



Master thesis:

Inverse Optimal Control of Nonlinear Transient Heat Equation Assisted by Machine Learning : Application on Welding Process

Author

Youness El Houssaini

*A thesis submitted in fulfillment of the requirements for
the Master of Science in applied mathematics at
Strasbourg University*

Carried out in Fraunhofer Institute for Mechanics of Materials IWM

Thesis period:
February 1, 2021 - August 23, 2021

Supervisors:
Dipl.-Ing. F. Dittmann, Fraunhofer IWM
Dr. I. Varfolomeev, Fraunhofer IWM

Examiner:
Prof. C. Prud'Homme, Strasbourg university

Abstract

Welded parts are used in many different applications where they often are exposed to high loads and their stability is relevant for safety. Proper evaluation of residual stresses can lead to better lifetime estimation, which can avoid failure and/or save resources. During welding processes, fused components undergo differential thermal expansion and contraction that cause residual stresses. Modeling such phenomena in order to reproduce reality is known to be a forward problem. But what if we have an idea of the reality, e.g. measurement data, and we want to reconstruct the causal factors, e.g. material properties or heat source? This is categorized as an inverse problem and it is useful for instance to design new products or in nondestructive testing. The search for an optimal solution governed by a nonlinear transient heat transfer equation is the subject of this thesis. In a study case of a cross-shaped component manufactured by welding, the parameters of the heat source shall be identified exemplary given measured temperature profiles. The problem is approached theoretically at two levels: the lower level which consists of identifying the set of applicable parameters of the heat source given the global volumetric density; the upper level where the optimal control problem has to be solved. The well-posedness of both, forward and inverse problems have been examined. Related assumptions have been proposed to analyze the uniqueness. In order to solve the inverse problem, two main approaches have been proposed. On the one hand, the black-box approach using ABAQUS/standard as a solver of the forward problem and iteratively estimating the steepest descent of a defined cost function. Besides that, a neural network model was trained using the ABAQUS FEA output data to estimate the solution of both, forward and inverse problems. On the other hand, the white box approach is a transparent solver where the partial differential equation is spatially discretized and transformed into a set of ordinary differential equations (ODEs). The resulting time-continuous algebraic equation is obtained based on a series of least-squares regression models using the thermal matrices and loads generated by ABAQUS/Standard. Additionnally, a model order reduction is performed and two main methods are proposed to solve the inverse optimal problem including an approximate gradient based optimization loop and the adjoint state based method. The proposed tools can be also combined in order to reach higher accuracy or faster results.

Foreword

This thesis was written and completed as part of the master's program in UFR Mathematics and Informatics with the specialization of scientific computing and mathematics of information (CSMI) at the University of Strasbourg. I am very grateful that I was given the opportunity to work on this thesis at the Fraunhofer Institute for Mechanics of Materials (IWM) where I could apply my acquired knowledge to a concrete practical problem. The time frame of this work reached from February, 1st until the end of July 2021.

First and foremost I would like to express my special thanks to Prof. C. Prud'Homme who accepted to supervise and examine this thesis and in addition to all professors for the support during this thesis along the whole way towards the master's degree.

My full gratitude also goes to the group manager Dr. Igor Varfolomeev and supervisor Dipl.-Ing. Florian Dittmann for giving me the opportunity to write this thesis at Fraunhofer IWM. I am very grateful for their enduring support and helpful insight and highly appreciate the good working conditions and positive atmosphere in the group.

Furthermore, my thanks go to my fellow students in the CSMI Master for their helpfulness and friendliness and the time we had together. And finally I want to thank everyone who contributed to this work by listening and discussing, proof reading and any kind of support.

Table of contents

1	Introduction	1
1.1	Context of the thesis	1
1.1.1	The research department	1
1.1.2	The research group	2
1.2	Modeling welding processes	3
1.2.1	Welding process	3
1.2.2	Welding heat source	4
1.2.3	Causal factors	7
1.2.4	Case study and framework	10
1.3	Description of the problem	12
1.3.1	Forward problem	13
1.3.2	Inverse problem	15
1.3.3	Objectves and overview	16
1.4	Methodology	17
1.4.1	Agile paradigm	18
1.4.2	Bibliography	20
1.4.3	Thesis roadmap	20

2 Fundamentals and problem formulation	22
2.1 Basics and notations	22
2.2 Problem setting	25
2.2.1 State equation	25
2.2.2 Bilevel optimization problem	27
2.3 ABAQUS/Standard solver	36
2.3.1 Discretized problem	37
2.3.2 Implicit scheme	38
2.3.3 Newton-Raphson method	39
3 Black box approach	43
3.1 Approximate gradient-based method	43
3.1.1 Gradient descent	44
3.1.2 Implementation and results	47
3.2 Multilayer perceptron neural network	56
3.2.1 Data collection and processing	57
3.2.2 Forward problem	58
3.2.3 Inverse problem	59
4 White box approach	64
4.1 ABAQUS simulation model	64
4.1.1 Description	64
4.1.2 Matrices generation	65
4.1.3 Numerical validation	67
4.1.4 Problematic	68
4.2 Parameter fitting	70

4.2.1	Heat source and losses models	70
4.2.2	Least square method	71
4.2.3	Internal loads parametrisation	73
4.2.4	Inertia parametrisation	75
4.3	Inverse problem	76
4.3.1	Approximated Gradient descent	76
4.3.2	Pontryagin Maximum Principle	77
5	Summary and achievements	79
5.1	Summary	79
5.2	Achievements	80
5.3	Comparison	81
6	Outlook	83
A	Repositories	86
A.1	Gantt chart	86
A.2	Source code	86
A.3	Documentation	87
B	Regression models	88
B.1	Convexity hypothesis	88
C	Optimization tool	90
C.1	Numerical experiments	90
D	White box	95
D.1	Conjugate gradient solution	95

D.2 Heat source code	97
D.3 Internal loads	98
Bibliography	103

Acronyms

BEM	Boundary Element Method.
CAD	Computer Aided Design.
CAE	Computer Aided engineering.
CNN	Convolutional Neural Network.
CSMI	Calcul Scientifique et Mathématique de l'Information.
DL	Deep Learning.
FDM	Finite Difference Method.
FE	Finite Element.
FEA	Finite Element Analysis.
FEM	Finite Element Method.
HAZ	Heat Affected Zone.
HPC	High Performance Computing.
ML	Machine Learning.
MLP	Multilayer Preceptron.
MOR	Model Order Reduction.
NN	Neural Network.
PDE	Partial Differential Equation.
PMP	Pontryagin Maximum Principle.

S.t Subject to.

SLURM Simple Linux Utility for Resource Management.

WRS Welding Residual Stresses.

List of Figures

1.1	Illustration of some configurations of the Butt joint types [9].	4
1.2	Illustration of the weldig heat source model.	6
1.3	Illustration of a realistic temperature distribution on a welded pipe [8]. .	7
1.4	Illustration of the heat transfer phenomena considered in the welding process.	10
1.5	Illustrative example from a former project for a welding process simulation model [39].	11
1.6	Illustrative example of the a tested component (right) and its dimension (right) [39].	11
1.7	A screenshot of an exemplary documented method.	13
1.8	A screenshot of the developed Abacal tool documentation (main page). .	18
1.9	A screenshot of an exemplary documented method.	19
1.10	A screenshot of the master thesis Gantt chart.	21
2.1	Illustrative comparison of the simulated temperature for three different controls at different positions on the x axis on the graphs label i.e. y and z coordinates are fixed (-4,10).	31
2.2	Illustrative estimations of the losses at an exemplary positions eventually far from the welding zone.	32
2.3	Illustrative case where the tempererute is affine linear with respect to u.	34

3.1	A schema on the proposed black box based optimisation tool to solve the inverse problem.	43
3.2	An overview of developed parameter identification tool.	48
3.3	Gradient descent method (boxed in green) and the backtracking line search (boxed in blue).	49
3.4	An illustration of the convexity of the cost function with respect to the maximum intensity Q	51
3.5	An illustration of the cost function convexity with respect to the identification of 2D heat source parameters.	52
3.6	An illustration of both measurements positions (left) and numerical target profiles (right).	53
3.7	The result of the gradient descent tool to identify the maximum heat source intensity Q	53
3.8	The result of the gradient descent tool to identify the maximum heat source intensity Q	54
3.9	The result of the gradient descent tool to identify all parameters (part 1 of the results).	54
3.10	The result of the gradient descent tool to identify all parameters (part 2 of the results).	55
3.11	The loss function for the identification of all parameters.	56
3.12	Collected data for training the MLP model.	57
3.13	Proposed MLP architecture.	58
3.14	The loss function during the training phase.	59
3.15	The coefficient of determination (to the left) and the target against the prediction (to the right) in the testing phase.	59
3.16	The coefficient of determination (to the left) and the target against the prediction (to the right) in the testing phase for parameter Q	60
3.17	The target against the prediction in the testing phase for parameters Q , B and C	61

3.18	The coefficient of determination in the testing phase for parameters Q, B and C	62
3.19	The Pearson correlation between all the variable including metrics and parameters Q, B and C	63
4.1	A screenshot of the lines to add (boxed in green) in the ".inp" file to extract thermal matrices.	66
4.2	The linearized model solution compared to ABAQUS solution for frame 1 using conjugate gradient solver.	67
4.3	The linearized model solution compared to ABAQUS solution for frame 1 using conjugate gradient solver.	68
4.4	The linearized model solution compared to ABAQUS solution for frame 5 using Newton's method solver	69
4.5	The linearized model solution compared to ABAQUS solution for frame 5 using Newton's method solver.	69
4.6	The respective heat source and losses weights over Y axis of the component.	72
4.7	The External heat source predicted model (in green) and the generated External load (in blue).	73
4.8	The predicted and linearized internal loads models and the corresponding R^2 evaluation on the study case model.	74
4.9	The predicted and linearized internal loads models and the corresponding R^2 evaluation on a validation case.	74
4.10	The predicted and linearized internal energies models and the corresponding R^2 evaluation on the study case model.	75
4.11	The predicted and linearized internal energies models and the corresponding R^2 evaluation on the validation case at frame 1 (first).	75
4.12	The predicted and linearized internal energies models and the corresponding R^2 evaluation on the validation case at frame 74 (last).	76
4.13	A schema on the proposed white box based optimisation tool to solve the inverse problem.	77

6.1	A screenshot of the generated documentation for ABACAL tool.	83
6.2	A screenshot of the designed GUI dialog windows.	84
6.3	Type of the training data to build the CNN model.	85
B.1	Illustrative case where the tempererature is affine linear with respect to u in the beggining of the heating phase.	88
B.2	Illustrative case where the tempererature is affine linear with respect to u in the end of the heating phase.	89
C.1	The result of the gradient descent tool to identify three parameters.	91
C.2	The cost function for the identification of three parameters.	92
C.3	The result of the gradient descent tool to identify four parameters.	93
C.4	The loss function for the identification of four parameters.	94
D.1	The linearized model solution compared to ABAQUS solution for frame 74 using Conjugate gradient solver.	95
D.2	The linearized model solution compared to ABAQUS solution for frame 74 using Conjugate gradient solver.	96
D.3	Snippet of the heat source code.	97
D.4	The predicted and linearized internal loads of a training model over the moving axis at frame 40 (~ half).	98
D.5	The predicted and linearized internal loads of a validation model over the moving axis at frame 40 (~ half).	99

List of Tables

1.1 Examples of efficiencies in some configurations of Arc welding process.	6
5.1 Illustrative comparison of the four proposed tools for one single parameter identifiction i.e Q.	82

Chapter 1

Introduction

In order to understand clearly the focus of this thesis, which deals with the inverse optimal control of non-linear heat equation, it is interesting to put a special emphasis on the context. First, a bird's-eye view explaining the motivation behind this thesis is given. Then, the domain of application, which is welding process, is defined generally, technically and also with regard to the adopted study case. After introducing the theoretical and practical background of the welding process, the connected problem is formulated and the objectives are set. Finally, the adopted methodology is presented, as well as the project plan with the different packages and the estimated workload.

1.1 Context of the thesis

1.1.1 The research department

The thesis took place in the Fraunhofer Institut for Mechanics of Materials IWM, which is a division of the Fraunhofer Society¹. The focus in Fraunhofer IWM is upon application-oriented research in partnership with both, industry and public institutions, throughout four main business units. Each department has a defined contextual focus; these being Component Safety and Lightweight Construction, Manufacturing Processes, Tribology and Assessment of Materials and Lifetime Concepts.

What about safety? And how can you ensure that designed components hold together under load?

¹German research organisation with 72 institutes spread all over Germany [4].

Those are exemplary daily life questions in the department Component Safety and Lightweight Construction where this Master thesis was written. More specifically at the group for Fatigue and Fracture Mechanics.

1.1.2 The research group

The focus of the Fatigue and Fracture Mechanics team is to combine cyclic plasticity models, numerical welding process simulation and fatigue and fracture mechanics models to determine realistic description of the lifetime of a welded component [41]. Generally, lifetime assessment in a welded joint may be approached in different ways [33]:

- in relation to fatigue
- in relation to welding defects
- in relation to corrosion rate

To perform the first two assessments for instance, the stress spectrum of the welding has to be known in addition to the material properties. In simple cases, an analytical solution or a strain gauge² measurement can assess the welding residual stresses (WRS)³. However, in most cases geometries are complicated and material properties are not well defined. In such circumstances a finite element analysis (FEA) is performed. It helps simulating welding processes and thereby estimate the welding residual stresses. In addition to that, it reduces the need for physical prototypes in the design process or in the failure analysis. This saves money and labor but for this purpose an accurate simulation model is required.

This leads to the following question: how can it be ensured that our finite element (FE) simulation model reproduces reality?

Investigating the feasibility and proposing systematic methods replacing a trial and error approach is my contribution to the Fatigue and Fracture Mechanics group during this thesis.

The FE welding process model was already developed in the group [39] and was slightly modified for this work to match the case study, i.e. boundary conditions and welding heat source. The adopted model of the welding heat source is Goldak model [22] which was applied was applied to circumferential butt-welds in a former project in 2016 [25]. The modifications were based on former models of other joint types, e.g.

²An electrical sensor to measure force or strain.

³They are caused by differential thermal expansion and contraction of the weld metal and parent material.

a butt joint with different geometries. More details about the case study model are available in subsection 1.2.4. It is also to mention that precious support was provided by the computer-aided engineering specialists inside the Fatigue and Fracture Mechanics group.

1.2 Modeling welding processes

The failure assessment of welded components for instance, requires an accurate determination of the welding residual stresses in vicinity of the weld. Generally, the experimental determination of the WRS is not possible or very expensive and the analytical assessment routines of guidelines and technical standards are very conservative. This leads to a need for an accurate finite element analysis (FEA) of the welding process. Therefore, the FEA is split up into two parts, a heat transfer analysis covering the heat input of the welding process and a mechanical analysis calculating the nodal translations and stresses resulting of the thermal expansion and phase transformations. In the first part, the heat transfer analysis, the double ellipsoidal heat source model by Goldak [22] is used to simulate the heat input of the moving heat source. This parametrized heat source model is the focus of this thesis. This section have gives an insight on the application domain, the physical phenomena, a description of both the study case and the FEA simulation environment and measurements.

1.2.1 Welding process

According to Wikipedia, welding is a fabrication process whereby two or more parts are fused together by means of heat, pressure or both forming a join as the parts cool. Welding is usually used on metals and thermoplastics but can also be used on wood. The completed welded joint may be referred to as a weldment. There are dozens of welding categories and we cite for instance, Arc welding, Oxyfuel gas welding, Laser beam welding. For more details about other categories, one can check [2]. The formed joint by welding can be categorized in five types of joints as referred by the American Welding Society [1]: butt, corner, edge, lap, and tee. For more details about those categories, one can check [9]. These types may have various configurations at the joint where actual welding can occur as shown in the figure 1.1.

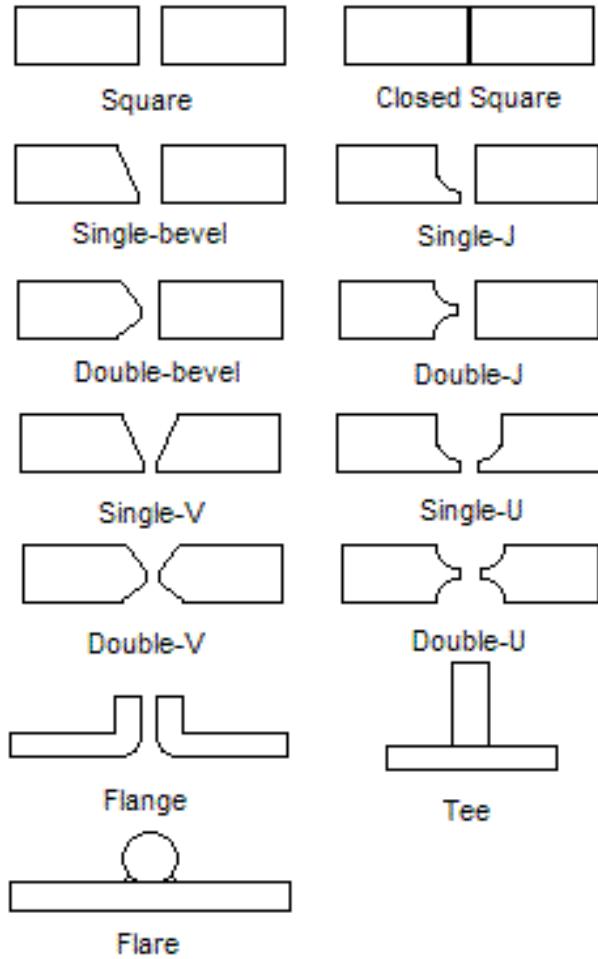


Figure 1.1: Illustration of some configurations of the Butt joint types [9].

1.2.2 Welding heat source

Many different energy sources can be used, including gas flame (chemical welding e.g. diffusion welding⁴) or an electric arc (electrical welding e.g. Arc welding⁵). According to [22], the heat source in the case of electrical heat source for instance can be modeled as a double ellipsoid distribution. It is the most used among all the heat source models and one of the more highly cited in the welding literature [35]. Let \vec{z} be the moving direction and x, y, z_t are the local coordinates, then the volumetric power distribution

⁴It operates on the principle of solid-state diffusion, wherein the atoms of two solid, metallic surfaces interperse themselves by time.

⁵Is a welding process that is used to join metal to metal by using electricity to create enough heat to melt metal, and the melted metals, when cool, result in a binding of the metals [2].

is represented as follows:

$$HS = \begin{cases} \frac{6\sqrt{3}Q_T}{a_f b c \pi \sqrt{\pi}} \exp(-3\frac{x^2}{b^2} - 3\frac{y^2}{c^2} - 3\frac{z_t^2}{a_f^2}) & \text{if } z_t > 0 \\ \frac{6\sqrt{3}Q_T}{a_f b c \pi \sqrt{\pi}} \exp(-3\frac{x^2}{b^2} - 3\frac{y^2}{c^2} - 3\frac{z_t^2}{a_r^2}) & \text{else } z_t \leq 0 \end{cases} \quad (1.1)$$

With:

$$Q_T = \eta V I$$

η is the welding process efficiency

V is the voltage [V]

I is the current [A]

$$z_t = z - v(\tau - t)$$

v is the velocity of the torch

τ is the lag factor

The original Goldak model is adapted, in the Fatigue and Fracture mechanics research team [25], to the following model.

$$HS_G = \begin{cases} Q \exp\left(-\left(\frac{x}{b}\right)^2 - \left(\frac{y}{c}\right)^2 - \left(\frac{z_t}{a_f}\right)^2\right) & \text{if } z_t > 0 \\ Q \exp\left(-\left(\frac{x}{b}\right)^2 - \left(\frac{y}{c}\right)^2 - \left(\frac{z_t}{a_r}\right)^2\right) & \text{else } z_t \leq 0 \end{cases} \quad (1.2)$$

Thus, the key parameters of the considered heat source model the study case are:

- Q : The maximum volumetric power density
- b : The half of the width of the bead
- c : Penetration of the bead
- a_f : Front length of the molten zone
- a_r : Rear length of the molten zone

They are illustrated in the figure 4.5.

Table 1.1: Examples of efficiencies in some configurations of Arc welding process.

Welding process	Efficiency
Shielded metal arc welding	0.75
Gas metal arc welding	0.8
Submerged arc welding	0.9
Gas tungsten arc welding	0.8

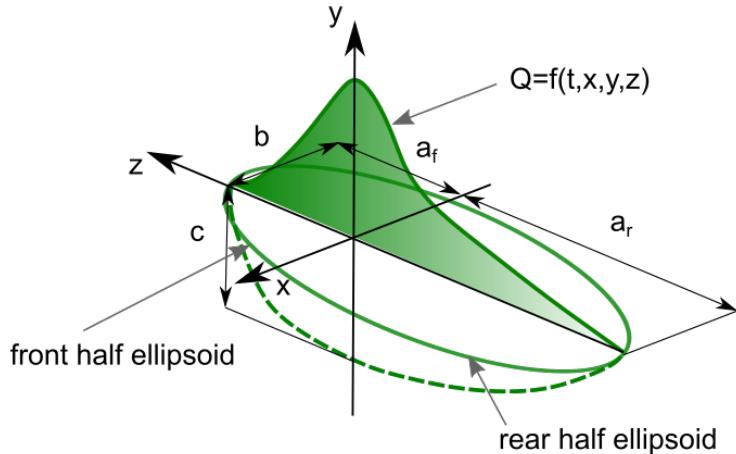


Figure 1.2: Illustration of the welding heat source model.

