if it came at a cost. One could also argue for not minding having one dimension neglected at a time due to the whole calculation being treated as a black box. Additionally, some might want to take the fourth root after raising the expressions to the power of four to have a value more akin to the actual values and more correct in terms of dimensionality. But, as this is, to some extent, a matter of preference, it was seen as an extra unnecessary computation. Moreover, it was decided against this, as allowing larger cost variations made it easier to interpret differences.
Then there is also the major detail that the data in this project was exclusively for point melting a single point. Naturally, this means that future improvements and development that include using several points or line melts, might need altering of major parts of the program to include these new points and their significance. An approach to this was seen in (3.1), which laid the foundation of the one used in this project where the length of the line melt was taken into account. An idea for a potential future implementation similar to

it is to combine the approach of this thesis with that of Stump et al. [11] and interpret the length as a third spatial dimension alongside width and depth in (3.2).

6.2 Dataset

The reasoning behind the format of the data was that the wide spread of isotherms and parameters ensured the relevancy for Colibrium. Further, the table-like structure made it easy and intuitive to handle, simplifying the process of incorporating different dimensions, time steps and isotherms. Allowing for a larger part of the AM process to be considered. Unfortunately, the data did contain errors, such as the 0.7 problem discussed further in Section 6.2.1, but this was worked around as much as possible. There is also the possibility of generating more data to optimise on in the future, both high-and low-resolution to compensate for this. Future improvements are consequently accessible as one can simply rerun the calculation of the transfer function with both the new and existing data.

6.2.1 0.7 problem discussion and example

Naturally as the nature of the 0.7 problem could cause larger portions of datasets to become faulty or needing truncation, it reduced the quality of the data and the results. But the data did agree well with reality up to the point of this problem. The predictable nature of the problem also made it easy to remove affected data points. Additionally, because it primarily affects datasets with higher energy, in most cases most of the important parts of the process (melting and the start of the cooling down of the material) had already occurred. Only in some rare cases did most or all of the data have to be removed. Regarding the background to the problem, since doing calculations like the ones in this thesis had not been on the agenda up until now, the problems it caused had not been a complication until recently. The owners of the source code were notified of the problems that their implementation could cause. Unfortunately, as they belonged to a different branch of the company, it took them until late May 2025 to provide an update to Colibrium Additive. And, as it would take months to generate new data free from faults, no patch to the problem has been implemented as of today. But the possibility for improvement is available and easily accessible in time as simply rerunning the program with the updated data most likely would lead to clear improvements of the results with little need to make changes to the program. Making it perhaps the first, or at least the simplest area, to look to for future improvements.
A concrete example of this problematic behaviour can be seen in Figure 6.1. The problem is perhaps most prominent for the $1300\,\mathrm{K}$ isotherm, since in this case $0.7 \cdot T_S = 1306 > 1300$, causing the isotherm to get stuck in the interface between what has never melted and that which has been resolidified. Leading to the isotherm levelling out without the possibility of decreasing in size due to its constant temperature. Consequently, all isotherms that encountered this problem had to be truncated at the point at which it occurred, causing a decrease in the amount of usable data. For isotherms larger than this value, such as $1400\,\mathrm{K}$, they quickly go to 0 as the temperature is fixed at a value smaller than what the isotherm represents. This naturally also required truncation of the data. A similar problem that was treated by the same error handling, but that was not caused by

an error but by natural causes was when the energy in the system was too low for any isotherms to form properly, resulting in the width and depth remaining at 0 for most of the time. As this meant that the program only would have a handful of points to train on, letting the program train on, for example, a series of two points was deemed to be more of an error source than reliable information, as rarely much can be said from that little information.

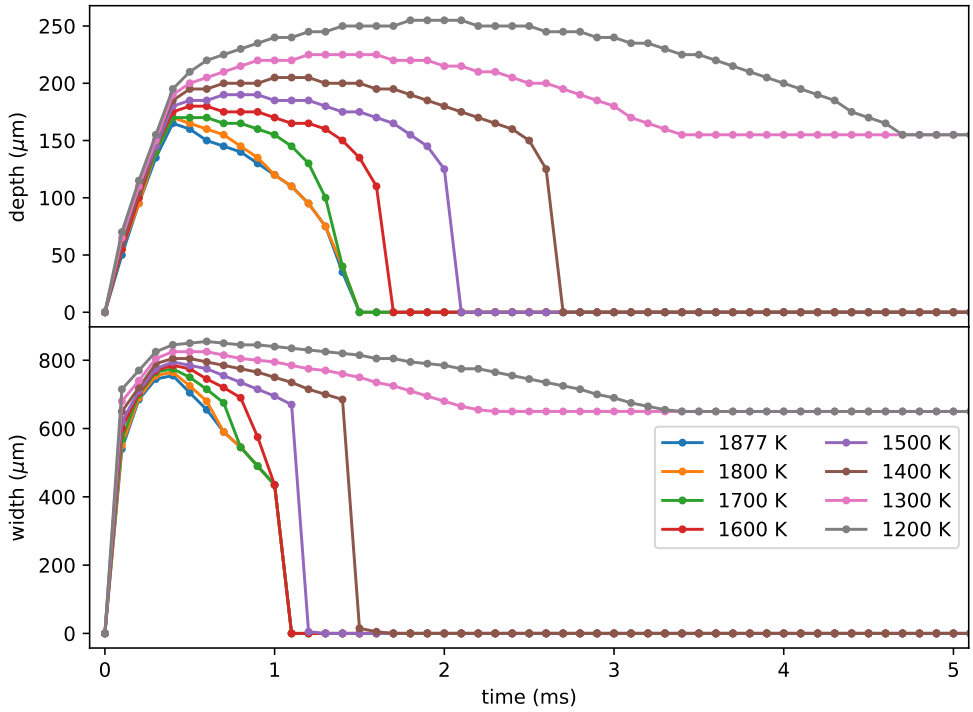


Figure 6.1: FastLab output affected by the 0.7 problem. The problem is perhaps most prominent for the $1300\,\mathrm{K}$ isotherm. In this case $0.7 \cdot T_S = 1306 > 1300$, quickly causing the isotherm to get stuck with no chance of shrinking. For isotherms over this limit, such as $1400\,\mathrm{K}$ the graph instead quickly falls to 0 as the new fixed value is less than what the isotherm represents.

6.3 Optimisation algorithm

For the optimisation algorithm, the Python library pySOT [19] was used and several combinations of settings for sampling, surrogate models, and strategies were tested. Furthermore, the more common NLopt [16] and scikit-learn [17] was tested, but without notable success. Even though the cost function at the time at which these other methods were relevant did not have all the normalisations that the final function had, they often had costs in the hundreds or even the thousands after hundreds of iterations. For comparison, pySOT often had a cost less than 10, rarely going above 30, with these cost functions after tens of

iterations. Due to this clear superior performance, pySOT was chosen. The reason why these other toolboxes underperformed to such an extent is uncertain, but it is believed to be because they got stuck in spurious minima [18].
For pySOT, using a latin hyper cube sampling process with starting points equal to 6 times the number of parameters was concluded to yield good results in reasonable amounts of time. As pySOT has a handful of predefined surrogate models [19], all of them were tested. In the end, the Multi-Adaptive Regression Splines Interpolant (MARS Interpolant) was deemed to work best for this problem, but other settings also showed promise. The other noteworthy setting was the search strategy, available were methods of Stochastic radial basis functions (SRBF), dynamic coordinate search (DYCORS), expected improvement (EI), lower confidence bound (LCB) and random search (RS) [19]. For shorter optimisations, around 50 iterations, EI often outperformed the rest, but with longer, higher resolution runs, SRBF and DYCORS performed noticeably better. The latter was slightly better than the former, which is why DYCORS was selected as the search strategy. LCB and RS typically did not perform very well and were quickly discarded as a viable option. A reason why the chosen combination performed the best and why it outperformed NLopt and scikit-learn might be its exploratory nature [19], which allowed it to reach costs in the order of $1e-6$. Figure 6.2 shows how the cost varied over 300 iterations. It started with a relatively high cost, taking more ambitious steps until it found a ”good” solution. It then searched around the area for a while before making a larger change to ensure that it is not stuck in a local minima, eventually finding an even better solution, repeating this process and always keeping track of the best solution found, which it could return to if no better solution was found. This search strategy was further illustrated by only optimising three parameters and tracking their change. The parameters were power P, background temperature T_0, and the focal width σ, the choice was arbitrary just to illustrate the method as seen in Figure 6.3. As seen, the algorithm searches around an area that has shown to yield a low cost, occasionally taking larger steps to ensure it is not a local minima, but later returning. This particular search used only 50 iterations to maintain readability, but the general behaviour from Figure 6.2 is also visible in Figure 6.3.

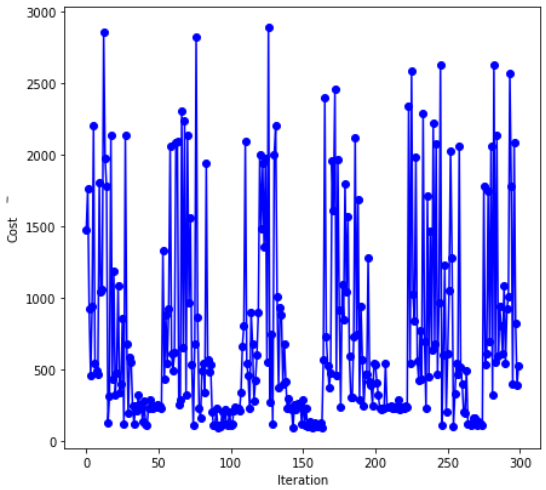


Figure 6.2: Cost over time for pySOT, visualising how it explores the landscape to find better and better solutions over 300 iterations.