The maximum volumetric power intensity Q is depending on the efficiency which is dependent, in turn, on the used welding process. Here are some examples of efficiencies values accoring to [8].

In the study case we considered the efficiency to be:

$$\eta = 0.8$$

There are some correlations made by experts which allow us to reduce the number of parameters. For instance, according to [35], a_r and a_f would be different from each other because of weld pool trailing effect⁶. In high conductive materials the trailing is not significant and $a_r = a_f$. In the study case, based on the expertise of Dipl. Ing F. Dittmann, we concidered:

$$a_r = 2 \times a_f$$

⁶As the heat source is moving then the weld pool tends to be trailed and look like teardrop shape.

The volumetric power intensity HS_G in $[w/m^3]$ is modeled as a Gaussian distribution with respect to the the heat affected zone (HAZ). The HAZ is defined by the local coordinates x, y, z which are depending on the angle of inclination of the welding source e.g electrode, the starting position (x_0, y_0, z_0) and the speed of welding v .

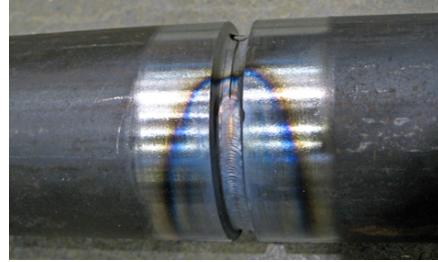


Figure 1.3: Illustration of a realistic temperature distribution on a welded pipe [8].

The figure 1.2.2 illustrates intuitively the choice of such Gaussian distribution.

1.2.3 Causal factors

Among the physical phenomena that take place in the welding processes, we cite:

- Heat transfer
- Fluid flow
- Mass transfer
- Phase transformation
- Stress effects

All those processes take place simultaneously, which make the welding process difficult to model. To reduce the complexity, only heat transfer and related stress effects are considered and the annalysis is called thermal-stress analysis.

- Heat transfer
- Stress effects

For such analysis ABAQUS FEA offer two methods in two different solvers i.e. ABAQUS/-Standard and ABAQUS/Explicit:

- Sequentially coupled thermal-stress analysis:

If the stress/displacement solution is dependent on a temperature field but there is no inverse dependency, a sequentially coupled thermal-stress analysis can be

conducted in ABAQUS/Standard. Sequentially coupled thermal-stress analysis is performed by first solving the pure heat transfer problem, then reading the temperature solution into a stress analysis as a predefined field [7].

- Fully coupled thermal-stress analysis:

A coupled temperature-displacement procedure is used to solve simultaneously for the stress/displacement and the temperature fields. A coupled analysis is used when the thermal and mechanical solutions affect each other strongly. For example, in rapid metalworking⁷ problems the inelastic deformation of the material causes heating, and in contact problems⁸ the heat conducted across gaps may depend strongly on the gap clearance or pressure[7].

In the study case, we assume that the welding process does not imply any inner heating of the material and the friction between the welded parts are negleted. Under such assumptions, the Sequentially coupled thermal-stress analysis is justified. The focus in this thesis is on the uncoupled heat trasfer analysis which is the basis of better accuracy in the mechanical analysis.

In order to simulate the absence of void between the welded parts, a tie constraint⁹ is included in the CAD model.

The heat transfer is extensively understood process and can be resumed to the following bullet points [35]:

- Heat gain from welding source
- Heat gain from leading heat source (initial heat source e.g two torches/hybrid welding)
- Heat loss from external heat sinks (to remove the heat so that dethermal distortion can be controled)
- Heat loss by convective mode
- Heat loss by radiation
- Heat loss by conduction in the base metal
- Enhanced heat extraction through water cooled backing setup
- Formation of compounds though exothermic reaction (e.g powder usage)
- Heating of base metal

⁷Metalworking is the process of shaping and reshaping metals to create useful objects, parts, assemblies, and large scale structures.

⁸A contact problem occurs when at least two bodies not mechanically joined touch each other without becoming rigidly attached.

⁹A tie constraint ties two separate surfaces together so that there is no relative motion between them.

- Melting
- Possible evaporation
- Solidification
- Cooling to ambient temperature

However we limit the scope of this study to the following:

- Heat gain from welding source (Subsection above [3.1](#))
- Heat loss by convective mode
- Heat loss by radiation
- Heat loss by conduction in the base metal

In [36], heat is defined as the energy that flows from the higher level to the lower level of temperature without any work performed. The general equation which quantify the amount of transferred heat all fit into the pattern:

$$flow = \text{transport coefficient} \times \text{potential gradient}$$

flow can be either heat flux q [W/m^2] or heat transfer rate Q [W]. The potential gradient represents a derivative or some difference expression. While, the transport coefficient depends on the particular mode of transfer.

★ Heat loss by conduction in the base metal:

Heat conduction is the diffusive transport of thermal energy. In liquids and gases, it is caused by moving atoms and moving molecules interaction, in solids by lattice oscillations and in electroconductive material by unboud electrons as well. Once the internal temperature distribution, we can calculate the heat flux by Fourier's law:

$$q_{cond} = -k\nabla T(t, x)$$

k is the conductive coefficient in [$W/(m.K)$]

★ Heat loss by convective mode:

The exchange of heat between a moving fluid (gas or liquid) and an adjoining body is named convection. Depending on the temperature proportions, heat is gained if ($T_{fluid} > T_{body}$) or lost if ($T_{fluid} < T_{body}$). If the current of the fluid is held up by a pump or blower, engineers denote that process forced convection. But also without such technical aids, a flow may emerge due to density differences in the fluid, which are brought forth on their part by temperature or concentration gradients (free convection). In our study case, we only consider free convection. Newton took the

following approach in order to calculate the heat flux from as:

$$q_{conv} = h(T(t, x) - T_0)$$

h is the heat convection coefficient in $[W/(m^2 \cdot K)]$.

★ Heat loss by radiation:

All bodies emit thermal radiation. According to the Stefan-Boltzmann law, the heat emission density q_{rad} of a body at temperature T with emissivity $\epsilon \in [0, 1]$ is calculated by:

$$q_{rad} = \epsilon \sigma (T^4(t, x) - T_0^4)$$

ϵ is the emissivity and σ the Stephan-Botlzmann $[W/(m^2 \cdot K^4)]$

All heat transfer phenomena are represented in the figure 1.4.

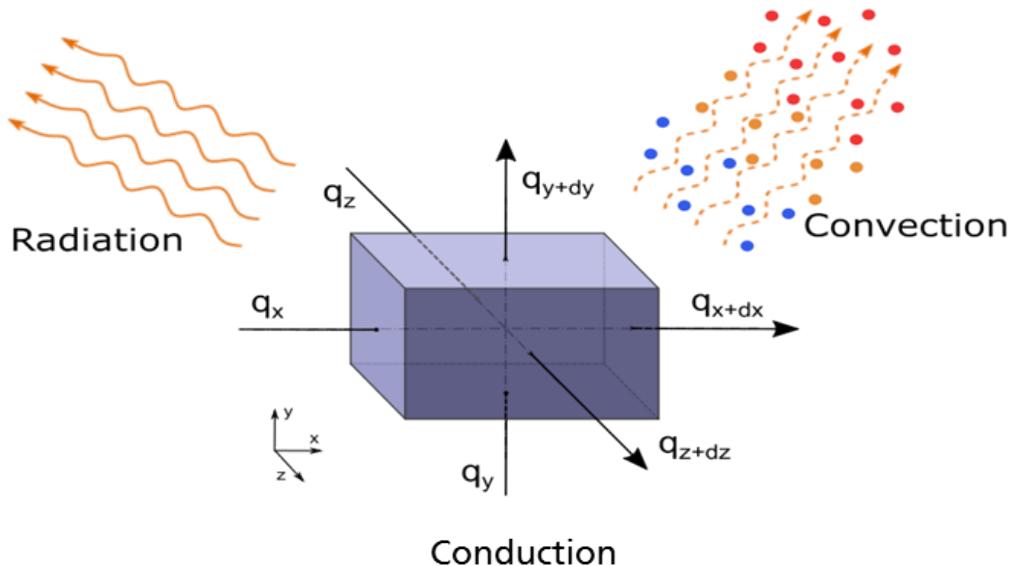


Figure 1.4: Illustration of the heat transfer phenomena considered in the welding process.

1.2.4 Case study and framework

The considered CAD model [39] is a symmetrical corss shape structure illustrated in the figure below:

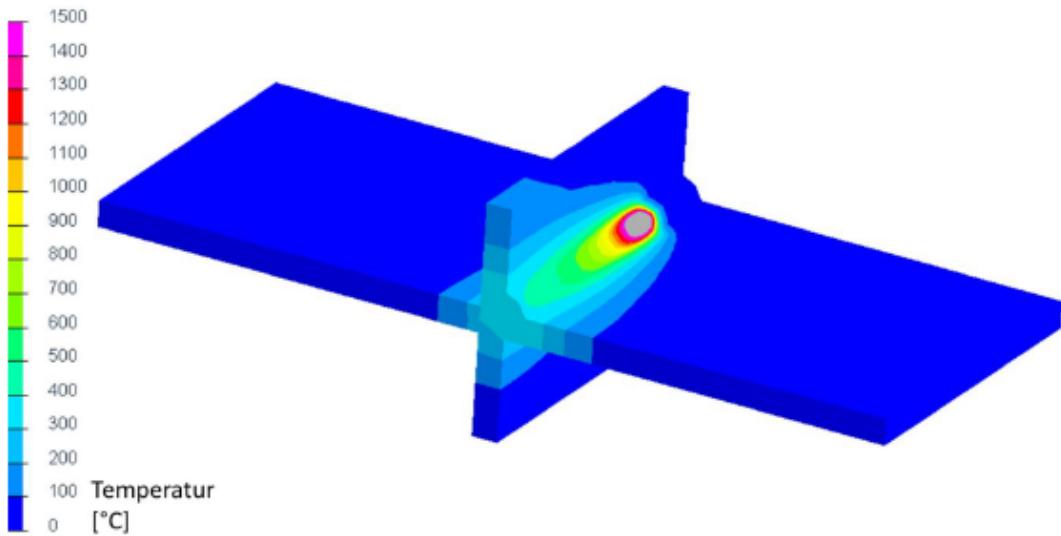


Figure 1.5: Illustrative example from a former project for a welding process simulation model [39].

Only the welding process of one layer is performed in the study case because solving the for one layer means being able to apply for other layers. Only the starting position of the heat source is changing. The position of available realistic measurements for this layer is shown in the figure below.

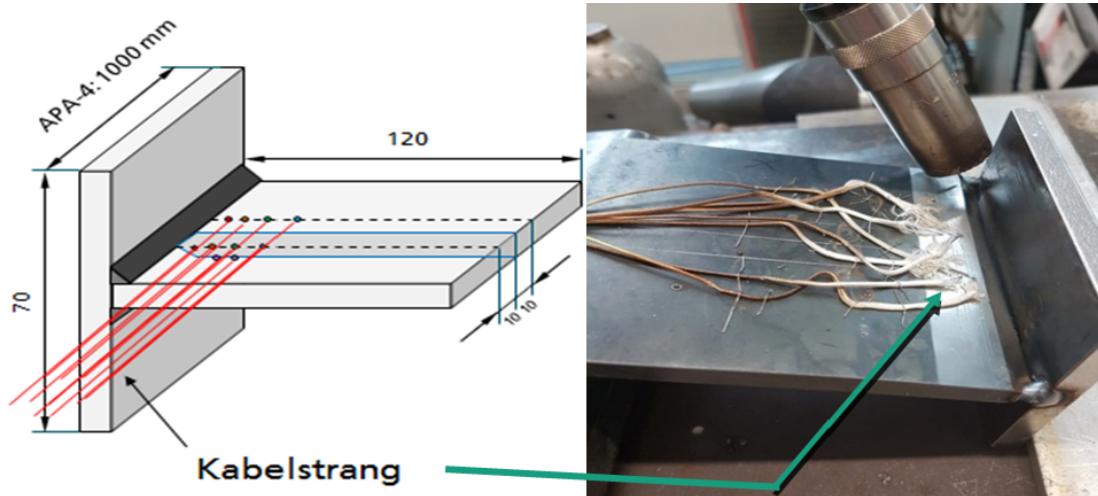


Figure 1.6: Illustrative example of the a tested component (right) and its dimension (right) [39].

And the sink temperature is 20°C . it is worth noting that the problem is first studied

for given numerical data at chosen positions.

The FEA is performed based on four necessary files:

- + Input file: Having information about the dynamic problem e.g. Time step, Boundary condition, Initial temperature.
- + User subroutine file: Where the parametrized heat source is implemented.
- + FE mesh file: Where the space discritisation is performed (C3D8 elements).
- + Material properties file e.g. specific heat capacity, conductivity, density.

Once the files above are complete, the FEA is run by specifying the number of threads (to divide the elements into parallel solvers) and submit the job throughout a batch file managed by SLURM¹⁰ in the HPC cluster.

1.3 Description of the problem

The modeling of welding process requires the simplification or the approximation of the underlying physics. Also, the material parameters are estimated or even unknown. This leads to the fact that original parameters of the process such as the heat source parameters can not be taken as input data to the model directly. The relationship between the adopted welding process model and the real process parameters is in general not predefined. In order to reach the best correlation between simulation and experiment, several simulations have to be conducted¹¹. This approach is governed by the forward problem 1.2.4 and requires highly trained personnel, much time and expenses¹². The method of choice for solving the forward problem is dependent on the accuracy required with respect to the expected use of the results. In the team I conducted the thesis at, ABAQUS FEA¹³ is used. It is a reliable and accurate software, but connected to high computational costs, especially for long evolution time. Moreover, the fitting of the parameters is done based on a trial and error approach and it requires a spetialist in that field. The problematic can be formulated in the following questions:

¹⁰SLURM is a free and open-source job scheduler for Linux and Unix-like kernels.

¹¹Depending on the complexity of the geometry several weeks of simulations can be required.

¹²For instance expenses related to the Software, energy and personnel.

¹³A software suite for finite element analysis and computer aided engineering, originally released in 1978.

- Is it possible to determine the model parameters (Inverse problem 1.3.2) systematically instead of using a trial and error approach?
- How can the parameters be determined in an efficient and effective way?
- Are there other alternatives to determine the temperature distribution (forward problem 1.3.1)?
- How can the decision on the trade-off between computational costs and accuracy made?

Those questions are centered around two problems presented in the figure 1.7.

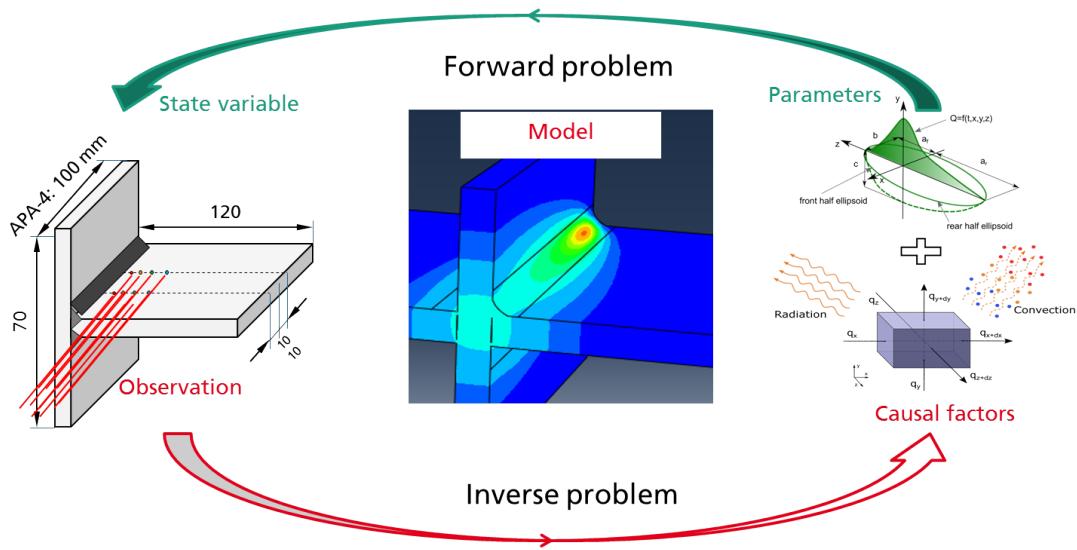


Figure 1.7: A screenshot of an exemplary documented method.

1.3.1 Forward problem

Most metallic materials¹⁴ have thermal properties (e.g. conductivity k , specific heat capacity c_p and density ρ) which are temperature dependent. This results in nonlinearities in the governing heat conduction equation. Also, the boundary conditions are considered non-linear because of the radiation model.

¹⁴Austenitic stainless steel 1.4541 https://de.materials4me.com/media/pdf/be/70/40/Werkstoffdatenblatt_zum_Werkstoff_1-4541.pdf

Given a temperature distribution $T(t, x)$ on the considered component, a heat source with volumetric power $G(x, t)$ 4.5 and an initial condition T_0 , the governing equation under non-linear boundary conditions can be formulated as follows:

$$\begin{cases} \rho C_p(T) \frac{\partial T(t, x)}{\partial t} - \operatorname{div}[k(T) \nabla T(t, x)] = G(t, x) & (t, x) \in Q \\ -k(T) \frac{\partial T(t, x)}{\partial n} = h(T(t, x) - T_0) + \epsilon\sigma(T^4(t, x) - T_0^4) & (t, x) \in \Sigma \\ T(0, x) = T_0 & x \in \Omega \end{cases} \quad (1.3)$$

The time and space domain are denoted in the notations section . In the meantime, simplifying assumptions such as considering the heat conductivity of the material as constant can reduce the difficulty to solve the forward problem. ABAQUS Standard solver does the same assumption when calculating the Jacobian matrix and considers that the thermal conductivity varies slightly with temperature. Also, the conductivity is considered to be space independent as the joint components have to be of the same material in order to reach the melting temperature at the same time without damaging one of the two parts. The simplified equation which is considered by ABAQUS solver can be formulated as follows:

$$\begin{cases} \rho C_p(T) \frac{\partial T(t, x)}{\partial t} - k(T) \nabla^2 T(t, x) = G(t, x) & (t, x) \in Q \\ -k(T) \frac{\partial T(t, x)}{\partial n} = h(T(t, x) - T_0) + \epsilon\sigma(T^4(t, x) - T_0^4) & (t, x) \in \Sigma \\ T(0, x) = T_0 & x \in \Omega \end{cases} \quad (1.4)$$

The Neumann boundary conditions related to convection and radiation are modeled as follows:

- Convective term:

$$q_{conv} = h(T(t, x) - T_0)$$

- Radiation term:

$$q_{rad} = \epsilon\sigma(T^4(t, x) - T_0^4)$$

The temperature of the environment is considered as the initial temperature of the material. The heat convection coefficient h , the emissivity ϵ and the Stephan-Bolztmann constant σ are considered as temperature independent. Thus, the non-linear boundary conditions are caused by radiation.

Amongst the commonly used methods to solve the dynamic non-linear heat transfer equation is the finite difference method (FDM) and the finite element method (FEM). The main advantages of the FEM over the FDM is that irregular boundaries due to complicated geometries can be handled easily. One can have a look at [40] and [15] for more details about FDM and FEM methods. Another method used to solve such problem is called boundary element method (BEM)[43], where the solution is performed only based on the boundary domain discretization.

Out of all the mentioned methods above and other engineering analysis methods, the FEM was found to be the most used one [14]. It is also the adopted approach in the team where this thesis has been conducted. ABAQUS FEA or formerly ABAQUS is the software used in both methods, the finite element analysis and computer-aided engineering. Some engineers prefer the software HyperMesh for the pre-processing phase. More details about the implemented finite element method in ABAQUS Standard to solve the nonlinear heat equation are given in section 2.3.

1.3.2 Inverse problem

We recall that our main goal is to reconstruct the density of the heat source during the welding process based on given realistic measurements¹⁵. The heat source model itself is parametrized according to the Goldak model [22] and the parameters have to be physically applicable as explained in subsection [?]. This problem belongs to the bilevel optimal control class as it is a hierarchical optimization problem of two decision makers, i.e the heat source density against the choice of the parameters. One can find a satisfying overview of existing literature in 'Bilevel Optimization' by S. Dempe [18] where more than 1350 references are listed [32]. According to P. Mehlitz and G. Wachsmuth [32], bilevel optimal control may be subdivided in three subclasses depending on which decision maker has to perform optimal control.

- **Case 1:** Only the upper lever problem solves the PDE/ODE while the lower level problem depends explicitly on the terminal state of the leader's problem, e.g. gas balancing in energy networks.
- **Case 2:** Only the lower level has to solve the optimal control problem and the parameters have to be reconstructed by certain measurements.
- **Case 3:** Both, leader and follower, have to solve an optimal control problem, e.g.