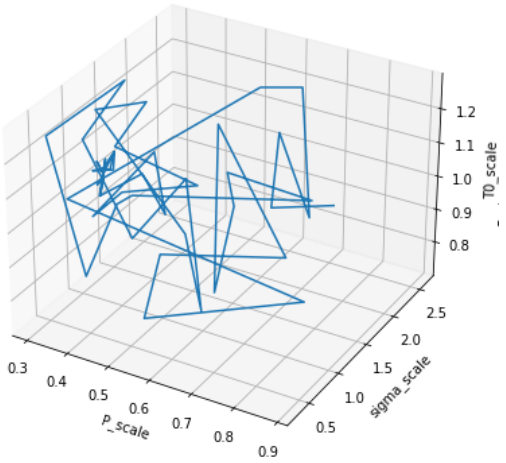


Figure 6.3: Figure showing how pySOT moves throughout the room to find better solutions for 50 iterations, occasionally returning to areas around a previously found better solution.

An aspect that could result in improvement is an increase to the number of iterations. This was tested but the improvement was marginal for the trade-off in time. With pySOT being used in the optimisation a good value will likely be found relatively fast, but an exact result might take some time [19] or never be found due to the nature of the problem. This is largely due to how the algorithm works. It uses a surrogate model to essentially approximate the objective function with a sort of higher dimensional landscape. It then selects a number of pseudo-random points to start its optimisation from, evaluating them against the surrogate model to get an indication of their effect on the actual cost function, to then use promising points in the cost function. This approach is great for this work since it is highly efficient for complex functions based on several correlated variables [19] and thus can search much faster in a larger range of values. But it is also a reason that the search for better values might stagnate, it is to some extent random [19]. Based on this, one could argue that leaving the computer to run for a very long time probably could improve on this problem, which it might, and is not an unreasonable idea, especially since you only have to produce the final results once for each dataset to calculate the final transfer function. This was even discussed as a future improvement by Colibrium, letting the program optimise over the course of months in hope of improving the results. This approach would most likely also not use all ten isotherms, but fewer as was done in this thesis, but the end results would likely improve due to more time to optimise most likely leads to the program finding ever so slightly better values every now and then. Moreover, due to the time it would take, it would likely be relevant when the 0.7 problem is solved to further increase the quality of the results.

6.4 Optimisation process

In the initial optimising of a single isotherm the cost improved significantly. There is still room for improvement but it is limited by the resolution in AnTem. This is the most

evident for the depth in Figure 4.3b where we can see how it tries to follow FastLab but fails due to FastLab having data points in values AnTem cannot reach. This was not seen as a major problem as AnTem is based on simpler physics and an exact match was never expected. Instead, having the melt pool geometry follow FastLab ”closely” was deemed sufficient since the behaviour and final result of the simulations were comparable. Even if you do not get exactly correct temperatures and melt pool geometries, the behaviour and end result were more than comparable. Another observation worth mentioning is how improving one isotherm can worsen another. This is especially clear for the width when optimising three isotherms in Figures 4.4a and 4.4b. This was a constant trade-off, when improving one thing, something else might suffer. However, with the somewhat drastic improvement of the $1500\,\mathrm{K}$ isotherm, the small increase in error for the $1200\,\mathrm{K}$ is acceptable. The reason behind this is likely the relation between spot size and power. A smaller spot concentrates the energy to cause a deeper, narrower melt pool, while a wider spot would result in a wider, more shallow melt pool. But the power could not account for the dynamic changes needed to match FastLab. Thus, a middle ground had to be found, and as the relation varied between isotherms and the datasets, correctly portraying all of them was not always possible and some had to be neglected to some extent at times.
For the case with all ten isotherms, the changes are less noticeable. During testing, it seemed to stabilise rather quickly and required somewhat drastic changes in the system for it to change. This is likely due to the system self-stabilising in some sense with the wide spread of isotherms. Additionally, even if the isotherms have room for improvement, the aforementioned difference between the physics that lay the basis for the models has to be taken into account even more at this stage. Optimising ten isotherms at once by varying five parameters in a system with no, or perhaps several, solutions, will force you to make compromises. Despite all this, the program still managed to fit the isotherms reasonably well.