¹⁵Temperature evolution over time on predefined spatial positions.

when time-dependent coupling of container crane is under consideration (see [27]).

According to the given definition [32], this problem belongs to case 2 and is a typical example of a bilevel optimal control problem. It is also common speak of a inverse optimal control, hence the chosen title of the thesis.

With this convention of calling the heat source feasible parameter set as a lower level (or follower's) problem and the state variable approximation as an upper lever (or leader's) problem our problem can be schematized as follows:

Solve the lower level problem
Subject to
Optimal control problem

The optimal control problem, in turn, is defined as follows:

Minimize the objective functional
Subject to
Upper level problem

This problem is approached theoretically in section 2.2 and then a gradient and/or machine learning based solutions are proposed in chapter 3 and 4, before all results are discussed in chapter 5.

1.3.3 Objectves and overview

The fact that both, forward and inverse problem, were introduced, it is here important to point out the scope of this master thesis. The goal is not to substitute ABAQUS solver and develop a method to solve the forward problem, but to set the main focus on the inverse problem.

The current way at the team where this thesis took place is a manual workaround where the parameters of the heat source are adapted manually and iteratively until the temperature-time history at defined positions outside the weld zone matches with experimental measurement data. This process is mainly based on expertise and for complicated geometries it can require days or even weeks of laborious fitting¹⁶. In the

¹⁶As stated by my supervisor Dipl. Ing F. Dittmann.

prototype simulation model chosen for this study [39], one simulation of 20 seconds of welding, with a time step of 0.266 s (76 frames) and using multithreading (2 threads) lasts around 20 minutes.

The objective then is to equip the Fatigue and Fracture Mechanics group with a decision tool which identifies systematically parameters¹⁷ with as less as possible human interaction.

Also, the goal is not to reach perfect scores on accuracy for the simple reason that our parametric study does not include material properties but only the heat source parameters. Also, the non-linearity of the problem results from chemical reactions or phase change transitions which are not modeled in the first place.

The optimal solution (locally in the worst case) can be inserted in ABAQUS FEA later on and serve as a good starting point for the specialists.

Within this thesis three solutions were proposed:

- Gradient-based optimization tool (in subsection 3.1)
- Multilayer perceptron (MLP) tool (in subsection 3.2)
- Model order reduction (MOR) tool (in subsection 4.3)

Another way of extracting the results is to use a machine learning based solution as an initial guess for the proposed gradient based tools (see summary chapter 6).

1.4 Methodology

The mathematician George Pólya, in his book *How to solve it* [38], suggests the following steps when solving a mathematical problem:

1. Understand the problem
2. Divide the problem
3. Carry out the plan
4. How could it be improved?

Those principles allowed me to structure this work and in the case of difficulties, dividing the problem in smaller sub problems is an extremely powerful principle.

¹⁷Here the focus is on Goldak parameters but they can be also material parameters.

1.4.1 Agile paradigm

There are two main paradigms to organize, develop and meet the delivery time:

- Classical project management
- Agile project management

On one hand, traditional project management approach is linear and depends on predictable tasks and former experience. In this case, the problem of inverse optimal control in non-linear boundary conditions is theoretically challenging and many approaches have to be explored. Thus, the traditional paradigm might not address all the challenges. On the other hand, agile management gives prominence to flexibility, it is iterative and aims at constantly incorporating and exploring new ideas while updating plans. The only feature in the agile model which was not compatible with the expectation is documentation. While the agile approach is more about a working product and documenting only when needed 'just barely good enough' [11], the traditional model gives prominence to static documentation. The documentation of the developed softwares is done using the Sphinx documentation generator based on the Python package numpy documentation as illustrated in the figures 1.8.



Figure 1.8: A screenshot of the developed Abacal tool documentation (main page).

```

module.Gradient_descent.Gradient_Loss(L1, Q1, Q2, target)
    Returns the gradient of the cost function estimated two initial parameter set. The partial derivatives are approximated by backward difference.

    Given the direction and the orthogonal basis:
        
$$h^i = (0, \dots, h_i, \dots, 0), e_{i \in \{1, \dots, n\}}$$

    With:
        
$$h_i = \langle Q_2 - Q_1, e_i \rangle_{R^n}$$

    We approximate the partial derivatives of  $J(x_0, \dots, x_i, \dots, x_n)$  by backward difference as follows:
        
$$\partial_{x_i} J = \frac{J(Q_1 + h^i) - J(Q_1)}{h_i}$$

    For more explanations one can check the following wikipedia pages: here or here.

Warning: When  $Q_1$  is equal to  $Q_2$  a default value of the gradient is set in the method. It can be changed in following line:


```
>>> if Q1[1] == Q2[1]:
>>> grad[1]= 0.1
```

Example:

```
>>> Q1 = np.array([1,2,3])
>>> Q2 = np.array([1,1,2,1,3,1])
>>> Targets = Gradient_descent.Load_profile(targets_file,nb_col_profiles)
>>> L1,profile1 = Gradient_descent.Loss_calc(Q1,Targets)
>>> grad = Gradient_Loss(L1,Q1,Q2,target)
```


```

Figure 1.9: A screenshot of an exemplary documented method.

The figures 1.8 and 1.9 are shown for illustration and the full documentation is hosted on Github and can be consulted via the following Github link: https://youness-elh.github.io/Abacal_doc/. For the other developed tools, the documentation is in notebooks in addition to slides when requested. The unitary tests are included in parallel to developing as recommended by the Agile paradigm proponents. Regarding the time management, frequent meetings were scheduled during the thesis on a weekly basis in addition to spontaneous meetings when results needed to be discussed. The tools which were essential to the accomplishment of this thesis are:

- HyperMesh for simulation model pre-processing
- ABAQUS Standard for simulation model processing
- ABAQUS Viewer for simulation model post-processing
- MobaXterm for remote computing in the HPC cluster
- Qt Designer for desktop applications development
- Python with its packages¹⁸ for coding
- Sphinx for generating static web page for documentation

In addition to that, other tools were used for communication like Microsoft Teams, image processing like Inkscape and Mendely for bibliography.

¹⁸For instance but not limited to: Numpy, Scipy, TensorFlow, Keras, OpenCV, mpi4py

1.4.2 Bibliography

In addition to the cited references all along this thesis, other articles or reports were inspirational and had provoked implicitly new ideas. I want to name here: T. Daniel et al.[17], S. Gros and M. Diehl [24], K. Eppler and F. Tröltzscher [20], I. M. Martynenko [31], D. Buckley [14], A. Pittner [36] and V. Karkhin et al. [26]. Moreover, the lecture of Prof. Christophe Prud'Homme [6] on Model order reduction (Reduced basis method) had inspired me and is the basis of the white box chapter 4. The lecture notes of Prof. Yannick Privat [37] in both Optimization and Optimal control courses were of an enormous help and this thesis would not have been possible without them. Also, the lectures in Machine learning and Statistical modeling of Dr. Vincent Vigon [19] were essential beside the practical work which was applicable to many of the encountered challenges along this thesis.

1.4.3 Thesis roadmap

The workload of the thesis is basically divided in three main proposed solutions of the problematic:

- Gradient-based optimization packages: Tools which can be integrated to ABAQUS as an optimization loop
- Multilayer Perceptron (MLP) packages: Forward and inverse problem
- Model order reduction packages: Inverse optimal control of the discretized problem

Here, a Gantt chart detailing the different thesis sub packages and the corresponding milestones is presented. The Excel template has been downloaded from [42].

Master thesis - Project Schedule

Gantt Chart Template © 2006-2018 by Vertex42.com.

Strasbourg University / Fraunhofer IWM

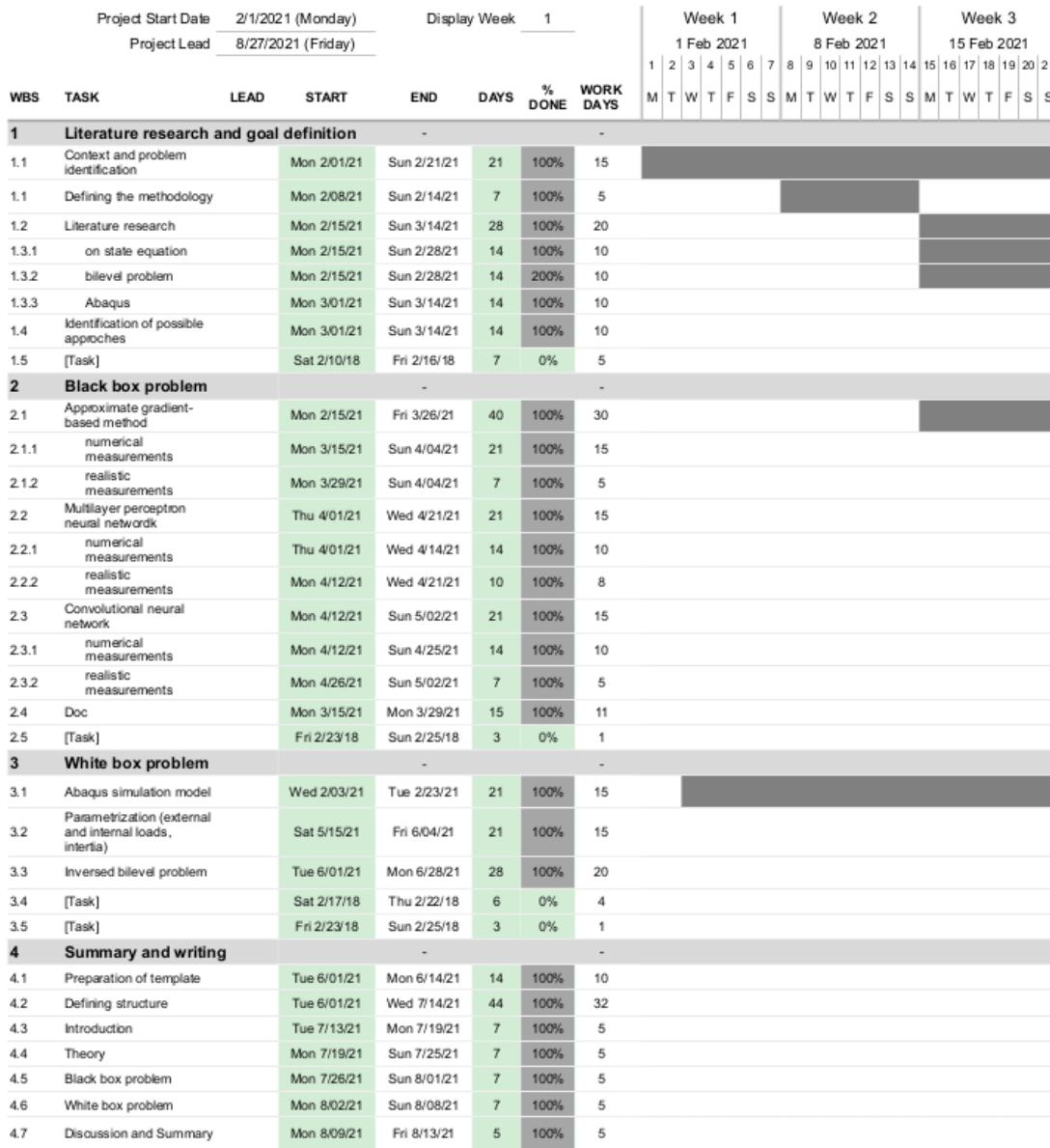


Figure 1.10: A screenshot of the master thesis Gantt chart.

For a closer look, the full Excel sheet is available in the following Github repository:

https://github.com/youness-elh/Master_Gantt/tree/main.

Chapter 2

Fundamentals and problem formulation

In this chapter, the optimal control problem under consideration is presented in abstract form with its related fixed notation. This is an abstraction of the previous chapter in which all features of the non linear heat equation were introduced phenomenologically. In section 2.1, the adopted notation which will be used throughout this report is specified in addition to relevant definitions. Then in section 2.2, the main components of the parabolic optimal control problem are introduced in a formal way. The state equation, the optimality conditions of the bilevel optimization problem are depicted. Finally, in section 2.3, the proposed descritisation method is described and the used simulation software i.e. ABAQUS CAE is introduced.

2.1 Assumptions and notations

Let $\Omega \subset \mathbb{R}^3$, a convex bounded domain representing a thermal conductive body with a smooth boudary $\partial\Omega$ which is for instance of class \mathbf{C}^2 . The outward normal derivaive is denoted $\frac{\partial}{\partial n}$. The problem is formulated in the time interval $\mathbf{I} = [0, t^e]$ where $t^e > 0$. Lebesgue and Sobolev spaces are denoted respectively with $\mathbf{L}^p(\Omega)$ and $\mathbf{W}^{k,p}(\Omega)$. $\mathbf{L}^p(\Omega)$ may be defined as a space of measurable functions for which the p-th power of the absolute value is Lebesgue integrable according to the p-norm defined such that:

$$\|f\|_{L^p(\Omega)} = \left(\int_{\Omega} |f|^p \right)^{\frac{1}{p}} < \infty$$

It is a complete normed vector space so it is a Banach space similarly to Sobolev space $\mathbf{W}^{k,p}(\Omega)$. This latter is intuitively a space of functions possessing sufficiently many derivatives for some application domain, such as partial differential equations. Within this master thesis report, we are placed in the case of the heat transfer equation. In that case, $p = 2$ is enough to set the framework of the needed derivatives in the heat equation 2.1. Also, in that particular case i.e $p = 2$, the Sobolev space is denoted $H^2(\Omega)$ as it is a Hilbert real inner product space. The inner product defined at a Hilbert space is denoted $\langle \cdot, \cdot \rangle_{H(\Omega)}$ and the induced norm is denoted $\|\cdot\|_{H(\Omega)}$.

We will be using the notations:

$$Q =]0, t^e[\times \Omega \quad \text{and} \quad \Sigma =]0, t^e[\times \partial\Omega.$$

From a concrete point of view, the Hilbert space allows to manipulate functions ¹ throughout functionals ² defined by the space structure e.g inner product.

Formally we have the following definition [16]:

Definition : *The dual of a Hilbert space V^* is the space of all bounded linear functionals on V . This has a natural norm induced by the norm on the underlying space:*

$$\|A\|_{V^*} = \sup_{\|f\|_V=1} |A(f)|$$

In fact, this is itself a Hilbert space.

Given a $A \in V^*$, denote the action of A on f (or equivalently f on A) by:

$$\langle A, f \rangle_{V^*} = A(f)$$

Example Let Φ be a test function in $L^2(\Omega)$. let the functional A which is defined as follows:

$$\begin{aligned} A : \mathbb{L}^2(\Omega) &\longrightarrow \mathbb{R} \\ A(T) &= \int_{\Omega} T\Phi \end{aligned}$$

Then $A(T)$ is an element of the dual space of $L^2(\Omega)$.

¹for example test function Φ and the approximated solution $Q \ni (t, x) \mapsto T(t, x) \in \mathbb{R}$

²Like the Schwartz distributions

There is a fundamental connection between a Hilbert space and its dual where the functionals are living e.g. nodal approximated solution in $\mathbb{R}^{\mathcal{N}}$ ³. This connection is very important to grasp the transition between the weak formulation 2.2 and its descritisation explained in section 2.3. The Riesz Representation Theorem captures this connection and state it as follows:

Theorem 2.1.1: *Any bounded linear functional $A \in V^*$ can be uniquely represented by a $g \in V$, via:*

$$\langle A, T \rangle_{V^*} = \langle g, T \rangle_V$$

Moreover, the norms agree: $\|A\|_{V^*} = \|g\|_V$.

This defines a canonical linear map, the Riesz map:

$$\begin{aligned}\mathcal{R} : V^* &\longrightarrow V \\ A &\longmapsto g\end{aligned}$$

This Riesz map is an isometric isomorphism.^a

^aA bijective distance-preserving transformation between metric spaces.

Example Let $V = L^2(\Omega)$ and let the functional A :

$$\begin{aligned}A : \mathbb{L}^2(\Omega) &\longrightarrow \mathbb{R} \\ A(T) &= \langle A, T \rangle = \int_{\Omega} T\end{aligned}$$

Then its $L^2(\Omega)$ Riesz representation is the constant function $g(x) = 1$.

When discritizing the space domaine Ω into finite elements, the considered dual space is $\mathbf{R}^{\mathcal{N}}$. In that case, the matricial Riesz representation generates Stiffness, Mass or Load matrices. More details about the method will given in section 2.3. For more insight on the function spaces we are going to exploit, the following monograph has been recommended [10].

Regarding the optimal control problem, we will use the Radamacher theorem to assume differentiability of some functionals.

³ \mathcal{N} is the number of nodes in the space domain discritisation

Theorem 2.1.2: Let U is an open subset of \mathbf{R}^n and $f : U \rightarrow \mathbf{R}^m$ is Lipschitz continuous^a, then f is differentiable almost everywhere in U . Those points in U at which f is not differentiable form a set of Lebesgue measure zero.

^a $\exists K > 0, \forall x_1, x_2 \in U, \mathbf{R}^m$ so that $\|f(x_1) - f(x_2)\|_{\mathbf{R}^m} \leq K|x_1 - x_2|\|_{\mathbf{R}^m}$ holds.

The proof of the Rademacher theorem is detailed in [34]. The application of such theorem is relevant to prove differentiability in distribution sens of objective functional of the optimal control problem 2.2.2.

The differentiability of a functional is relevant for our class of optimization problems. It allows for instance to have an idea about the convexity as shown in the proposition below retrieved from the lectures of Prof. Yannick Privat [37].

Proposition 2.1.3: Let $J : H \rightarrow \mathbf{R}$ a differentiable function, then the following propositions are equivalent:

- J is strongly convex
- $J - \frac{\alpha}{2}$ is convex
- J is α -convex $\iff \langle \nabla J(v) - \nabla J(u), v - u \rangle \geq \alpha \|v - u\|^2$

The third proposition is very practical and will be used to prove the *alpha*-convexity of the objective functional of the optimal control problem 2.2.2.

Also the following theorem is important for the uniqueness of the solution in Hilbert space H with its induced norm $\|\cdot\|$ and its inner product $\langle \cdot, \cdot \rangle$ [37].

Theorem 2.1.4: Let $J : H \rightarrow \mathbf{R}$ a continuous function, α -convex and U_{ad} the feasible nonempty convex closed set.

Hence the problem $\{\inf J(v), v \in U_{ad}\}$ has a unique solution.

2.2 Problem setting

2.2.1 State equation

We recall that the welding process simulation is based on solving the non linear partial differential equation in order to find the temperature distribution $T(t, x)$ on the

considered component.

$$\begin{cases} \rho C_p(T) \frac{\partial T(t, x)}{\partial t} - k(T) \nabla^2 T(t, x) = G(t, x) & (t, x) \in Q \\ -k(T) \frac{\partial T(t, x)}{\partial n} = h(T(t, x) - T_0) + \epsilon \sigma(T^4(t, x) - T_0^4) & (t, x) \in \Sigma \\ T(0, x) = T_0 & x \in \Omega \end{cases} \quad (2.1)$$

This partial differential equation models the heat flow subject to non linear cooling and also fluid diffusion within a semi-permeable membrane [21]. it is worth noting that in welding processes, the material conductivity is space independant as the joint materials are supposed to have the same properties. Also, the conductivity is known to have a low variation wit the temperature, so we consider from now on that $k = k(T)$.

- **Weak formulation** It is worth noting that the discontinuous coefficients of the problem were not considered in the analysis presented below. The generalized functions or Schwartz distributions allow us to generalize the classical notion of functions which are not differentiable e.g. $T \mapsto C_p(T)$. In that context all functions become differentiable in the Schwartz distribution sens.

The solution we are seeking satisfies the following variational equation:

$$\int_Q \rho C_p(T) \frac{\partial T(t, x)}{\partial t} \Phi - \int_Q k \nabla^2 T(t, x) \Phi = \int_Q G(t, x) \Phi$$

Where Φ is a test function in the Hilbert subspace defined in 2.3.

After applying the Green's formula,

$$\int_Q \rho C_p(T) \frac{\partial T(t, x)}{\partial t} \Phi + \int_Q k \nabla T(t, x) \nabla \Phi = \int_Q G(t, x) \Phi + \int_{\Sigma} k(t) \frac{\partial T(t, x)}{\partial n} \Phi$$

- **Existence and uniqueness of the solution** We will be seeking the solution $T(t, x) \in C^0(I, H^2(\Omega)) \cap L^2(I, V)$, satisfying:

$$\int_Q \rho C_p(T) \frac{\partial T(t, x)}{\partial t} \Phi + \int_Q k \nabla T(t, x) \nabla \Phi = \int_Q G(t, x) \Phi + \int_{\Sigma} Loss(T) \Phi \quad (2.2)$$

We denote $k \frac{\partial T(t, x)}{\partial n} = Loss(T)$.

$$Loss(T) = -h(T(t, x) - T_0) - \epsilon\sigma(T^4(t, x) - T_0^4) \leq 0$$

Where V is a Hilbert space defined according to H. Brezis [13] as follows:

$$V = \left\{ T(t, x) \in H^2(\Omega) \mid -\frac{\partial T(t, x)}{\partial n} = h(T(t, x) - T_0) + \epsilon\sigma(T(t, x)^4 - T_0^4) \text{ a.e. on } \Sigma \right\} \quad (2.3)$$

Problem 2.1 has been investigated by H. Brezis [13], who proved that if the initial solution T_0 is in $L^2(\Omega)$ then $T(., t)$ is in $L^2(\Omega)$ for each $t > 0$. The proof of uniqueness is funded on the maximal monotony of the predefined operator generating for each $T_0 \in V = L^2(\Omega)$ a unique strong solution $T(., t)$ of 2.1 with $T(t, x) \in C^0(I, L^2(\Omega))$. For details, one can consult [28] where Lions-Magenes defined $\frac{\partial T(t, x)}{\partial n}$ as an element of the boundary space $H^{\frac{1}{2}}(\Sigma)$ and $T(t, x)|_{\Sigma} \in H^{\frac{3}{2}}(\Sigma)$.