6.4.1 AnTem errors

A bug that was encountered related to AnTem and most likely had an effect on the results was that a small enough step size would cause the width to shift downward, an example can be seen in Figure 6.4. Similar problems have existed in earlier versions of AnTem, hence the belief that the cause lies with the source code. Attempts to counteract and even patch the problem were made, but to no avail. The consequence of this is that the resolution of the melt pool width came in steps of $20\,\mu\mathrm{m}$, which at times made it impossible for AnTem to align with FastLab. This bug was verified to not be optimisation related by letting it run for a few hours. No matter for how long or short duration, the problem persisted, but it vanished as soon as the step size was increased, and thus $20\,\mu\mathrm{m}$ was deemed to be the lower limit to avoid the problem. For comparison, FastLab had an accuracy of $5\,\mu\mathrm{m}$. As such, patching this problem will likely allow for a significant improvement of the results as one would be able to have an accuracy on-par with FastLab.

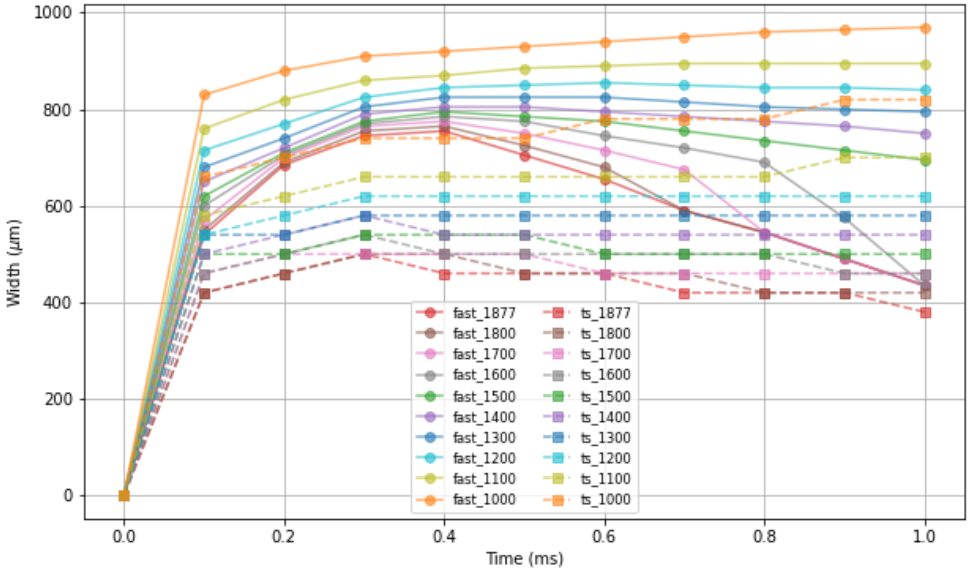
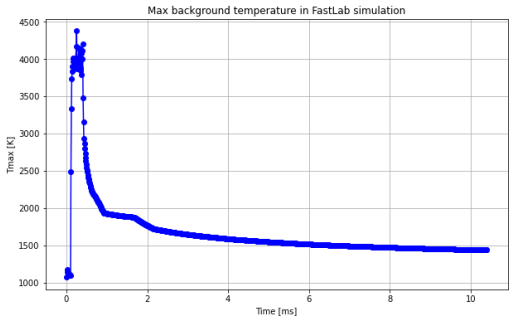
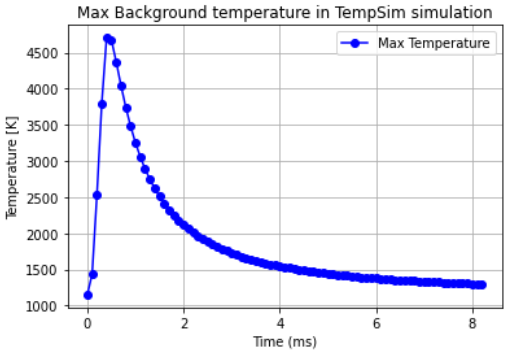


Figure 6.4: Illustration of the problem of a too small step size causes the isotherms to shift downwards by a few hundreds of μm, thought to be caused by a bug in AnTem as similar problems having existed earlier.

Additionally, there was a suspicion of an AnTem related problem due to a slight mismatch in the solidification process when there is no external energy, only the energy in the material left over from the electron beam as well as the background temperature. The belief was that it could be because AnTem did not handle the cooling process correctly, to investigate it, the background temperature was tracked for both FastLab and AnTem simulation with standard values and then plotted against time as seen in Figure 6.5. Figure 6.5a shows the maximum background temperature in the FastLab simulation for each time step, which quickly peaks at around $4500\,\mathrm{K}$ and quickly levels out at around $1500\,\mathrm{K}$ after about $5\,\mathrm{ms}$. Figure 6.5b shows the corresponding test for AnTem and we can see the same behaviour but with a slightly higher maximum temperature and a slightly faster cooling down, and they only differ slightly, which is a positive indication. The small plateau at about $1.8\,\mathrm{ms}$ in 6.5a is the latent heat radiating away. It is primarily after this point the graphs should behave similarly, which they do. The behaviour is overall similar and the values are fairly close, indicating that both calculate the temperature correctly. The errors are hence likely not due to this suspicion but something else, and thus is an area of improvement for the future.



(a) The maximum background temperature in the system during a FastLab simulation and how it evolves with time.



(b) The maximum background temperature in the system during a AnTem simulation and how it evolves with time.

Figure 6.5: Comparison between FastLab and AnTem and how the maximum background temperature in each system changes over time, investigated in belief that there was a bug in AnTem, but as seen in the figure, AnTem looks to perform the calculations correctly.

6.5 Derivation of the transfer function

The values for the number of iterations, weights, form of cost function, etc. were all based on optimising a single parameter set at a time. However, they were tested on several different sets to ensure their flexibility. These were values that resulted in a low and stable cost, where the algorithm had found a close fit and a slight decrease in cost no longer was worth the trade-off of significantly increased time. With an approach that included changing the initial scale parameter values of P and σ to a number of different values instead of always starting the optimisation at value 1, the total number of iterations for each dataset was 360. This exploratory search was especially helpful in this step of optimising different datasets, as you would artificially take these larger steps when needed and rather quickly find a good solution.

As this search for a transfer function, the main task was to repeat the initial step of optimising a single dataset, but several times over, it naturally suffered from the same problems, such as the 0.7 problem. As such, some of the sets used had fewer data points than one could have hoped for. An attempt to counteract this was made with sheer numbers of sets used. Even if the results were satisfactory, increasing the amount of datasets and data points more will likely improve the results further. Letting the transfer function predict values for these parameter combinations with fewer data points did at times pose a problem and the fit could be subpar. However, as a solution to the problem is nearing, when this program is to be implemented into the companies work, it will likely have been trained on new data that does not suffer from a lack of data points, hopefully making the predictions better for all parameters. In waiting for the 0.7 problem to be solved, a promising next step looks to be combining the high-and low-resolution data. When testing, the increase in data points for each isotherm looked to help the optimisation better fit the curves, but as only a handful of these datasets were usable, no real conclusions

could be drawn. Thus, a strong recommendation is made to test this in the future when time allows for it. Regardless if only using high-resolution data or a combining it with low-resolution data, increasing the amount of data points is suspected to lead to some new problems that likely will need to be considered. Mainly, when increasing the length of the ”tail” of the graph, it could cause the function to put too much emphasis on these values and less on the rise and peak of the curve, which is a major part of interest. Thus, including something like extra weights to help the function fit the peak of the curves in both time and size might be of extra interest at that point.
Other modifications to be tried in the future could be to change the allowed range for the scale parameters. Looking at the earlier work by Stump et al. [11], more specifically Table 3 in [11], one can see that he allows for a calibrated power of over $100\,\%$. In contrast, in this thesis it was limited to a maximum of $100\,\%$ as having more power than inserted into the system was deemed unreasonable. This was supported by physical observations pointing to tens of percent of the energy being lost to reflection, evaporation or radiation, indicating that a reasonable value should be well below $100\,\%$. One could only speculate as to why they took the approach they did, perhaps to compensate for the lack of physics as was done with the background temperature in this project, or maybe other parameters that were not mentioned were altered to compensate for this increase. Nevertheless, it might be worth looking into this approach as even if having a power higher than your input power might sound unreasonable, the calculations are in some regard treated like a black box, making the argument for unreasonability less compelling.
Finally, there is a need to mention the decision by Colibrium Additive to not publish the complete transfer function. The main reason behind the decision is that competing companies have similar programs to AnTem but not programs comparable to FastLab. At least not that they know of. As with these results, what is required to produce results comparable to FastLab is the transfer function and a program that uses the same approach as AnTem. This means that all that is allowed to be shown to the public eye is the final product that the transfer function generates, as it was concluded that competitors would not be able to retrieve the original expression from a handful of scalar values. This might change in the future, but that is a decision to be made by Colibrium Additive as the results are considered their IP.