In additon to $L^2(I, \Omega)$, other smoothing effects of the solution in the other spaces $L^p(\Omega)$ were partially invistigated in [21].

2.2.2 Bilevel optimization problem

The bilevel problem is approached as one problem theoritically according to the article [32]. Later we split into two sub-problems which are analysed according to the lectures of Prof. Yannick Privat [37] on optimal control and optimization.

Uniqueness of the solution

The problem can be written in a formal way as follows [32]:

Let F be the objective functional of the parametric optimization (follower's) problem and $X \in X_{ad} \subset \mathbb{R}^d$ is the parameter set of the Goldak model [22]. $G(., .) \in L^2(I, L^2(\Omega))$ is the control. The bilevel programming problem is:

$$\begin{aligned}
& \min_{X,G} F(X, G) \\
\text{S.t.} \\
& X \in X_{ad} \\
& G \in \Psi(X)
\end{aligned}$$

Where $\Psi : \mathbb{R}^d \mapsto L^2(I, L^2(\Omega))$ is the solution mapping of the parametric optimization problem.

$$\begin{aligned}
& \min_G f(G) \\
\text{S.t.} \\
& G \in \Gamma(X)
\end{aligned}$$

Existence of the solution

And f is the objective functional of the state variable level (leader's) problem subject to $\Gamma : \mathbb{R} \mapsto L^2(I, L^2(\Omega))$ which is the solution mapping of the upper problem defined in 2.1.

Existence of the solution

Theorem 2.2.1: According to [32], the bilevel problem 2.2.2 possesses an optimal solution in each of the following settings:

1. The mapping f is weakly sequentially upper semicontinuous on $L^2(I, L^2(\Omega))$ while Γ is weakly inner semicontinuous on $L^2(I, L^2(\Omega))$.
2. The Banach space \mathbb{R}^d is finite-dimensional. The mapping f is upper semicontinuous on $L^2(I, L^2(\Omega))$ while Γ is weakly inner semicontinuous on \mathbb{R} .

The proof is based on the assumptions that the feasible set X_{ad} is nonempty and the objective functional F is lower semicontinuous. One can find more details in the proof provided in [32].

Objective functionals

For the lower level problem we will be measuring the difference between the obtained optimal control solution and the parametric Goldak heat source and the ideal case is when the value is zero. For such problem, the following target function is suitable:

$$\begin{aligned} \text{Minimize } F(X) &= \frac{1}{2} \|HS_G(X) - G\|_{L^2(I, L^2(\Omega))}^2 \\ X &\in X_{ad} \\ G &= \text{Optimal control solution} \end{aligned}$$

Where HS_G is the Goldak heat source model defined in the introductory chapter 4.5 and the optimal control problem is defined below.

For the upper level problem we consider:

$$\begin{aligned} \text{Minimize } J(G) &= \frac{1}{2} \|T(t, x) - \hat{T}\|_{L^2(I, L^2(\Omega))}^2 + \frac{\alpha}{2} \|G\|_{L^2(I, L^2(\Omega))}^2 \\ G &\in L^2(I, L^2(\Omega)) \\ T &\text{ solve 2.1} \\ \hat{T} &= \text{Measurements data} \\ \alpha &= \text{regularization coefficient} \end{aligned}$$

The first term of the functional aim to minimize the difference between reality and simulation state variable. While, the second term is for both smoothing the solution and penalizing the consumed power to avoid unrealistic high intensities.

Having one unique solution in the lower level for at least one solution of the upper level is not trivial. It is actually depending on the convexity , regularity and smoothness of the objective functions which makes the problem theoretically challenging. For such situations, the so-called optimistic or pessimistic approach have been proposed [32]. The optimistic approach examines the best possible outcome and chooses the 'best' of the 'best'. While the pessimistic approach examines the worst possible outcome and chooses the 'best' of the 'worst'. Practically, we consider the presented bilevel problem as two separate problems. First, we solve the optimal control problem then we inject the solution in the lower level problem to determine the best parameter set $\in X_{ad}$. The outcome of the lower level problem is thus depending on the best solution given by the upper level criterion subject to the non linear heat transfer equation 2.1.

- **Optimal control problem:** With analogy to the linear example detailed in the optimal control lecture of Prof. Yannick Privat [37], we will set the conditions under which the problem 2.2.2 have a unique solution. In ordre to obtain such result guaranteed by theorem 2.1, we prove that:

★ Is the objective functional J differentiable?

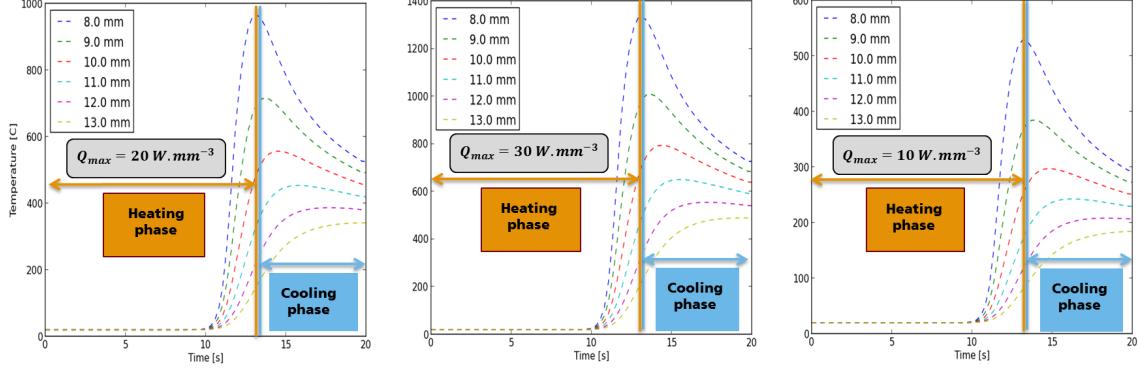
$$J : U_{ad} \ni u \longmapsto J(u) = \frac{1}{2} \|T(t, x) - \hat{T}\|_{L^2(I, L^2(\Omega))}^2 + \frac{\alpha}{2} \|u\|_{L^2(I, L^2(\Omega))}^2$$

Differentiability of J is depending on the differentiability of $u \longmapsto T_u$ thanks to the chaine rule as the norm is differentiable. In fact, given a direction $h = v - u \in U_{ad}$ ⁴ we could prove that $\lim_{\epsilon \rightarrow 0} \frac{T_{u+\epsilon h} - T_u}{\epsilon} = DT(u).h$ (if it exists) is linear and continuous. Unlike the linear example given in the lecture [37], we have non linear Robin boundary conditions. The continuity and linearity coming from the fact that $DT(u).h = DT(v).h = T_v - T_u$ is not possible as far as I know. The alternative is using the Rademacher theorem 2.1 applied to the $u \longmapsto T(u) = T_u$ functional. The assumed condition is based on the intuitive definition of a Lipschitz continuous function which is limited on how fast the function can change. If we consider only the problem at the cooling phase, (see figure 2.1) the change on the state variable $T_u(t, x)$ given a control u is not as steep as in the heating phase. Hence, the time interval on which we integrate should be changed to $I' = [\tau, t^e]$ where τ depends on the position of the measurements and the speed v of the welding assumed to be constant parameter. If \vec{y} is the moving direction and y_0 is the starting position and y is the measurement positon then $\tau \geq \frac{y - y_0}{v}$.

At $t = \frac{y - y_0}{v}$ we will have the maximum temperature if the measurement position is close to the welding area.

⁴ $u + \epsilon h = \epsilon v + (1 - \epsilon)u \in U_{ad}$ which is convex

Figure 2.1: Illustrative comparison of the simulated temperature for three different controls at different positions on the x axis on the graphs label i.e. y and z coordinates are fixed (-4,10).



With such hypothesis, we can write the derivative of J on u in $h = v - u$ direction $DJ(u).h$ as follows:

$$\langle \nabla J(u), h \rangle_{L^2(I', L^2(\Omega))} = \langle T_u - \hat{T}, DT(u).h \rangle_{L^2(I', L^2(\Omega))} + \alpha \langle u, h \rangle_{L^2(I', L^2(\Omega))} \quad (2.4)$$

Now we prove the second necessary condition to apply 2.1.

* Is J α -convex ?

$$\langle \nabla J(v) - \nabla J(u), h \rangle_{L^2(I', L^2(\Omega))} \geq \alpha \|h\|_{L^2(I', L^2(\Omega))}^2$$

Once again, in the non linear case, we don't have $DT(u).h = DT(v).h = T_v - T_u$ ⁵ which allows to write:

$$\langle \nabla J(v) - \nabla J(u), h \rangle_{L^2(I', L^2(\Omega))} = \|T_v - T_u\|_{L^2(I', L^2(\Omega))}^2 + \alpha \|h\|_{L^2(I', L^2(\Omega))}^2$$

And conclude $\langle \nabla J(v) - \nabla J(u), h \rangle_{L^2(I', L^2(\Omega))} \geq \alpha \|h\|_{L^2(I', L^2(\Omega))}^2$, hence J is *alpha*-convex.

Neglecting all losses

The proposed alternative is to integrate the objective functional on a space domain Ω' which is far from the welding zone, so that the non linear loss function can be neglected. In other words:

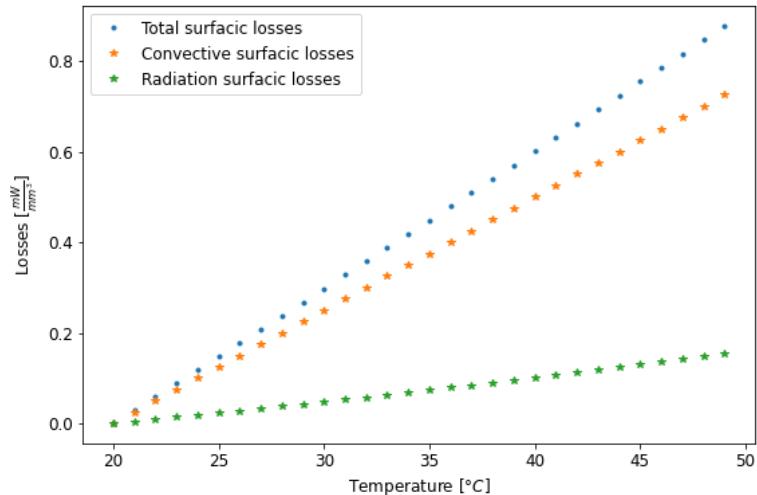
$$Loss(T) = -h(T(t, x) - T_0) - \epsilon \sigma(T^4(t, x) - T_0^4) \approx 0$$

⁵ $DT(u).h$ is only depending on the direction h . Details in the lecture's complement [37]

We recall that the convective coefficient is $h = 25e - 3mW/(mm^2.K)$ and the emissivity of air $\epsilon = 0.8$ and the Stephan-Boltzmann constant $\sigma \approx 5.67e - 11mW/(mm^2.K^4)$. As an illustrative, we consider a state variable average on I' at a position $\in \Omega'$ which is approximatively 50 °C for instance with an ambiant temperature of 20 °C. After injecting in 2.2.1, the approximated average of the surfacic total losses is $0.9mW/mm^2$. Which can be neglected from en engineering point of view.

The figure below shows more examples on the estimated losses for different measured temperatures.

Figure 2.2: Illustrative estimations of the losses at an exemplary positions eventually far from the welding zone.



For more accuracy, we can take the measurements as far as possible but with a value greater than the sink temperature. In the meantime, the measurement data has to be very precise and smoothed. So, the results accuracy for such assumption is dependant on two factors:

- **Geometry constraints:** Weither the welded component is long enough to put sensors on positions where the losses can be neglected. The biggest the component is, the better it is.
- **Measurements uncertainty:** If we consider very low temperatures in the optimal control problem, small perturbations in the target data might mislead the optimization algorithm.

Linearized model of the losses

From the figure above 2.2, we can see that the total surfacic losses are comparable to the convective losses, whereas the radiation losses can be neglected. The interval of temperatures for which the radiation are neglected is higher this time is higher. This allows positionning sensors on closer areas to the welding zone. It is a better solution for small components or/and measurements with high uncertainty. The proposed linearisation is based on the intuition that $u \mapsto T_u$ does not change a lot in the cooling phase 2.1. Hence the Lipschitz constante is very low ~ 0 and $DT(u).h \sim 0$. With such hypothesis, we have $T_u \approx au + b$. Thus, $Loss(T_u) \approx hau + hb - hT_0 = a'u + b'$ with $a' = ha, b' = hb - hT_0$ and $a, b \in \mathbb{R}$ can identified using the least square method. The least square problem is reformulated as follows:

$$\begin{aligned} \text{Minimize } Loss(a, b) &= \|AX - Y\|_{\mathbf{R}^n}^2 \\ X &= (a, b)^T \in \mathbf{M}_{2,1}(\mathbb{R}) \end{aligned}$$

Where $A = \begin{pmatrix} u_1 & 1 \\ \vdots & \vdots \\ u_n & 1 \end{pmatrix}$ and $Y = \begin{pmatrix} T(u_1) \\ \vdots \\ T(u_n) \end{pmatrix}$. Given that $(a, b) \mapsto Loss(a, b)$ is convex of class C^1 , hence the first order optimality condition $\nabla Loss(X^*)$ gives:

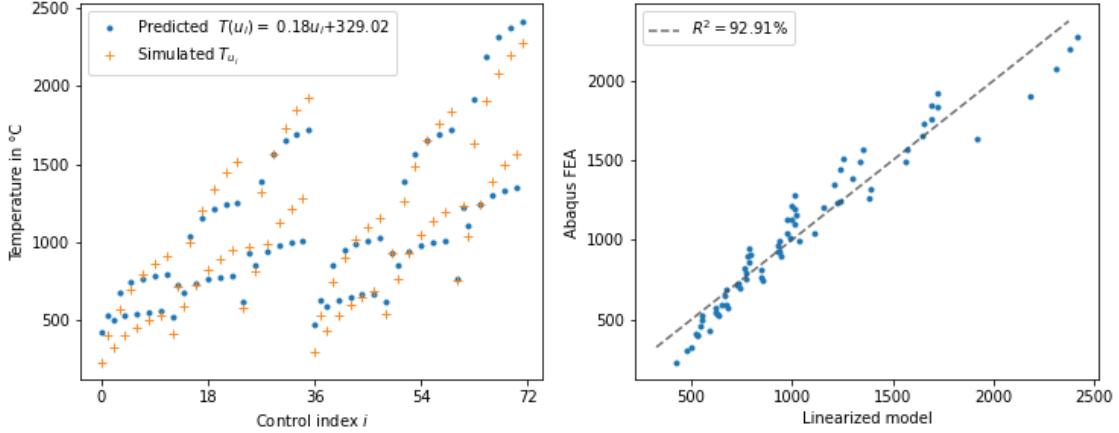
$$A^T @ A @ X^* = A^T @ Y$$

And given that $u_i! = 1 \forall i \in [1, n]$ then the second order optimality condition deliver the uniqueness of the solution (i.e $A^T @ A = \nabla^2 Loss$ is positive-definite).

The solution is $X^* = (A^T @ A)^{-1} @ A^T @ Y$ and the related determination coefficient⁶ is shown below and the comparison of both predicted state variable and simulation outcome.

⁶ $R^2 = 1 - \frac{\sum_i (T(u_i) - au_i - b)^2}{\sum_i (T(u_i) - \bar{T})^2}$, \bar{T} is the mean of the sample $\sum_i \frac{T(u_i)}{n}$.

Figure 2.3: Illustrative case where the temperature is affine linear with respect to u .



The figure approves the intuition using simulation data ⁷ at a chosen position and time frame t . It also gives an idea about the proportion of the variation of the dependant variable T_u from the independant variable u i.e coefficient a . More graphs are included in the appendix B.1 chapter with additional comments which are relevant when it comes to chosing the position of the sensors for instance.

The purpose here is merely illustrative in order to justify the assumption that T_u is affine linear wih respect to u when choosing carefully the positon of sensors. With a such assumption we have the property that $DT(u).h$ is depending only on the direction i.e $DT(u).h = DT(v).h = T_v - T_u$ because $DT(u).h = \lim_{\epsilon \rightarrow 0} \frac{T_{u+\epsilon h} - T_u}{\epsilon}$ which can be injected in the the PDE 2.1 with affine linearity with repect to u in the boundary condition and satisfy the PDE below. The coefficient h of the convection will be denoted h_c to not mix up with the direction $h = v - u$.

$$\begin{cases} \rho C_p(DT(u).h) \frac{\partial DT(u).h}{\partial t} - k \nabla^2 DT(u).h = h & (t, x) \in Q \\ -k \frac{\partial DT(u).h}{\partial n} = ah_c h & (t, x) \in \Sigma \\ DT(u).h(0, x) = 0 & x \in \Omega \end{cases} \quad (2.5)$$

Now we proceed as shown above, and we replace $DT(u).h$ by $T(v) - T(u)$ ⁸

⁷The solution of the dual problem in \mathbb{R}^N based on Galerkin approach using ABAQUS software.

⁸Because of the uniqueness of the solution of the PDE with the same initial condition.

in

$$\langle \nabla J(v) - \nabla J(u), h \rangle_{L^2(I', L^2(\Omega))} = \|T_v - T_u\|_{L^2(I', L^2(\Omega))}^2 + \alpha \|h\|_{L^2(I', L^2(\Omega))}^2$$

And we obtain $\langle \nabla J(v) - \nabla J(u), h \rangle_{L^2(I', L^2(\Omega))} \geq \alpha \|h\|_{L^2(I', L^2(\Omega))}^2$, hence J is *alpha*-convex.

Given the assumptions on $u \mapsto T_u$:

- + $u \mapsto T_u$ is **Lipschitz continuous**
- + $u \mapsto T_u$ is **affine linear with respect to u**

and thanks to the theorem 2.1 the optimal control problem 2.2.2 has a unique solution. The solution is characterized with the Euler inequality:

$$\forall v \in U_{ad} \quad \langle \nabla J(u), h \rangle >$$

- **The lower level problem:** We recall the so called follower's problem as follows:

$$\begin{aligned} \text{Minimize } F(X) &= \frac{1}{2} \|HS_G(X) - G\|_{L^2(I, L^2(\Omega))}^2 \\ X &\in X_{ad} \\ G &= \text{Optimal control solution} \end{aligned}$$

We recall that if \vec{z} is the welding direction, then in the local coordinates we have:

$$HS_G = \begin{cases} Q \exp(-(\frac{x}{b})^2 - (\frac{y}{c})^2 - (\frac{z_t}{a_f})^2) & \text{if } z_t > 0 \\ Q \exp(-(\frac{x}{b})^2 - (\frac{y}{c})^2 - (\frac{z_t}{a_r})^2) & \text{else if } z_t \leq 0 \end{cases} \quad (2.6)$$

We recall the feasible set $X_{ad} \in R^4$, and consider the parameters $(Q, b, c, ar) \in X_{ad}$. The speed is considered fixed for this study case and $a_r = 2a_f$ as an assumption from welding experts. Instead of minimizing the $F(X)$ we minimize $\tilde{F}(X) = \frac{1}{2} \|\ln(HS_G(X)) - \ln(G)\|_H^2$. The necessary conditions is that $G! = 0$ and $Q \geq 1$, b, c, a_r are not null. The linearized optimization problem in that case is:

$$\begin{aligned}
& \text{Minimize } \tilde{F}(X) = \frac{1}{2} \|\tilde{Q} + x^2\tilde{b} + y^2\tilde{c} + z^2\tilde{a} - \tilde{G}\|_H^2 \\
& \quad \tilde{X} \in X_{ad} \\
& \quad \tilde{G} = \ln(\text{Optimal control solution})
\end{aligned}$$

With $\tilde{Q} = \ln(Q)$ and $\tilde{b} = \frac{-1}{b^2}$, $\tilde{c} = \frac{-1}{c^2}$ and $\tilde{a} = \frac{-1}{a^2}$, we denote a is a_f or a_r depending on the measurements positions.

In this formulation, \tilde{F} is differentiable and convex as the norm and linear affine maps are differentiable and convex. X_{ad} is considered to be convex and a closed subset of an Hilbert. We apply theorem 2.1 to ensure the solution with its characterisation:

$$\forall X \in \tilde{X}_{ad} < \nabla \tilde{F}(X), h >$$

It is worth noting that the considered Hilbert here is the dual of V 2.3 which is \mathbb{R}^N the descritized space.

Both problems are well paused under the stated conditions, which are mostly related to the choice of sensors/measurements position. The unicity is then depending on the choice of the strict convexity of the cost function.