7
Conclusions

Although the simulated results using the derived transfer function did not agree exactly with FastLab, it is concluded by Colibrium Additive that they are satisfactory for their purpose. Therefore, a digital twin of a known multi-physics model based on an analytical thermal model is deemed derived. And with it, the main goal of the thesis is considered to have been achieved, as it is now possible to generate results comparable to that of FastLab in a fraction of the time using AnTem. The results show potential for future improvements, but with the limitations that were the faulty and problematic data, where to start with these improvements is clear. The belief of the author and the company is that rerunning the optimisation with new and improved data will produce even more accurate results. And hopefully, the future can reduce the need for running FastLab for these types of calculations even more, saving both time and money. The results show that a mapping between the models is possible, and that Colibrium can start using the results of this thesis in their work. They will of course take the improvements and further development into consideration for future projects as they can use the current version as a foundation. Other areas of improvement could, of course, arise later, like improving on the cost function, or deriving a completely new one. Especially if more data points are included in the future. Bug fixing AnTem, allowing for a higher resolution, could also be an area of improvement as having an accuracy on par with or even better than FastLab would allow for even more accurate simulations.

7.1 Summary

In summary, a mapping in the form of a transfer function has been derived between two physical models used in the process of simulating additive manufacturing, more specifically, layer-wise powder based fusion using electron beam melting. For the first model, FastLab was used. It is based on a Lattice-Boltzmann model and is highly accurate, capable of producing reliable simulations that agree well with experimental results. Thus, simulations from it could be treated as ground truth data. It is however time consuming, as milliseconds of simulation can take hours to generate. For the second model, AnTem was used. It takes a simpler approach as it is in its essence the analytical solution to the heat equation. This simplicity makes it much faster than FastLab, but at the cost of accuracy. However, because the physics of the two models agree at the edge of melt pools, an attempt was made to change the material-and-process parameters in AnTem until the melt pool geometry agreed well with FastLab. This was then done for 180 parameter combinations, allowing a regression to be made. Deriving a transfer function that could take the parameters that were supposed to be used in FastLab, instead to be altered and used

in AnTem. Although the results have room for improvements, it is deemed to be proven that a mapping between the models is possible, demonstrating the feasibility of using a low-resolution analytical model with modified material-and-process parameters to create a digital twin of FastLab capable of producing results on par with it in the same computational time frame as AnTem. Consequently, demonstrating the possibility to bridge the gap between computational efficiency and model accuracy, establishing a foundation for more cost-effective and scalable simulations in additive manufacturing.

Bibliography

[1] A. Zakirov, S. Belousov, M. Bogdanova, *et al.*, "Predictive modeling of laser and electron beam powder bed fusion additive manufacturing of metals at the mesoscale," *Additive Manufacturing*, vol. 35, p. 101 236, 2020, ISSN: 2214-8604. DOI: https://doi.org/10.1016/j.addma.2020.101236. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214860420306084.

[2] T. DebRoy, H. Wei, J. Zuback, *et al.*, "Additive manufacturing of metallic components – process, structure and properties," *Progress in Materials Science*, vol. 92, pp. 112–224, 2018, ISSN: 0079-6425. DOI: https://doi.org/10.1016/j.pmatsci.2017.10.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079642517301172.

[3] M. Li, W. Du, A. Elwany, Z. Pei, and C. Ma, "Metal binder jetting additive manufacturing: A literature review," *Journal of Manufacturing Science and Engineering*, vol. 142, no. 9, Jun. 2020, ISSN: 1528-8935. DOI: 10.1115/1.4047430. [Online]. Available: http://dx.doi.org/10.1115/1.4047430.

[4] H. Lee, C. H. J. Lim, M. J. Low, N. Tham, V. M. Murukeshan, and Y.-J. Kim, "Lasers in additive manufacturing: A review," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 4, no. 3, pp. 307–322, Jul. 2017, ISSN: 2198-0810. DOI: 10.1007/s40684-017-0037-7. [Online]. Available: http://dx.doi.org/10.1007/s40684-017-0037-7.

[5] C. Körner, "Additive manufacturing of metallic components by selective electron beam melting — a review," en, *Int. Mater. Rev.*, vol. 61, no. 5, pp. 361–377, Jul. 2016.

[6] Y. Wang, S. Roy, H. Choi, and T. Rimon, "Cracking suppression in additive manufacturing of hard-to-weld nickel-based superalloy through layer-wise ultrasonic impact peening," *Journal of Manufacturing Processes*, vol. 80, pp. 320–327, Aug. 2022, ISSN: 1526-6125. DOI: 10.1016/j.jmapro.2022.05.041. [Online]. Available: http://dx.doi.org/10.1016/j.jmapro.2022.05.041.

[7] T. Childerhouse and M. Jackson, "Near net shape manufacture of titanium alloy components from powder and wire: A review of state-of-the-art process routes," *Metals*, vol. 9, no. 6, p. 689, Jun. 2019, ISSN: 2075-4701. DOI: 10.3390/met9060689. [Online]. Available: http://dx.doi.org/10.3390/met9060689.

[8] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, *Additive Manufacturing Technologies*. Springer International Publishing, 2021, ISBN: 9783030561277. DOI: 10.1007/978-3-030-56127-7. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-56127-7.

[9] C. Körner, M. Markl, and J. A. Koepf, "Modeling and simulation of microstructure evolution for additive manufacturing of metals: A critical review," en, *Metall. Mater. Trans. A*, vol. 51, no. 10, pp. 4970–4983, Oct. 2020.

[10] G. Vastola, G. Zhang, Q. X. Pei, and Y.-W. Zhang, "Modeling the microstructure evolution during additive manufacturing of ti6al4v: A comparison between electron beam melting and selective laser melting," *JOM*, vol. 68, no. 5, pp. 1370–1375, Apr. 2016, ISSN: 1543-1851. DOI: 10.1007/s11837-016-1890-5. [Online]. Available: http://dx.doi.org/10.1007/s11837-016-1890-5.

[11] B. Stump, A. Plotkowski, and J. Coleman, "Solidification dynamics in metal additive manufacturing: Analysis of model assumptions *," *Modelling and Simulation in Materials Science and Engineering*, vol. 29, no. 3, p. 035 001, Feb. 2021, ISSN: 1361-651X. DOI: 10.1088/1361-651x/abca19. [Online]. Available: http://dx.doi.org/10.1088/1361-651X/abca19.

[12] J. B. Malmberg and M. Wallenås, "Solving the heat equation in connection with electron beam melting," 2012. [Online]. Available: https://publications.lib.chalmers.se/records/fulltext/159984.pdf.

[13] R. Forslund, A. Snis, and S. Larsson, "Analytical solution for heat conduction due to a moving gaussian heat flux with piecewise constant parameters," 2018. DOI: 10.48550/ARXIV.1803.10668. [Online]. Available: https://arxiv.org/abs/1803.10668.

[14] Y. Huang, T. G. Fleming, S. J. Clark, *et al.*, "Keyhole fluctuation and pore formation mechanisms during laser powder bed fusion additive manufacturing," *Nature Communications*, vol. 13, no. 1, Mar. 2022, ISSN: 2041-1723. DOI: 10.1038/s41467-022-28694-x. [Online]. Available: http://dx.doi.org/10.1038/s41467-022-28694-x.

[15] E. Britannica, *Thermal conductivity*, Accessed: 2025-05-13, Apr. 2025. [Online]. Available: https://www.britannica.com/science/thermal-conductivity.

[16] S. G. Johnson, *Nlopt: Nonlinear optimization*, Available at https://nlopt.readthedocs.io/, MIT, 2023.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*, 2014. DOI: 10.48550/ARXIV.1406.2572. [Online]. Available: https://arxiv.org/abs/1406.2572.

[19] D. Eriksson, D. Bindel, and C. A. Shoemaker, "Pysot and poap: An event-driven asynchronous framework for surrogate optimization," *arXiv preprint arXiv:1908.00420*, 2019.

[20] L. Ståhle and S. Wold, "Analysis of variance (anova)," *Chemometrics and Intelligent Laboratory Systems*, vol. 6, no. 4, pp. 259–272, 1989, ISSN: 0169-7439. DOI: https://doi.org/10.1016/0169-7439(89)80095-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0169743989800954.

[21] S. Seabold, J. Perktold, *et al.*, *Statsmodels: Statistical modeling and econometrics in python*, https://www.statsmodels.org/, Accessed: 2025-04-09, 2024.

[22] Minitab, LLC, *Minitab*, version 20.2, Apr. 16, 2021. [Online]. Available: https://www.minitab.com.

[23] V. T. Le and H. Paris, "A life cycle assessment-based approach for evaluating the influence of total build height and batch size on the environmental performance of electron beam melting," *The International Journal of Advanced Manufacturing Technology*, vol. 98, no. 1–4, pp. 275–288, Jun. 2018, ISSN: 1433-3015. DOI: 10.1007/s00170-018-2264-7. [Online]. Available: http://dx.doi.org/10.1007/s00170-018-2264-7.