2.3 ABAQUS/Standard solver

In the area of numerical analysis, Galerkin methods, named after the Russian mathematician Boris Galerkin, convert a continuous operator problem, such as a nonlinear heat equation, commonly in a weak formulation 2.2, to a discrete problem by applying linear constraints determined by finite sets of basis functions 2.1.1. It is to mention that we are dealing with finite-dimensional problem which is already discritized in time and finite elements space domains using ABAQUS/Standard solver⁹ and proving the convergence of such discritisation is beyond the scope of the thesis. In the meantime, the article [30] about nonlinear Galerkin methods gives a good mathematical background in the context of approximating dissipative partial differential equations. The focus of this chapter is rather on how the ABAQUS/Standard solver solves the finite discritized problem.

⁹One of the products of ABAQUS FEA that employs an implicit integration scheme.

2.3.1 Discretized problem

In abstract way we consider a finite dimensional subspace $V_h^{10} \subset V^{11}$ and we apply the usual Galerkin approximation. We solve the variational formulation in V_h :

$$\int_{V_h} \rho C_p(T_h) \frac{dT_h(t, x)}{dt} \Phi_h + \int_{V_h} k \nabla T_h(t, x) \nabla \Phi_h = \int_{V_h} G(t, x) \Phi_h + \int_{\Sigma_h} Loss(T_h) \Phi_h \quad (2.7)$$

Notice that the equation has remained unchanged and only the spaces have changed. Reducing the problem to a finite-dimensional vector subspace allows us to numerically compute T_h as a finite linear combination of the basis vectors in V_h . where Φ_h is an arbitrary variational field (test functions) satisfying the essential boundary conditions. The body is approximated geometrically with finite elements (C3D8 in our case), so the temperature is interpolated as:

$$T = N^N(x) T^N$$

Where $N^N(x)$ are the nodal interpolation functions and T^N are nodal temperatures. The Galerkin approach assumes that the variational field Φ , is interpolated by the same functions.

$$\Phi = N^N(x) \Phi^N$$

According to ABAQUS/Standard documentation [7], the first-order heat transfer elements¹² use a numerical integration rule with the integration stations located at the corners of the element for the heat capacitance terms. This means that the Jacobian term associated with the internal energy rate is diagonal. The second-order elements use conventional Gaussian integration. Thus, second-order elements are to be preferred for problems when the solution will be smooth (without latent heat effects), whereas the first-order elements should be used in nonsmooth cases (with latent heat).

First and second order polynomials in all dimensions are used for the the interpolation function as described above. With these interpolations the variational statement 2.7

¹⁰The key property of the Galerkin approach is that the error is minimized in the chosen subspaces on 8-node brick element e.g.C3D8.

¹¹ V is Hilbert space introduced in 2.3

¹²Such as 2-node link, 4-node quadrilateral, and 8-node brick.

becomes:

$$\Phi^N \left\{ \int_{V_h} \rho C_p(T_h) \frac{dT^N}{dt} N(x) + \int_{V_h} \nabla T^N \cdot k \cdot \nabla N^N(x) = \int_{V_h} N^N(x) G(t, x) + \int_{\Sigma_h} N^N(x) Loss(T^N) \right\} \quad (2.8)$$

And since Φ^N the are test functions, this leads to the following:

$$\int_{V_h} \rho C_p(T_h) \frac{dT^N}{dt} N(x) + \int_{V_h} \nabla T^N \cdot k \cdot \nabla N^N(x) = \int_{V_h} N^N(x) G(t, x) + \int_{\Sigma_h} N^N(x) Loss(T^N) \quad (2.9)$$

The nodal assembled matricial representation of which the uniqueness is guaranteed by Riesz theorem 2.1.1 can be denoted as follows:

$$C^N \left(\frac{dT^N}{dt} \right) - I^N = G^N + Loss^N \quad (2.10)$$

Where:

- Nodal interal energy with $C^N = \int_{V_h} \rho C_p(T_h)$
- Nodal internal load related to conductive energy with $I = - \int_{V_h} \nabla T^N \cdot k \cdot \nabla N^N(x)$
- Nodal heat source load with $G^N = \int_{V_h} N^N(x) G(t, x)$
- Nodal losses with $Loss^N = \int_{\Sigma_h} N^N(x) Loss(T^N)$

This set of equations is still continuous with respect to time.

2.3.2 Implicite scheme

Abaqus/Standard uses the backward difference algorithm:

$$\frac{dT}{dt} = \frac{T_{t+\Delta t} - T_t}{\Delta t} \quad (2.11)$$

This originally coming from taylor developement of the map $t \mapsto T(t)$ given $\lambda \in \mathbb{R}$:

$$\begin{cases} (1 - \lambda) \{ T_{t+\Delta t} = T_t + \Delta t T_t + o(\Delta t) \} \\ \lambda \{ T_t = T_{t+\Delta t} - \Delta t T_{t+\Delta t} + o(\Delta t) \} \end{cases} \quad (2.12)$$

Then we add the equation of set above to each other and we obtain a more general Euler backward discretisation depending on λ :

$$T_{t+\Delta t} = T_t + \left(\lambda \frac{dT_{t+\Delta t}}{dt} + (1 - \lambda) \frac{dT_t}{dt} \right) \Delta t + o(\Delta t) \quad (2.13)$$

Notice that the backward difference above 4.3 is when $\lambda = \frac{1}{2}$. The choice of ABAQUS/-Standard solver for such schemes is because their simplicity in implementation (for example, no special starting procedures are needed) and well-understood behavior.

According to ABAQUS documentation, for $\lambda < \frac{1}{2}$ such operators are only conditionally stable for linear heat transfer problems. So, ABAQUS prefers using unconditionally stable methods, because it is most common deal with nonlinear transient problems with very long time periods (compared to the stability limit for the explicit form of the operator i.e. $\lambda = 0$). Thus ABAQUS $\lambda \geq \frac{1}{2}$. ABAQUS also consider that $\lambda = \frac{1}{2}$ has the highest accuracy but tending to produce oscillations at the start and hence the solver goes with $\lambda = 1$ [7].

The nodal set of equations becomes:

$$C^N \left(\frac{T_{t+\Delta t}^N - T_t^N}{2 * \Delta t} \right) = I^N + G^N + Loss^N \quad (2.14)$$

And the matical representation is:

$$C @ \left(\frac{T_{t+\Delta t} - T_t}{2 * \Delta t} \right) = I + G + Loss \quad (2.15)$$

2.3.3 Newton-Raphson method

The Newton-Raphson method, or Newton Method, is a numerical method for solving equations. Like so much of the differential calculus, it is based on the simple idea of linear approximation. We will be explaining this method based on the documentation of ABAQUS/Standad solver.

In the previous sebsection we ended with the matricial representation for each moment t 2.15. Which physically, a set of equilibrium equations obtained by discretizing

in space and time the nonlinear heat transfer equation. It can be denoted as follows:

$$F(T_t) = 0 \quad (2.16)$$

Where $F(T_t) = C @ \left(\frac{T_{t+\Delta t} - T_t}{2 * \Delta t} \right) - I - G - Loss$ and $@$ denoted as the matricial product.

ABAQUS/Standard uses Newton's method as a numerical technique for solving the nonlinear equilibrium equations. The motivation for this choice is primarily the convergence rate obtained by using Newton's method compared to the convergence rates exhibited by alternate methods¹³ for the types of nonlinear problems most often studied with ABAQUS. The basic formalism of Newton's method is as follows. Assume that, after an iteration i , an approximation T_t^i , to the solution has been obtained. Let $h_{i+1} = T - T_t^i$ be the difference between this solution and the exact solution which is supposed to be T_t^{i+1} to the discrete equilibrium equation Equation 2.16. This means that:

$$F(T_t^i + h_{i+1}) = 0$$

Expanding the left-hand side of this equation in a Taylor series (with an error term) about the approximate solution T_t^i then gives:

$$F(T_t) \approx F(T_t^i + h_{i+1}) - h_{i+1} @ \nabla F(h_{i+1}) - h_{i+1}^2 @ \nabla^2 F(h_{i+1}) \quad (2.17)$$

Which is equivalent to writing:

$$F(T_t) + h_{i+1} @ \nabla F(h_{i+1}) + h_{i+1}^2 @ \nabla^2 F(h_{i+1}) \approx 0 \quad (2.18)$$

If T_t^i is a close approximation to the solution, the magnitude of each h_{i+1} will be small, and so all but the first two terms above can be neglected giving a linear system of equations:

$$K @ h_{i+1} \approx -F \quad (2.19)$$

Where the Jacobian matrix or denoted Stiffness by ABAQUS/Standard is calculated

¹³Usually modified Newton or quasi-Newton methods

as follows:

$$K = \nabla F(h_{i+1}) = \nabla \left[C @ \left(\frac{T_{t+\Delta t} - T_t}{2 * \Delta t} \right) - I - G - Loss \right]$$

\iff

$$K = \nabla \left[\frac{C}{2 * \Delta t} - I - Loss \right]$$

Because the heat source G is temperature-independant.

The Newton's method consist thus on iterating on i until h_n is almost null given a tolerance by the user.

Convergence of Newton's method is best measured by ensuring that all entries in F_i and all entries in h_{i+1} are sufficiently small. Both these criteria are checked by default in an ABAQUS/Standard solution. ABAQUS/Standard also prints peak values in the force residuals, incremental displacements, and corrections to the incremental displacements at each iteration so that the user can check for these contingencies himself. Those informations are found in file with extention `.stat`.

According to ABAQUS documentation, Newton's method is usually avoided in large finite element codes, apparently for two reasons. First, the complete Jacobian matrix is sometimes difficult to formulate; and for some problems it can be impossible to obtain this matrix in closed form, so it must be calculated numerically—an expensive (and not always reliable) process. Secondly, the method is expensive per iteration, because the Jacobian must be formed and solved at each iteration. The most commonly used alternative to Newton is the modified Newton method, in which the Jacobian in Equation 2.19 is recalculated only occasionally (or not at all, as in the initial strain method of simple contained plasticity problems). This method is attractive for mildly nonlinear problems involving softening behavior (such as contained plasticity with monotonic straining) but is not suitable for severely nonlinear cases. (In some cases ABAQUS/Standard uses an approximate Newton method if it is either not able to compute the exact Jacobian matrix or if an approximation would result in a quicker total solution time. For example, several of the models in ABAQUS/Standard result in a nonsymmetric Jacobian matrix, but the user is allowed to choose a symmetric approximation to the Jacobian on the grounds that the resulting modified Newton method converges quite well and that the extra cost of solving the full nonsymmetric system does not justify the savings in itera-

tion achieved by the quadratic convergence of the full Newton method. In other cases the user is allowed to drop interfield coupling terms in coupled procedures for similar reasons.)

Now that we understood how the ABAQUS/Standard solver works, we will be using this theory in order to imitate the same solver so we can have comparable results with ABAQUS/Standard. The idea is to transfer the information already contained in the matrices C and K and the load vectors I , G and $Loss$ into a reduced accessible model which can be solved without assembling the matrices for each iteration/simulation. This where machine learning is contributing. We can thus apply the theory of optimal control on a non linear ordinary differential equation to solve the inverse problem. More details about this proposed solution can be found in the white box chapter [4](#).

Chapter 3

Black box approach

3.1 Approximate gradient-based method

In this section, we will be implementing the gradient descent to determine the parameters of the heat source model. The figure below illustrate the approach and symbolize solving the forward problem as black box as long as it is solver by ABAQUS/Standard solver.

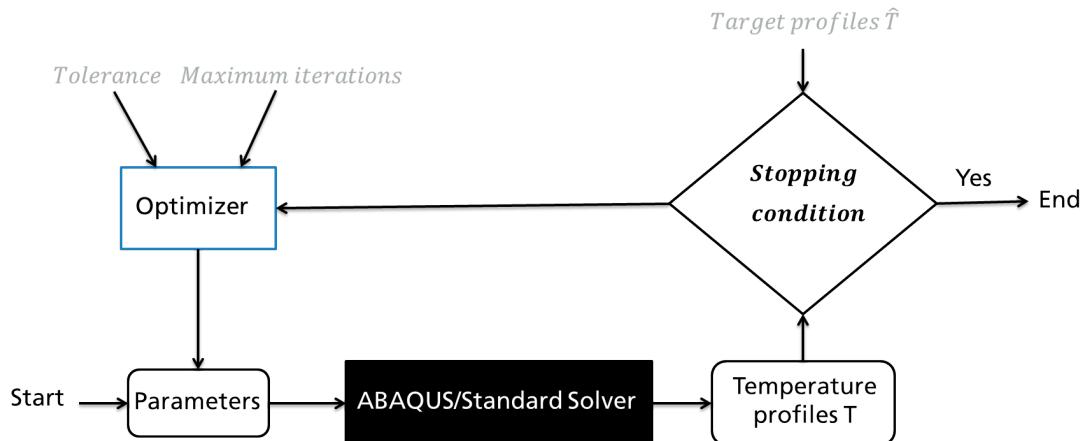


Figure 3.1: A schema on the proposed black box based optimisation tool to solve the inverse problem.

The adopted optimizer is Gradient descent method.

3.1.1 Gradient descent

By definition, the Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. The idea is to take repeated steps in the opposite direction of the gradient (or approximate gradient) of the function at the current point, because this is the direction of steepest descent.

Let $J : X_{ad} \mapsto \mathbf{R}$ differentiable, and $X_{ad} \subset \mathbf{R}^d$ where d is the dimension of the parameters to be optimized. We would like to solve the problem:

$$\text{Minimize } J(X)$$

$$X \in X_{ad}$$

Let $X^{k+1} = X^k + \rho^k d^k \in X_{ad}$. The Taylor series at X^{k+1} when ρ^k is small enough is :

$$J(X^{k+1}) \approx J(X^k + \rho^k d^k) = J(X^k) + \rho^k \langle \nabla J(X^k), d^k \rangle_{\mathbf{R}^d}$$

We wish to have at each iteration a lower value of the cost function J . Which equivalent to:

$$J(X^{k+1}) < J(X^k)$$

So if we choose $d^k = -\nabla J(X^k)$ and we take the projection so that $X^k + \rho^k d^k \in X_{ad}$, we will obtain:

$$J(X^{k+1}) - J(X^k) \approx -\rho^k (\nabla J(X^k))^2 < 0$$

The method of Gradient descent takes its name from choosing the descent direction

$$d^k = -\nabla J(X^k)$$

The gradient descent algorithm is as follows:

```

let  $k = 0$ 
choose  $X^0$ 
while( $\|X^{k+1} - X^k\| \geq \epsilon$ ) and ( $k \leq k_{max}$ ):
    calculate  $d^k = -\nabla J(X^k)$ 
    calculate  $\rho^k$ 
    let  $X^{k+1} = X^k + \rho^k d^k$ 
end while

```

It is worth mentionning that the gradient descent algorithm finds local solutions. However, when we have strict convexity of J , the local solution is global. Yet, numerically and with approximated gradient, the uniqueness is depending on how smooth the cost function is.

In this chapter, the lower and upper problems (see section 2.2.2) are appraoched as one single optimization problem. And we chose the cost function to be:

$$J(X) = \frac{1}{2} \sum_i^m \|T - \hat{T}\|_{R^n}^2 + \frac{\gamma}{2} \|G(X)\|_{R^n}^2$$

Where γ is for both regularisation¹ and penalisation to minimize the energy/power. The number of measurements² is denoted m . The time dependant temperature $T \in \mathbf{R}^n$ is calculated by ABAQUS/Standard solver for each iteration. While $\hat{T} \in \mathbf{R}^n$ is the time dependant target data (given). And n is the number of frames (discirtisation in time).

And as defined in 4.5, $X = \begin{pmatrix} Q \\ b \\ c \\ a_r(a_f) \end{pmatrix}$ are the parameters of the heat source power density $G(X)$. This latter is expressed as:

$$G(X) = \begin{cases} Q \exp(-(\frac{x}{b})^2 - (\frac{y}{c})^2 - (\frac{z_t}{a_f})^2) & \text{if } z_t > 0 \\ Q \exp(-(\frac{x}{b})^2 - (\frac{y}{c})^2 - (\frac{z_t}{a_r})^2) & \text{else if } z_t \leq 0 \end{cases} \quad (3.1)$$

¹e.g. to have strict convexity or avoid high oscilations especially for data with strong variation e.g. noise)

²Number of temperature profiles which is either numeric or experimental.

We define X_{ad} as $[Q_{min}, Q_{max}] \times [b_{min}, b_{max}] \times [c_{min}, c_{max}] \times [a_{min}, a_{max}] \subset \mathbb{R}_+^4$. We recall that the x, y, z in the $G(x)$ are local coordinates and a denotes both a_f or a_r ($a_r = 2a_f$), which is depending on the time t , the speed v and global coordinates (Z) of the measurements.

For such branch of optimizers one would need to calculate the gradient $\nabla J(X)$. Hence, it is necessary to prove first that the cost function J is differentiable with respect to X .

We consider:

$$\begin{aligned} J : X \in X_{ad} \subset \mathbb{R}^d &\xrightarrow{J_1} \mathbb{R}^n \xrightarrow{J_2} \mathbb{R}^n \xrightarrow{J_3} \mathbb{R}^1 \\ J_1(x) &= G(x) \\ J_2(y) &= T_y = \alpha_t y + \beta_t \\ J_3(\cdot) &= \|\cdot\|_2 \end{aligned}$$

Given that the sum of differential functions or/and their composition is differentiable. The differentiability of J depends then on the differentiability of J_1, J_2, J_3 which are differentiable. In fact, the norm, a linear-affine map and $X \mapsto G(X)$ ³ are differentiable.

How to calculate the gradient $\nabla J(X)$?

Numerical differentiation approximation:

A simple two-point estimation is to compute the slope of a nearby secant line through the points $(X, J(X))$ and $(X + h, J(X + h))$. Choosing a small number h , h represents a small change in X , and it can be either positive or negative. The slope of this line is:

$$\frac{J(X + h) - J(X)}{h}$$

The slope of this secant line differs from the slope of the tangent line by an amount that is approximately proportional to h . As h approaches zero, the slope of the secant line approaches the slope of the tangent line. In other words, also by Taylor expansion:

$$\nabla J(X) \lim_{h \rightarrow 0} \frac{J(X + h) - J(X)}{h}$$

³ $\forall t \in I G'(z = 0^-) = G'(z = 0^+) = G'(z = 0) = 0$ because $\lim_{x \rightarrow +\infty} xe^{-x} = 0$

Machine learning based approximation: We will calculate the gradient based on the assumption $T_t = \alpha_t G(X) + \beta_t$ and by identifying the gradient $\nabla J(X)$ as Riesz representation 2.1.1 of a linear continuous map

$$DJ(X).h = \langle \nabla J(X), h \rangle_{\mathbb{R}^d}$$

We recall:

$$J(X) = \frac{1}{2m} \sum_i^m \|T - \hat{T}\|_{R^n}^2 + \frac{\gamma}{2} \|G(X)\|_{R^n}^2$$

We calculate the derivative of $J(X)$ in h direction based on the chain rule:

$$\nabla J(X).h = \frac{1}{m} \left\langle \sum_i^m (T - \hat{T}), \nabla T(G(X)) \nabla G(X).h \right\rangle_{R^n} + \gamma \left\langle G(X), \nabla G(X).h \right\rangle_{R^n}$$

Assumption of linear-affinity of T with respect to G ($\exists \alpha_t, \beta_t T_G = \alpha_t G + \beta_t$) holds 2.2.2)

$$\implies \nabla J(X).h =$$

$$\frac{1}{m} \left\langle \sum_i^m (T - \hat{T}), \begin{pmatrix} \alpha_{t=1} \\ \vdots \\ \alpha_{t=n} \end{pmatrix} \right\rangle_{R^n} . \left\langle \nabla G(X), h \right\rangle_{R^n} + \gamma \left\langle G(X), \begin{pmatrix} 1_{t=1} \\ \vdots \\ 1_{t=n} \end{pmatrix} \right\rangle_{R^n} . \left\langle \nabla G(X), h \right\rangle_{R^n}$$

Which means that the gradient

$$\nabla J(X) = \nabla G(X) \left[\frac{1}{m} \left\langle \sum_i^m (T - \hat{T}), \begin{pmatrix} \alpha_{t=1} \\ \vdots \\ \alpha_{t=n} \end{pmatrix} \right\rangle_{R^n} + \gamma \left\langle G(X), \begin{pmatrix} 1_{t=1} \\ \vdots \\ 1_{t=n} \end{pmatrix} \right\rangle_{R^n} \right]$$

3.1.2 Implementation and results

The first approximation of the gradient is adopted because of its applicability to different models without making any assumption on the variation of T with respect to the volumetric heat source power.

Description: The following flowchart gives an overview on the implementation of the optimization tool including synchronization with both the optimizer and the ABAQUS/-Standard solver.

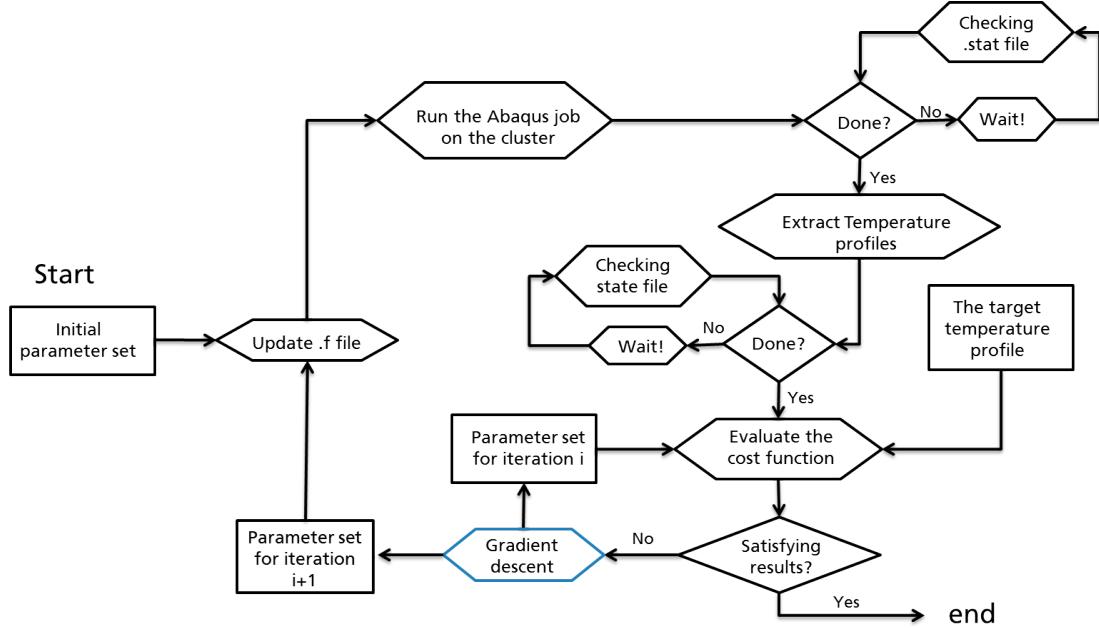


Figure 3.2: An overview of developed parameter identification tool.

First, the user provides the initial guess of the parameters to be identify, then the tool take the lead and start the iteration by modifying the user subroutine file (.f) and running the job, waiting, postprocessing the FEA output (.odb), and follow the steepest descent for the next iteration based on the calculated step ρ .

How to calculate the step?

Armijo–Goldstein condition:

Given a starting position X^k and a search direction $-\nabla J(X^k)$, the task of a line search is to determine a step size $\rho^k > 0$ that reduces $J(X^{k+1})$ relative to $J(X^k)$. One approach is to find an optimal step using root finding algorithms such as Bisection method⁴ or Newton’s method 2.19. However, it is undesirable to devote substantial resources to finding a value of ρ^k to precisely minimize J . Especially, that the ABAQUS/Standard solver might take up to weeks depending on the complexity of the FEA. For such reason, finding ρ^k in one particular direction could instead be employed to identify a better search direction. The goal, then, is just to identify a value of ρ^k that provides a reasonable amount of improvement in the objective function J , rather than to find the optimal step. The backtracking line search starts with a large estimate of ρ^k and iteratively

⁴The method is also called the interval halving method, the binary search method, or the dichotomy method [37]

shrinks it. The shrinking continues until a value is found that is small enough to provide a decrease in the objective function that adequately matches the decrease that is expected to be achieved, based on the local function gradient $\nabla J(X^k)$.

Practically, we define the local slope of the function of ρ^k along the direction $-\nabla J(X^k)$ as $m = -\langle \nabla J(X^k), \nabla J(X^k) \rangle$ ($m < 0$). Based on a selected control parameter $c \in [0, 1]$ and the shrinking rate $\tau \in]0, 1[$, the Armijo–Goldstein condition tests whether a step-wise movement from a current position X^k to a modified position $X^{k+1} = X^k - \rho^k \nabla J(X^k)$ achieves an adequately corresponding decrease in the objective function. The condition is fulfilled, according to [12], if:

$$J(X^k - \rho^k \nabla J(X^k)) - J(X^k) \leq \tau \rho^k m \quad (3.2)$$

Thus, the backtracking line search strategy starts with a relatively large step size, and repeatedly shrinks it by a factor τ until the Armijo–Goldstein condition is fulfilled. The following figure represents the implemented backtracking line search with $c = 1$ and $t = -cm$.

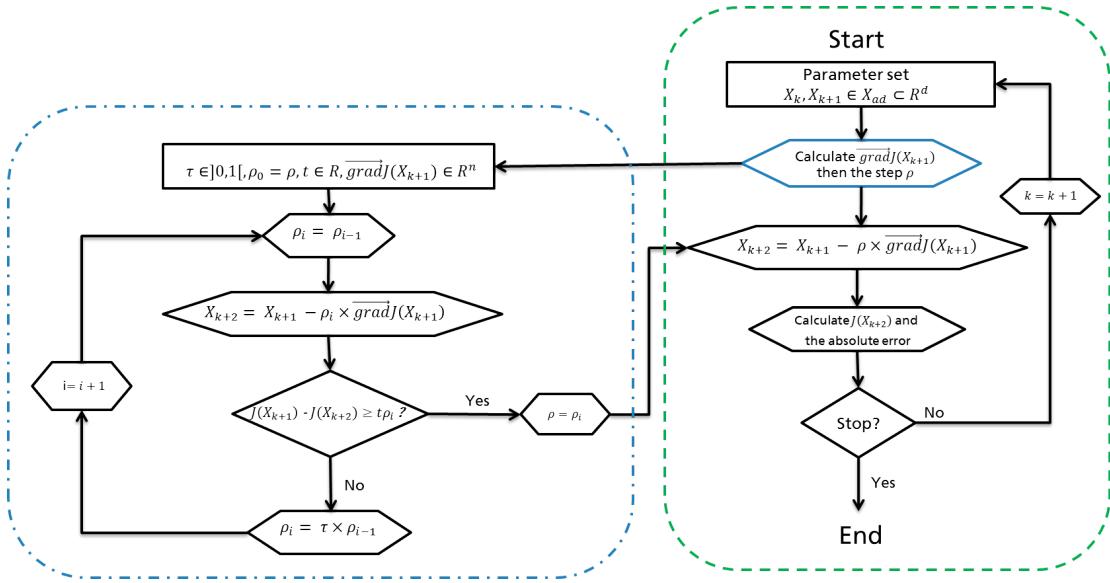


Figure 3.3: Gradient descent method (boxed in green) and the backtracking line search (boxed in blue).

Convergence:

The convergence of gradient descent is depending on the following:

- Convexity⁵
- Smoothness⁶

Theoretically, according to [23], the average iterates is defined as follows:

Theorem 3.1.1: *Average iterates N:*

Let f be convex and L -smooth and let x^k for $k = 1, \dots, n$ be the sequence of iterates generated by the gradient method. It follows that:

$$\frac{1}{N} \sum_{k=1}^N [f(x^k) - f(x^*)] \leq \frac{\rho^k}{4L} \frac{f(x^1) - f(x^*)}{N} + \frac{1}{\rho^k} \frac{\|x^1 - x^*\|^2}{N}$$

And the convergence rate of gradient descent for J (convex) is $o(\frac{1}{N})$ ⁷, where N is the number of iterations. This implies that in order to achieve a bound of $J(x^k) - J(x^*) \leq \epsilon$, we must run $o(\frac{1}{\epsilon})$ iterations. In case of strong convexity we have the following theorem [23]

Theorem 3.1.2: *In case f is L -smooth and α -convex (strongly convex), from a given $x^0 \in R^d$ and $\frac{1}{L} \geq \rho > 0$,*

$$x^{k+1} = x^k - \rho \nabla f(x^k)$$

$$\|x^{k+1} - x^*\|^2 \leq (1 - \rho\alpha) \|x^0 - x^*\|^2$$

The convergence rate is $o(\alpha\rho)$.

Study case:

We recall the explicit established formula of the gradient:

$$\nabla J(X) = \nabla G(X) \left[\frac{1}{m} < \sum_i^m (T - \hat{T}), \begin{pmatrix} \alpha_{t=1} \\ \vdots \\ \alpha_{t=n} \end{pmatrix} \right] >_{R^n} + \gamma < G(X), \begin{pmatrix} 1_{t=1} \\ \vdots \\ 1_{t=n} \end{pmatrix} \right] >_{R^n}$$

The strict convexity is clear when considering only the maximum power intensity as

⁵See section 2.2

⁶A differential function f is said to be L -smooth if its gradients are Lipschitz continuous, that is $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.

⁷Little o notation is used to describe an upper bound that cannot be tight.

parameter to identify i.e. $X = Q$. In fact,

$$\nabla G(X) = e^{-\left(\frac{x}{b}\right)^2 - \left(\frac{y}{c}\right)^2 - \left(\frac{z_t}{a_r}\right)^2}$$

Which does not depend on X . Which means that the Hessian:

$$Hess J(X) = \nabla^2 J(X) = \gamma < (\nabla G(X))^2, \begin{pmatrix} 1_{t=1} \\ \vdots \\ 1_{t=n} \end{pmatrix} >_{R^n}$$

is definite positive (i.e. $X^T(\nabla G(X))X > 0$) An illustration of the strict convexity with respect to $X = Q$ is shown in the figure below.

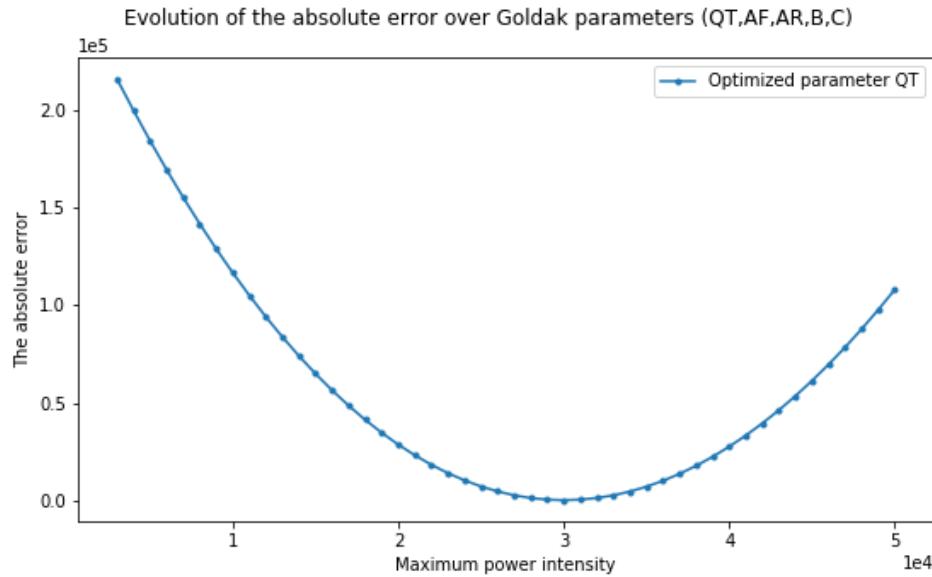


Figure 3.4: An illustration of the convexity of the cost function with respect to the maximum intensity Q .

With the same assumptions presented in 2.2.2:

- + $G \mapsto T_G$ is Lipschitz continuous
- + $G \mapsto T_G$ is affine linear with respect to \mathbf{u}

We can substitute $T_{G(X_1)} - T_{G(X_1)}$ directly in $\langle \nabla J(X_1) - \nabla J(X_2), h \rangle$ formula and conclude that:

$$\langle \nabla J(X_1) - \nabla J(X_2), h \rangle \geq \alpha \|X_1 - X_2\|_2^2$$

The figure 3.5 illustrate the non strict convexity with respect to b , c and both a_f and a_r . In fact, the made assumption of linear-affinity of the temperature with respect to the heat source intensity is not a certainty. Having material properties which depend on tempererute does not allow such linearity to be true. Raising the regulation parameter might be a solution to the convexity issue.

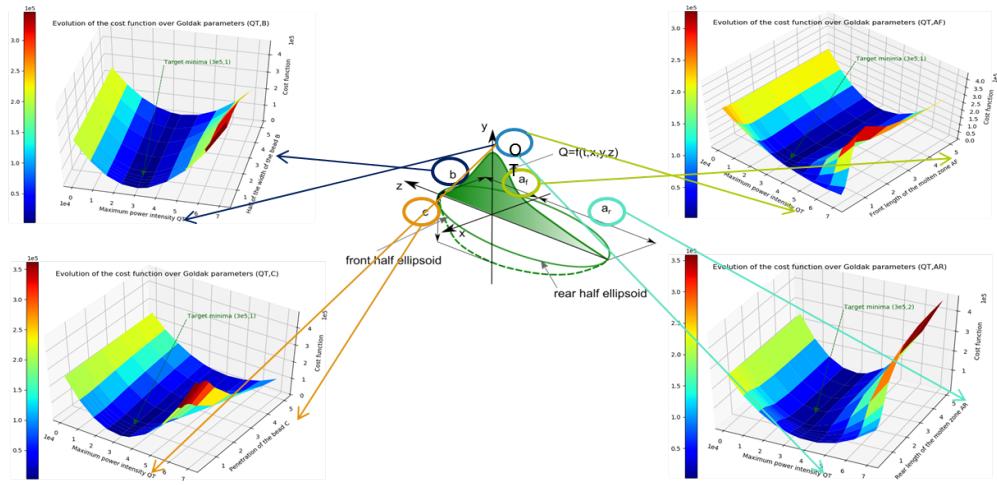


Figure 3.5: An illustration of the cost function convexity with respect to the identifica-tion of 2D heat source paramters.

The figure 3.6 represents the numerical measurement positions (left) and the target profiles (right).

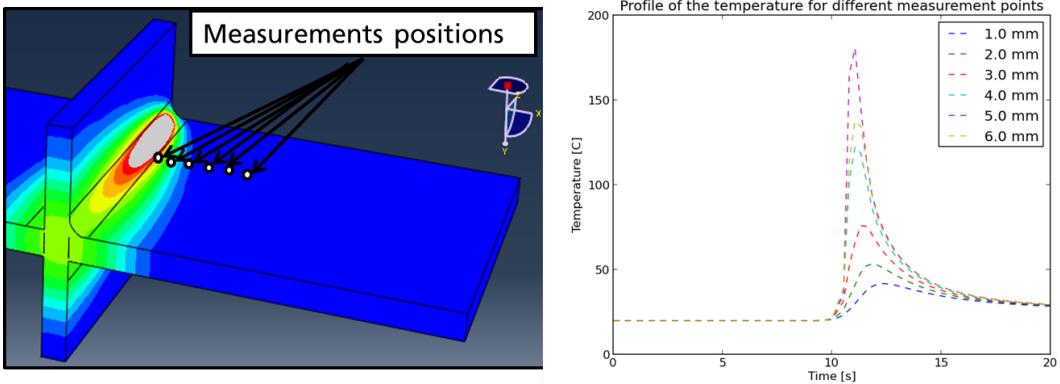


Figure 3.6: An illustration of both measurements positions (left) and numerical target profiles (right).

Numerical experiment to identify the maximum power density Q:

The choice of target profiles is exemplary and the purpose is to simulate the parameter identification based on unknown target profiles. After the phase of numerical experiments and the tuning of the hyperparameters such us γ a Lab measurement is fed to the optimization tool in order to verify on realistic data.

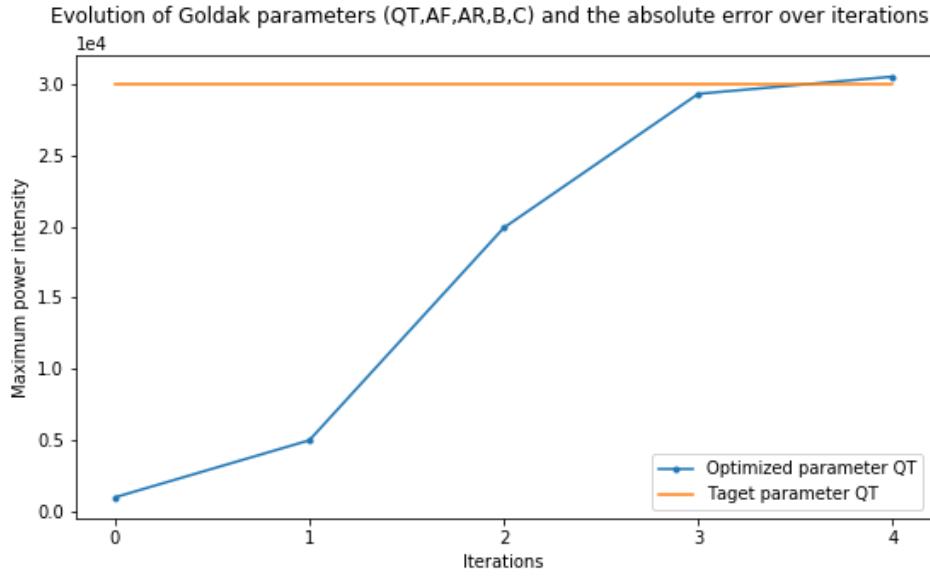


Figure 3.7: The result of the gradient descent tool to identify the maximum heat source intensity Q .

The step computing is not displayed. The stopping criteria is the square absolute error $\sum_{t=i}^{76} \|T - \hat{T}\|_2^2 < 1e-1$.

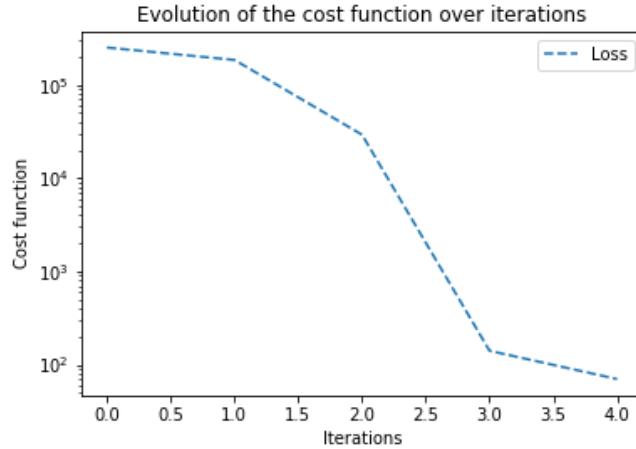


Figure 3.8: The result of the gradient descent tool to identify the maximum heat source intensity Q .

Numerical experiment to identify the five parameters at once:

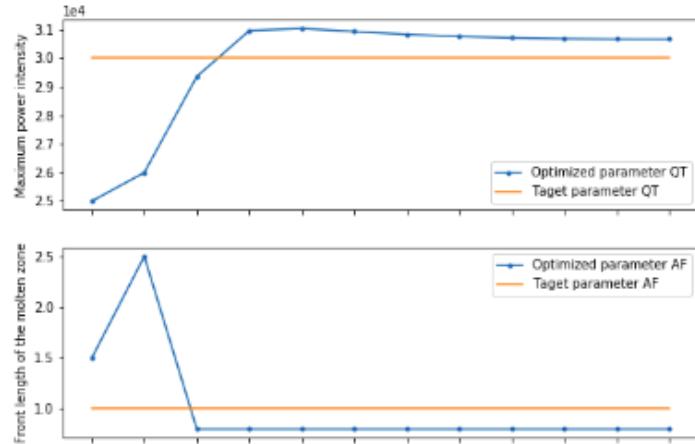


Figure 3.9: The result of the gradient descent tool to identify all parameters (part 1 of the results).

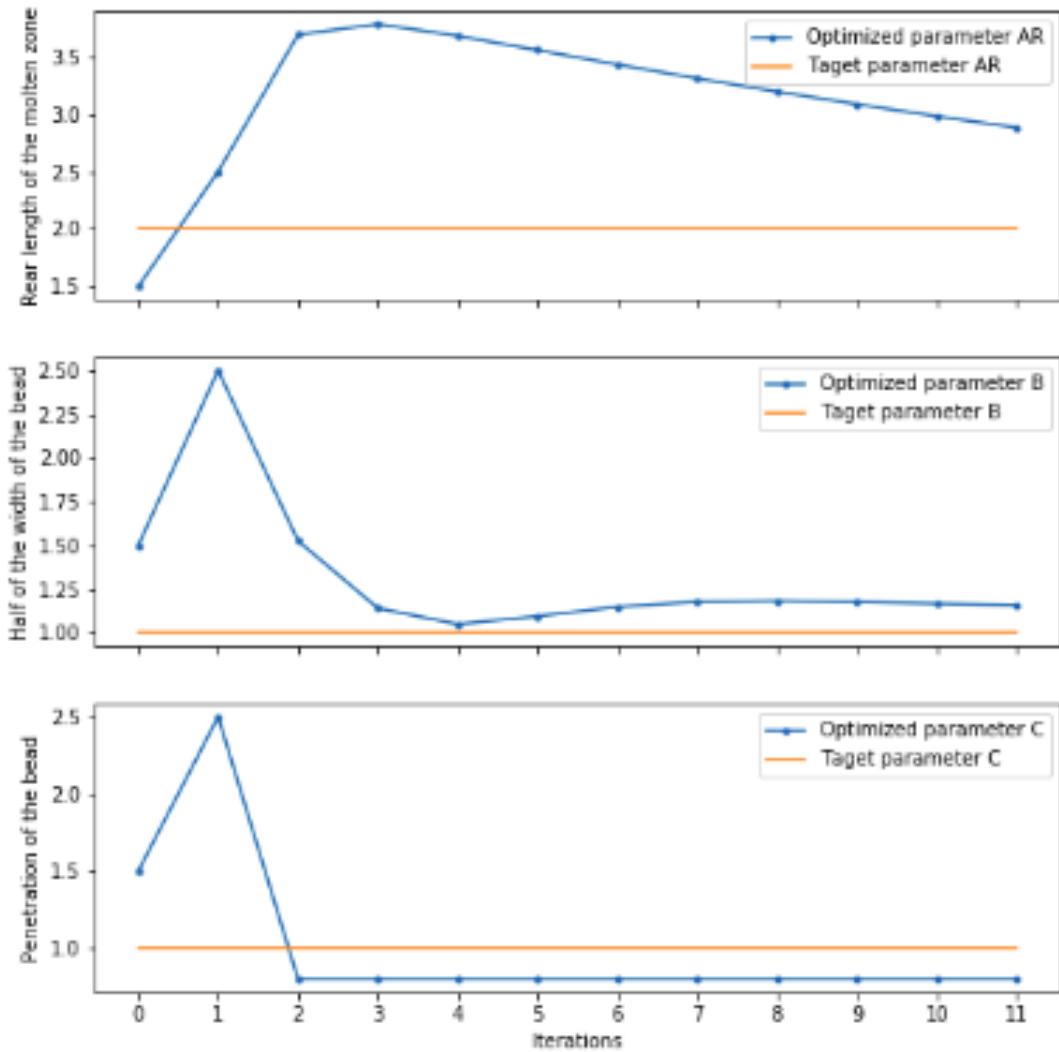


Figure 3.10: The result of the gradient descent tool to identify all parameters (part 2 of the results).

The stopping criteria is the square absolute error $\sum_{t=i}^{76} \|T - \hat{T}\|_2^2 < 1e-1$. In realistic problem with a more noisy data, it would challenging or even impossible to reach such precision. In order to have an idea on the chosen criteria, in terms of average temperature deviation it is equivalent to have an absolute deviation of $\sim 3.610^{-2}C$ per frame. The regulation coefficients equal $1e-18$ because the numerical data are smooth and no penalisation was involved.

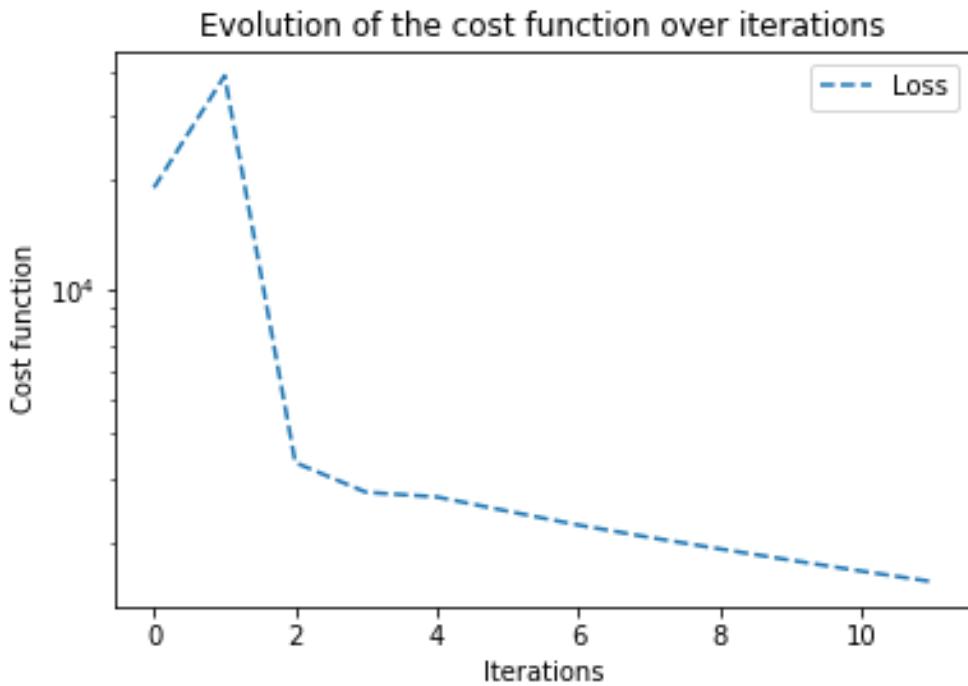


Figure 3.11: The loss function for the identification of all parameters.

More experiments are available in the appendix [C.1](#).

3.2 Multilayer perceptron neural network

By definition, a multilayer perceptron (MLP) is a class of feedforward artificial neural network. A MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. In this section we propose a solution for both forward and inverse problems. In the forward problem the mean squared error is the predicted variable (output) and we feed heat source parameters as training data. In the inverse problem, the mean squared error of the measurements profiles are fed as a single input and the predicted variable are in this case the heat source parameters. They are recommended in the following problems:

+ Tabular datasets

- + Classification prediction problems

- + Regression prediction problems

3.2.1 Data collection and processing

In order to train the MLP model, 512 ABAQUS FEA simulations were run for three parameter constellations. Q, B and C were varied in the respective intervals:

- [10e3, 70e3]
- [1,5]
- [1,5]

The data was then normalized and shuffled which serves the purpose of reducing variance and making sure that models remain general and overfit less. The figure 3.13 illustrates the distribution of the collected data.

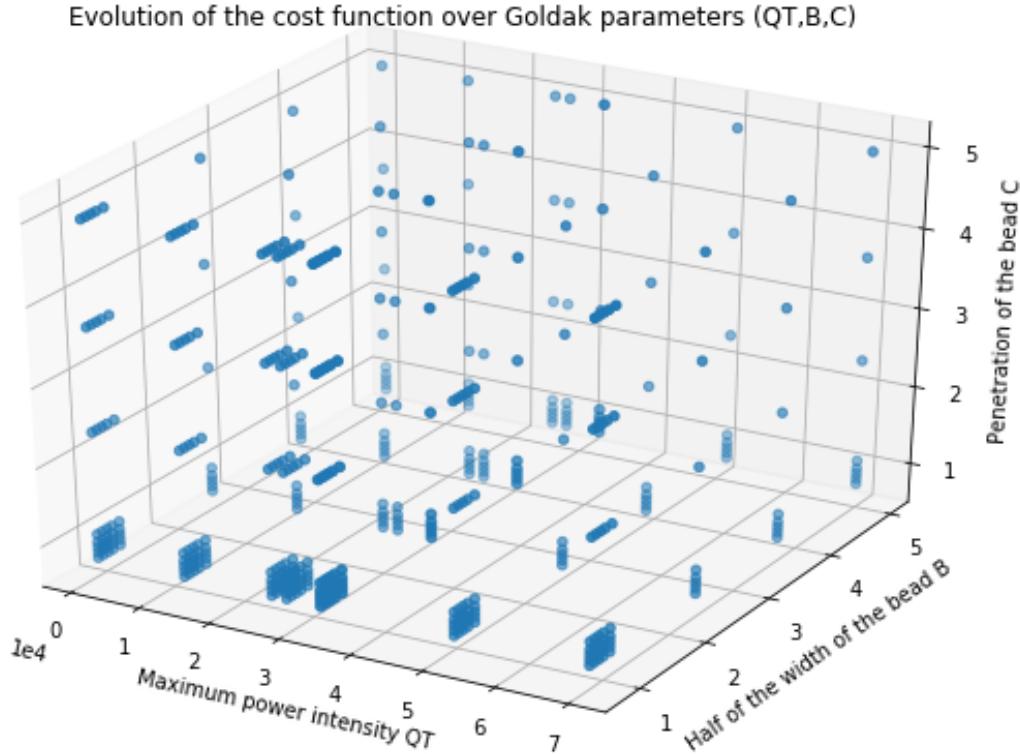


Figure 3.12: Collected data for training the MLP model.

3.2.2 Forwad problem

The proposed architecture for the direct problem is summarized in the figure ??.

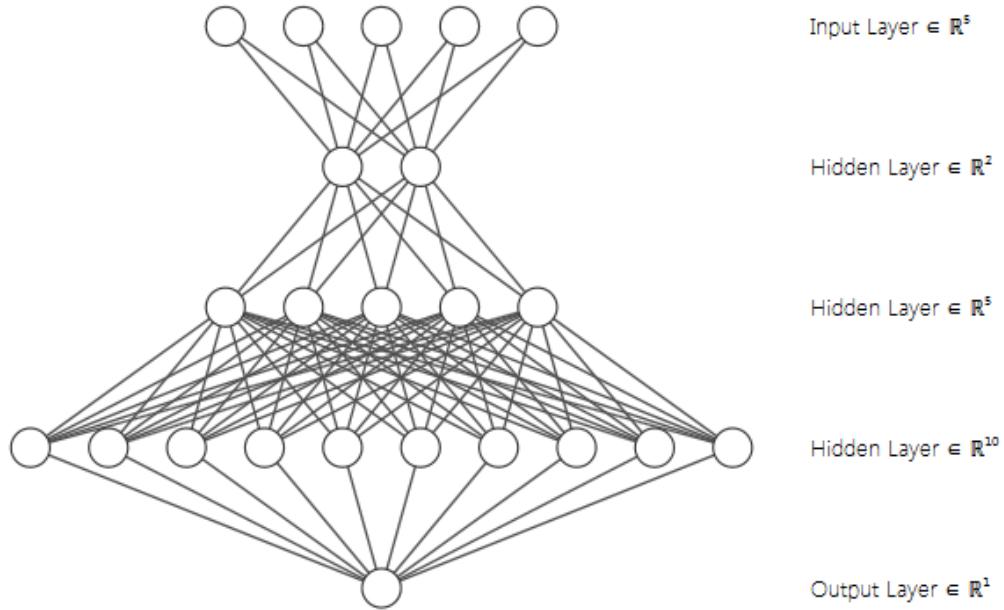


Figure 3.13: Proposed MLP architecture.

The model has the following characteristics:

- **Metric:** mean squared error
- **Activation functions:** RELU
- **Kernal initializer:** "he uniform"
- **Optimizer:** ADAM(0.001)
- **Epochs, Batch size = 120,5**

At the training stage the figure 3.14 approves the choice of the learning rate.

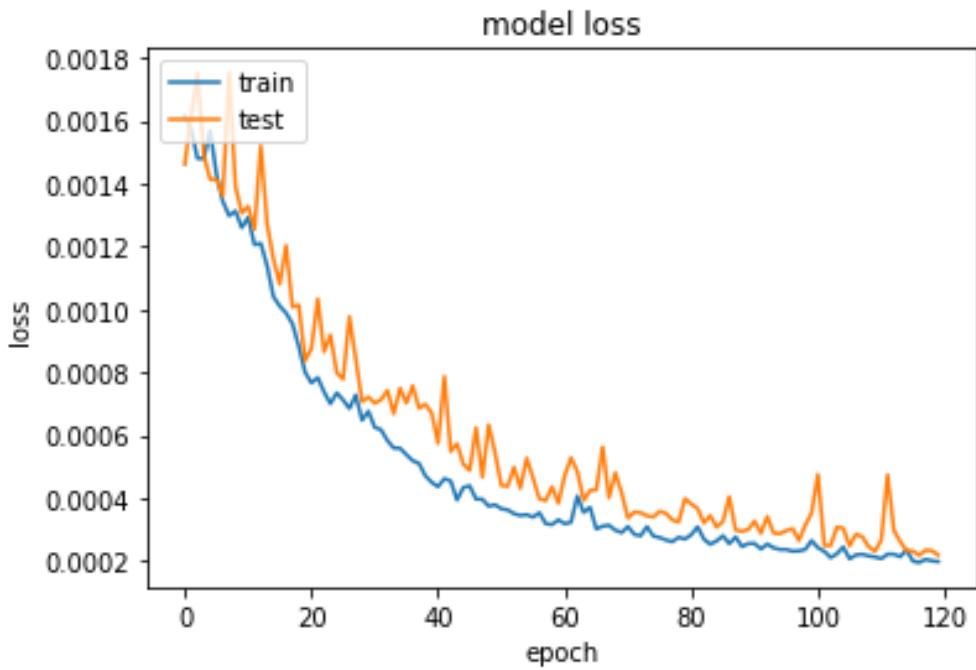


Figure 3.14: The loss function during the training phase.

The figure 3.15

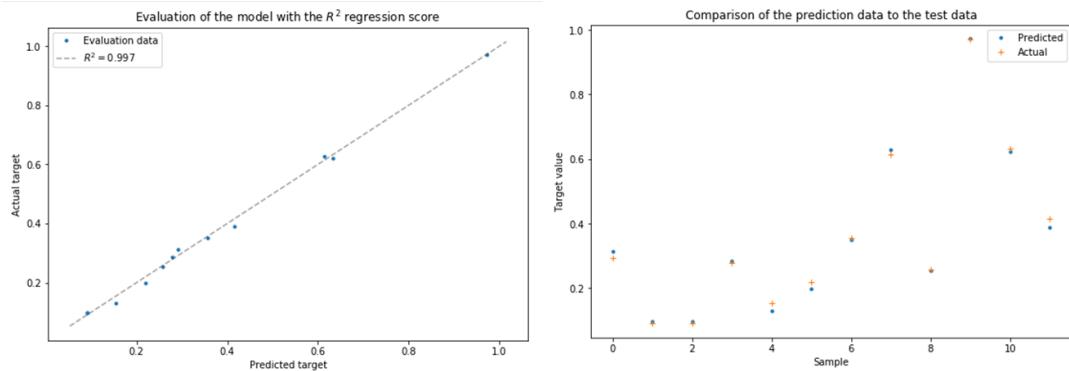


Figure 3.15: The coefficient of determination (to the left) and the target against the prediction (to the right) in the testing phase.

3.2.3 Inverse problem problem

With the same collected data we would like to train the MLP model with mean squared error of the profiles and have the parameters as output. We begin with the predictiton

of one parameter which is the maximum heat intensity Q .

The same architecture is used. The figure 3.16 illustates the results of the prediction model.

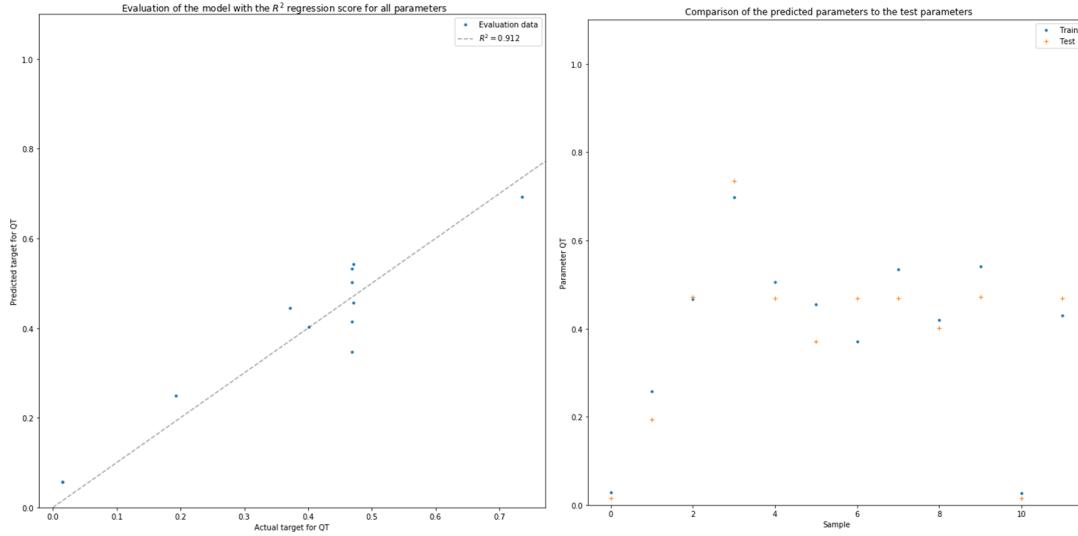


Figure 3.16: The coefficient of determination (to the left) and the target against the prediction (to the right) in the testing phase for parameter Q .

The R^2 is over 90% and there still room for enhancement. The proposed models does not have the goal to substitute FEA simulations but they might be a starting point of the specialist who would like to fit the parameters. It can also be a good initial guess for the gradient descent based optimization algorithms. Now we include two more parameters as shown in the figure 3.17 illustates the results of the prediction model.

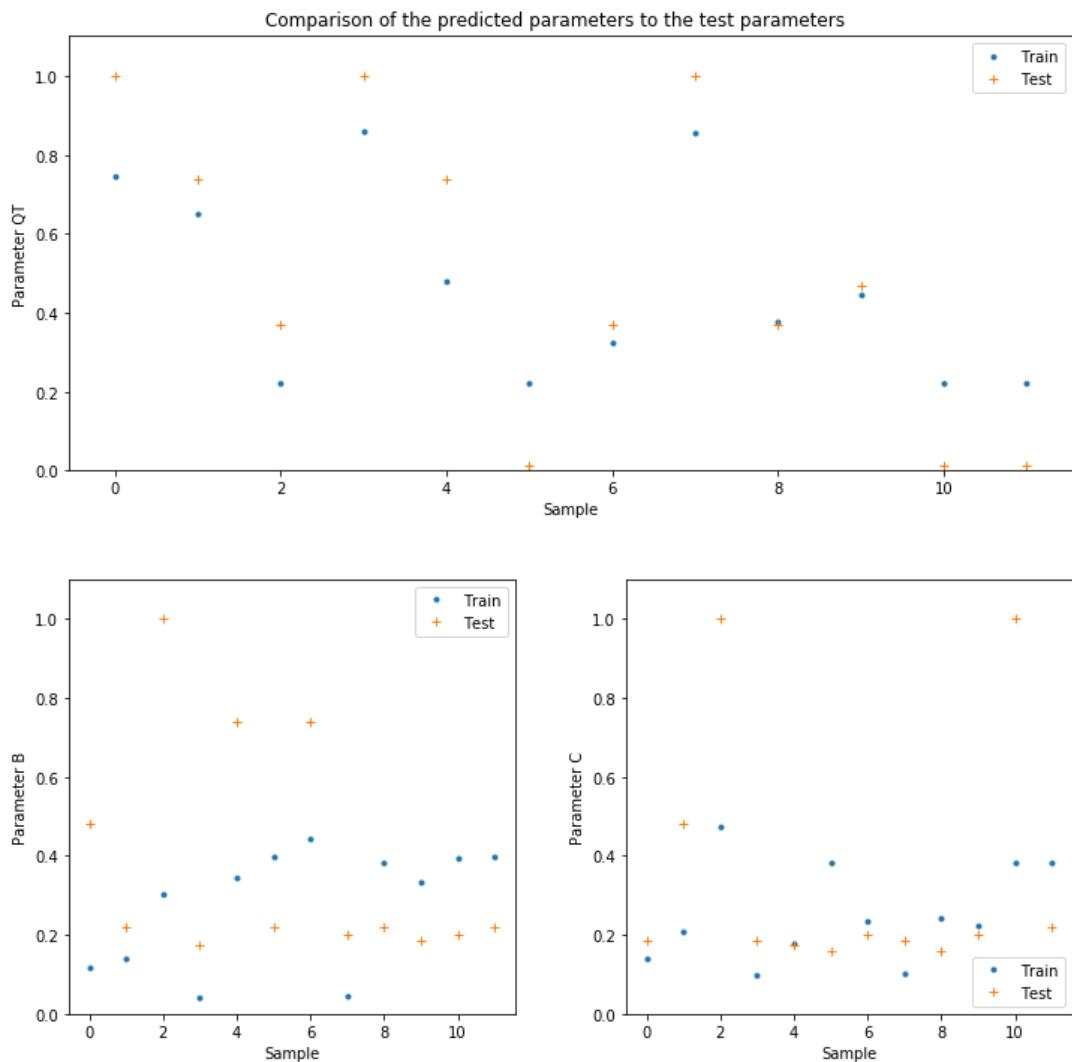


Figure 3.17: The target against the prediction in the testing phase for parameters Q, B and C.

The scores for B and C are very low compared to the case of the maximum intensity Q.

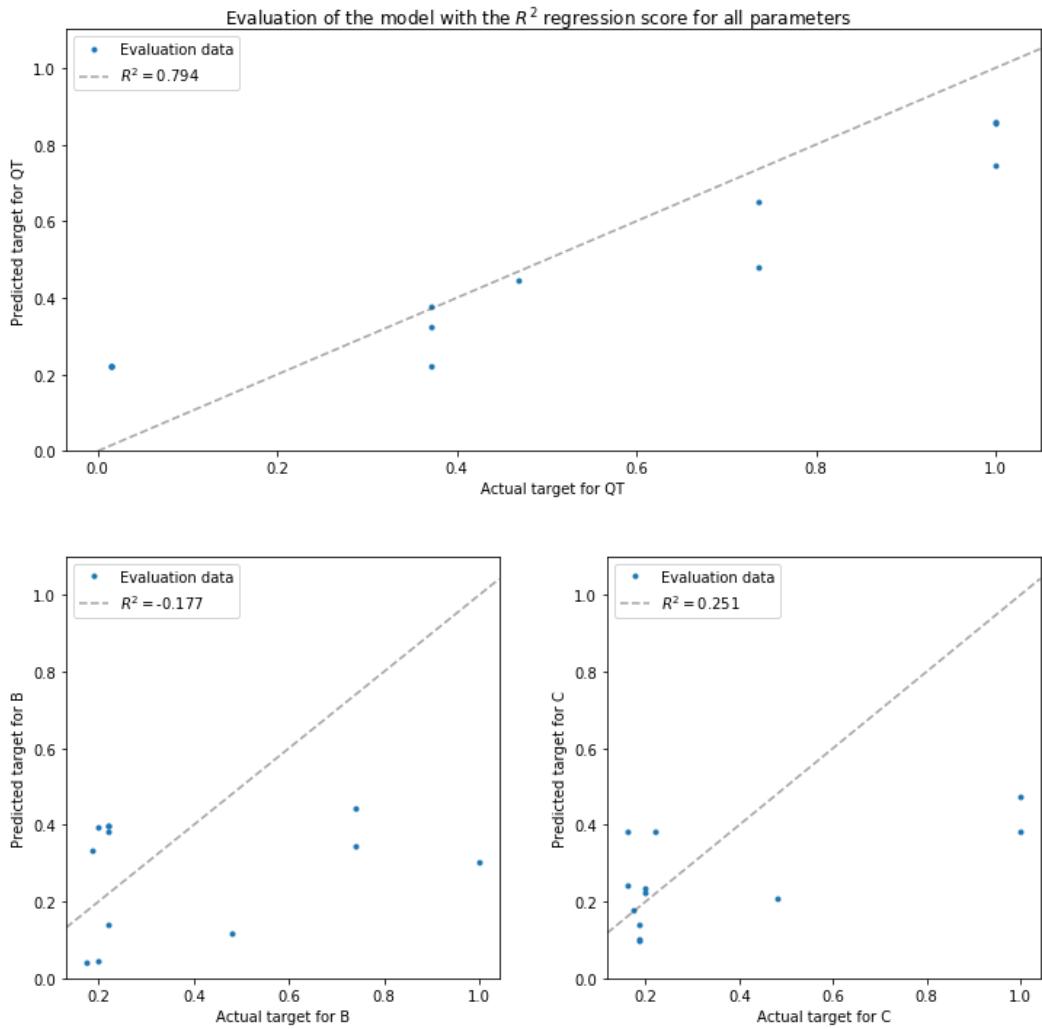


Figure 3.18: The coefficient of determination in the testing phase for parameters Q, B and C.

The purpose of such chapter to demonstrate the capability of the MLP Neural Networks to assist in solving inverse problems. Especially, when there is already a correlation between the metric and the parameter to be identified as shown in figure 3.19.

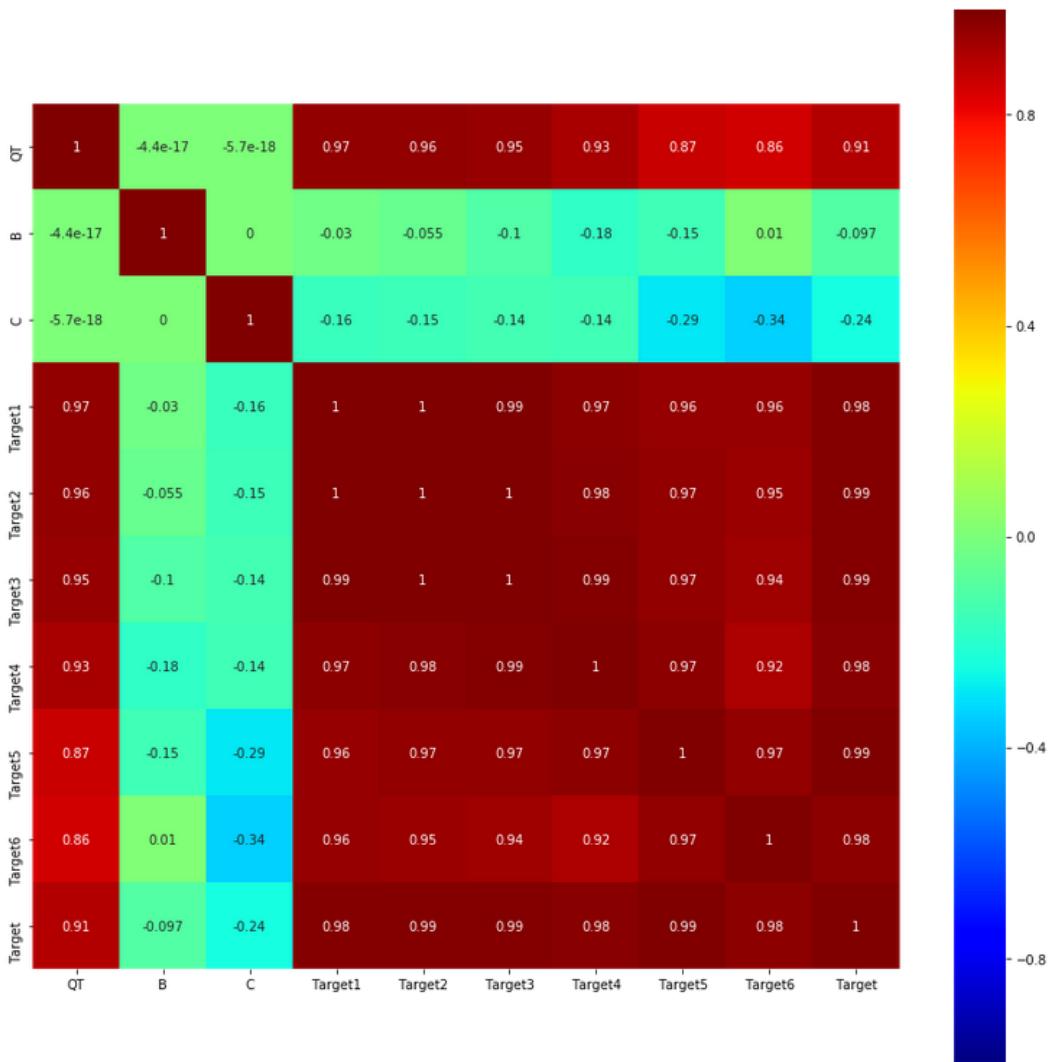


Figure 3.19: The Pearson correlation between all the variable including metrics and parameters Q, B and C.

There is a very low correlation with the parameters B and C with all the measurement metrics. So, those parameters does not impact considerably enough the temperature in the measurement area. They describe rather the distribution of such intensity. In the Goldak model, they represents local geometrical shape of the weld bead and thus capturing them with such models does not look obvious as demonstrated for the case of the maximum heat source intencity Q.

Chapter 4

White box approach

This chapter is based on the Galerkin equation finite element approach explained in section 2.3.

4.1 ABAQUS simulation model

4.1.1 Description

We recall that the obtained nodal discretized Galerkin equation for the nonlinear transient heat source equation is:

$$C^N \left(\frac{T_{t+\Delta t}^N - T_t^N}{2 * \Delta t} \right) = I^N + G^N + Loss^N \quad (4.1)$$

Where:

- Nodal internal energy with $C^N = \int_{V_h} \rho C_p(T_h)$
- Nodal internal load related to conductive energy with $I = - \int_{V_h} \nabla T^N \cdot k \cdot \nabla N^N(x)$
- Nodal heat source load with $G^N = \int_{V_h} N^N(x) G(t, x)$
- Nodal losses with $Loss^N = \int_{\Sigma_h} N^N(x) Loss(T^N)$

The nodal thermal conductivity value (given by the energy balance) at node N as it

is defined by ABAQUS documentation is:

$$K^{NM} = -\frac{\partial [I^N + Loss^N]}{\partial T^M}$$

This set of equations is still continuous with respect to time and Abaqus/Standard uses the backward difference algorithm:

$$\frac{dT}{dt} = \frac{T_{t+\Delta t} - T_t}{2\Delta t} \quad (4.2)$$

Is it possible to extract those matrices in order to build a similar but faster solver than ABAQUS/Standard? And how if yes?

4.1.2 Matrices generation

A linearized finite element model can be summarized in terms of matrices representing the internal energy, stiffness (Jacobian), and loads in the model. ABAQUS made it possible, in order to exchange the model data between other users, vendors, or software packages without exchanging mesh or material data. Matrix generation, or precisely thermal matrix generation in our case, is a linear perturbation step that accounts for all current boundary conditions, loads, and material response in a model. The generated matrices are input to a matrix usage model.

The matrix generation procedure uses *.SIM* file, which is a high-performance database available in ABAQUS. The generated matrices are stored in a file named "*jobname**xn.sim*", where *jobname* is the name of the input file or analysis job and *n* is the number of the Abaqus step that generates the matrices.

In order to extract those matrices, preferably in assembled format, we include the following lines in the ".inp" file [5].

```

*HEADING
...
**
*STEP
Options to define an uncoupled heat transfer analysis.
...
*BOUNDARY
Options to define the boundary conditions for the heat transfer step.
**
*CFLUX and/or *DFLUX and/or *DSFLUX
Data lines to define thermal loading
*FILM and/or *SFILM and/or *RADIATE and/or *SRADIATE
Data lines to define convective film and radiation conditions
**
*ELEMENT OPERATOR OUTPUT, ASSEMBLE, STIFFNESS, DAMPING,
LOAD, LOADTYPE=EXTERNAL, FREQUENCY=1
**
Options to define the output requests for the heat transfer step.
**
*END STEP

```

Figure 4.1: A screenshot of the lines to add (boxed in green) in the ".inp" file to extract thermal matrices.

With such lines we will be extracting: the heat capacity C and the conductivity K matrices, which are temperature dependant. It is worth mentionning that the extracted conductivity matrice include the loss variation as well. In order to obtain the pure conductivity matrice which we call internal load matrix (denoted K_{cond} to not mix with the ABQUS called conductivity matrix K) a regression model is proposed in section ???. Additionally, the external load vector is generated and denoted P. In order to generate the internal load vector, the key word "EXTERNAL" in the second line of the green box in figure 4.1 has to be changed with "NET" in order toextract Net vector. The net vector is defined as the sum of the internal and external vectors.

$$Net = I + P$$

$$\text{Hence, } I = Net - P$$

4.1.3 Numerical validation

Now that we have C_T , K_T , Net and P , we can proceed to a first verification, before further steps, by solving:

$$C_T \frac{T_{t+\Delta t} - T_t}{2\Delta t} = Net \quad (4.3)$$

For each time step. Each time step is denoted frame.

To solve this linear algebraic equation we use the conjugate gradient method. The figure 4.2 illustrate a perfect score and thus the soundness of the extracted matrices, the time descritisation and the implemented conjugate gradient.

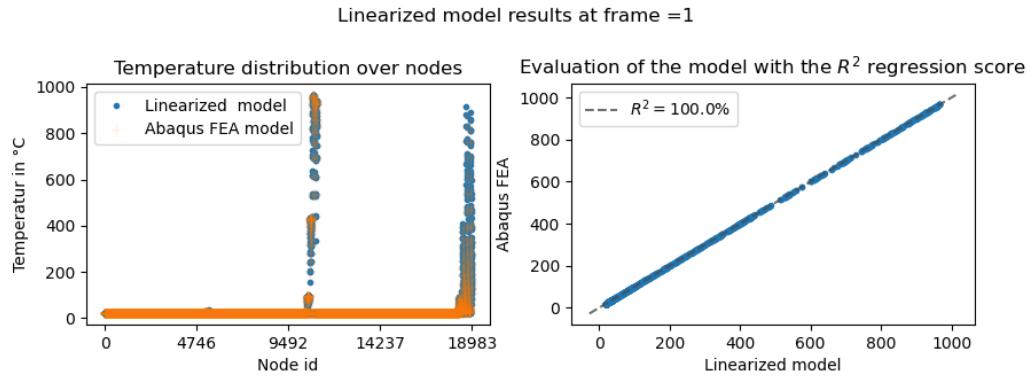


Figure 4.2: The linearized model solution compared to ABAQUS solution for frame 1 using conjugate gradient solver.

The relative error is very low, hence the conjugate gradient solver applied to the linearized model is giving a precise solution as it is illustrated in the figure 4.3.

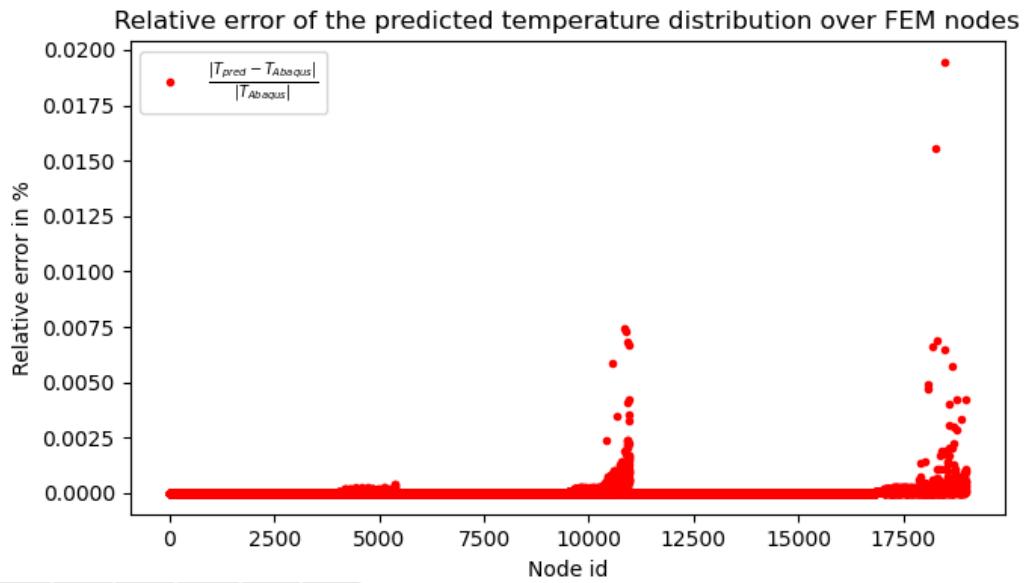


Figure 4.3: The linearized model solution compared to ABAQUS solution for frame 1 using conjugate gradient solver.

Same magnitude of precision is kept until the end of the welding simulation. The figure can be checked in the appendix [D.1](#).

4.1.4 Problematic

Regarding the Newton's method solver, a strong deviation is observed along the frames. The figure [4.4](#) (right figure) represents a lower R^2 score which is still high because of the big nodes which are not in the heat affected zone.

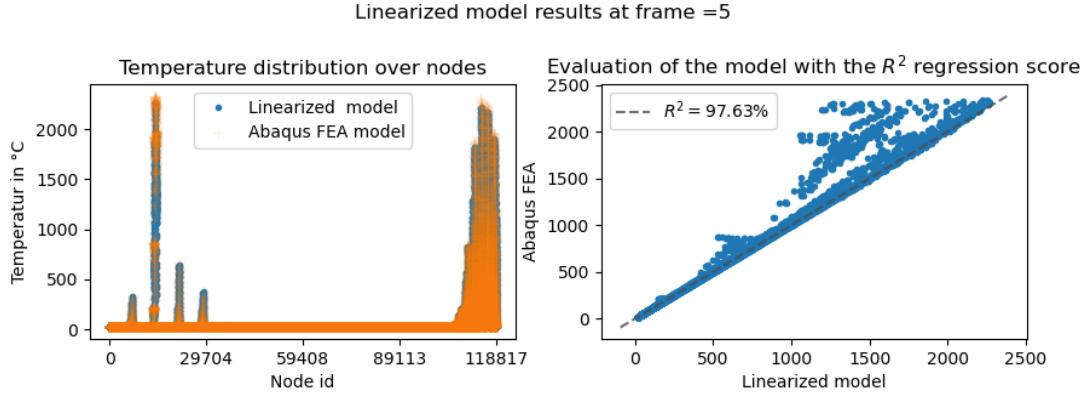


Figure 4.4: The linearized model solution compared to ABAQUS solution for frame 5 using Newton's method solver

The relative error is more reliable and it is as expected very high, thus the Newton's method solver applied to the linearized model is giving low precise solution as it is illustrated in the figure 4.5.

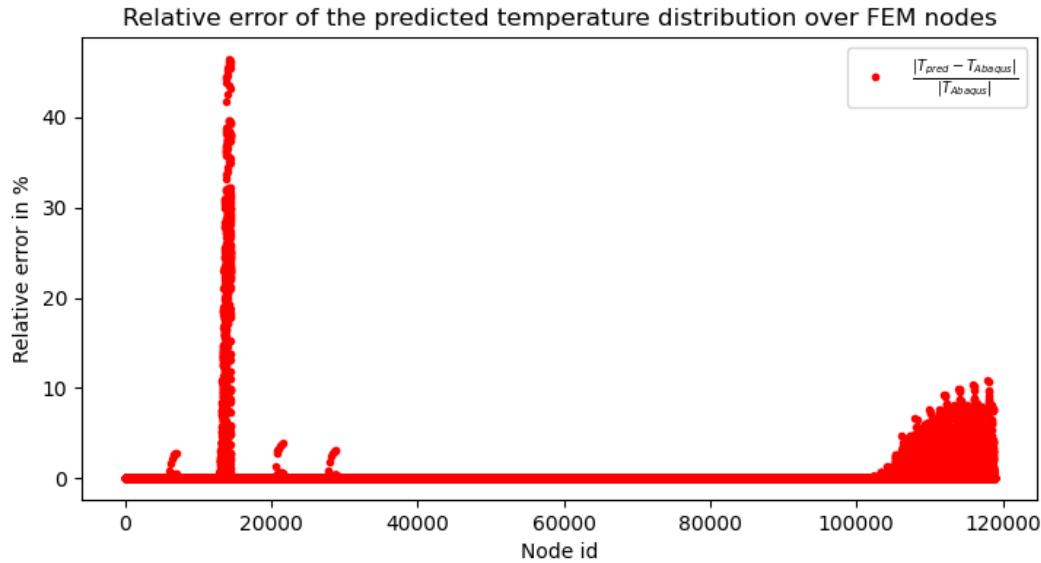


Figure 4.5: The linearized model solution compared to ABAQUS solution for frame 5 using Newton's method solver.

This is expected because the model is linearized and the temperature variation in the iteration of the Newton's method inner loop are not considered. The Newton's can be used but only if we explicit the terms with temperature dependency. This means that we have to substitute the *Net* load with internal I and external load P . And formulate

the vectors and matrices separately from material coefficients heat conductivity $k(T)$ and specific heat capacity $c(T)$.

$$I = Net - P = K_{cond} @ T = \gamma_t K_T @ T, \quad K_{cond} = \gamma_t k(T) K$$

and

$$P = \beta Loss(T) + \alpha G(X)$$

$\frac{dLoss(T)}{dT}$ is a language abuse to express the dual representation of the loss variation with respect to T $\frac{d \int_{\Sigma_h} Loss(T^M) N^M(x) N^N(x)}{dT^M}$. In addition to the heat capacity:

$$C_T = \epsilon c(T) C$$

The coming sections are proposing least square regression models to identify the following coefficients:

- α and β : only geometry dependant (finite volumes shape functions $N(X)$)
- γ_t : Depending on the geometry and the heat loss variation (because of heat affected zone change)
- C which is diagonal matrix depending on the density (fixed) and the geometry

In order to have the temperature explicit nonlinear algebraic equation:

$$c(T) \epsilon C \frac{T_{t+\Delta t} - T_t}{2\Delta t} = \gamma_t k(T) K @ T_{t+\Delta t} + \beta Loss(T) + \alpha G(X) \quad (4.4)$$

Where X is the nodal coordinates in the vectorial representation of the Goldak heat source. The nodal representation of the Galerkin equation allows us to write the algebraic equation 4.4 in a vectorial form thanks to Riesz theorem.

4.2 Parameter fitting

4.2.1 Heat source and losses models

$G(X)$ is the vectorial representation of the nodal formula $G^N = \int_{V_h} N^N(x) G(t, x)$. In this part, the heat source parameters are fixed and $G(X) \in \mathbb{R}^N$ where $G^N =$

$\int_{V_h} N^N(x)G(t, x)$ and $G(t, x, y, z)$ denoted $G(t, x)$ equals piecewisely to the continuous introduced heat source. We recall that:

$$G(t, x, y, z) = \begin{cases} Q \exp\left(-\left(\frac{x}{b}\right)^2 - \left(\frac{y}{c}\right)^2 - \left(\frac{z_t}{a_f}\right)^2\right) & \text{if } z_t > 0 \\ Q \exp\left(-\left(\frac{x}{b}\right)^2 - \left(\frac{y}{c}\right)^2 - \left(\frac{z_t}{a_r}\right)^2\right) & \text{else } z_t \leq 0 \end{cases} \quad (4.5)$$

The Python source code of the heat source vectorial representation is presented in appendix D.2. It is method of the "Linearized Model" class available in the provided in link in appendix A.2.

The losses are modeled as follows:

$$\text{Loss}(T) = -h(T - T_0) - \epsilon \sigma(T^4(t, x) - T_0^4)$$

Now we have to solve the following regression problem:

$$\text{Minimize } J(\alpha, \beta) = \sum_{\text{frames}} \|P - \beta \text{Loss}(T) - \alpha G(t, x)\|_2^2$$

4.2.2