A

Algorithm

A more detailed description of the code and functions related to the simulation environment and settings can be seen in Algorithm 2

Algorithm 2 Optimisation of Simulation Parameters

```
function GET_FASTLAB_DATA(Initial values)
    Initialise FastLab with given dataset
    Retrieve material parameters
    melt pool geometries = Read_FastLab_Data(Settings)
    Return material parameters and melt pool geometries
end function
function READ_FASTLAB_DATA(Settings)
    Read and process melt pool geometries
    Format data
    Remove any problematic/faulty data
    Return melt pool geometries
end function
function RUN ANTEM(Material and beam parameters)
    Locate relevant AnTem files
    Initialise and configure Calculator
    Set material properties and simulation settings
    Process material properties and settings using Calculator
    run Get_get_AnTem_data(Settings)
    Return simulation results as Data frames containing melt pool geometries
end function
function GET_GET_ANTEM_DATA(Settings)
    Given chosen settings calculate melt pool geometries
    Reformat to have same format as FastLab data
    Drop values affected by faulty data in FastLab
    return data
end function
```

For a more detailed description of the code and functions related to cost function and optimisation, see Algorithm 3

Algorithm 3 Optimisation of Simulation Parameters

function OPTIMISE COST FUNCTION(Isotherms, AnTemData, FastLabData, parameters, max_iterations)
    Define the optimisation problem as minimising the cost function
    Initialise pySOT and declare its settings
    function RUN OPTIMISATION(pySOT settings, FastLabData, AnTemData, paramaters)
        for $i = 1$ to max_iterations **do**
            Update Scale parameters
            Calculate cost according to equation 3.2
            Compare against best cost
        end for
    end function
    Update parameters to those that resulted in best cost
    Run AnTem one final time with these values
    Return results
end function

B

Tested cost functions

A list of tested cost functions with less success than (3.2). The ϵ that can be seen in a few equations is a small constant to avoid dividing by 0. As for (B.1), it only looks at the maximum and minimum value of each isotherm, as well as at what indices they occur and calculates a cost according to how they differs from FastLab. w_t are indices specific weights, or time step specific weights, index$()$ is the index/time step of the maximum and/or minimum value of each isotherm. The initial values are ignored for the minimum as they always are 0 and not of interest. This was also the cost function that performed the best outside of the final one. The performance of the others varied. As most of them are very close in structure to the final version (3.2), they will not be discussed in detail. This small difference in structure naturally often lead to a small difference in results, but as this difference typically was negative, the decision was made to improve on them, which eventually resulted in (3.2).

$$\begin{aligned}\text{cost}_{S,k} += \sum_{t \in \mathcal{T}_k} \Bigg[& w_t \left(|\textbf{index}(\max(q_t)) - \textbf{index}(\max(y_t))|^2 \right. \\ & \left. + |\textbf{index}(\min(q_t)) - \textbf{index}(\min(y_t))|^2 \right) \\ & + w_t \cdot \left(\left| \frac{\max(q_t) - \max(y_t)}{\textbf{mean}(q_t, y_t)} \right|^2 + \left| \frac{\min(q_t) - \min(y_t)}{\textbf{mean}(q_t, y_t)} \right|^2 \right) \Bigg] \end{aligned} \tag{B.1}$$

$$\text{cost}_{S,k} = w_S w_k \left(\frac{\sum_{t \in \mathcal{T}_k} \left((q_{S,k,t} - y_{S,k,t})^4 \right)}{\sum_{t \in \mathcal{T}_k} \left(\left(\frac{q_{S,k,t} + y_{S,k,t}}{2} \right)^4 \right)} \right) \tag{B.2}$$

$$\text{cost}_{S,k} = w_S w_k \left(\left(\sum_{t \in \mathcal{T}_k} \left(\left(\frac{2(q_{S,k,t} - y_{S,k,t})}{q_{S,k,t} + y_{S,k,t} + \epsilon} \right)^4 \right) \right)^{\frac{1}{4}} \right) \tag{B.3}$$

$$\text{cost}_{S,k} = w_S w_k \left(\left(\sum_{t \in \mathcal{T}_k} \left(\left(\frac{2(q_{S,k,t} - y_{S,k,t})}{q_{S,k,t} + y_{S,k,t} + \epsilon} \right)^4 \right) \right)^{\frac{1}{4}} \right) \tag{B.4}$$

$$\text{cost}_{S,k} = w_S w_k \left(\sum_{t \in \mathcal{T}_k} \left(\left(\frac{2(q_{S,k,t} - y_{S,k,t})}{q_{S,k,t} + y_{S,k,t}} \right)^4 \right)^{\frac{1}{4}} \right) \tag{B.5}$$

$$\text{cost}_{S,k} = w_S w_k \left(\frac{\sum_{t \in \mathcal{T}_k} \left((q_{S,k,t} - y_{S,k,t})^4 \right)}{\sum_{t \in \mathcal{T}_k} \left(q_{S,k,t} + y_{S,k,t} \right)} \right) \tag{B.6}$$

$$\text{cost}_{S,k} = w_S \cdot w_k \cdot \left(\frac{\sum_{t \in \mathcal{T}_k} \left((q_{S,k,t} - y_{S,k,t})^4 \right)}{\left(\max(y_{S,k,t}) \right)^4} \right) \tag{B.7}$$

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY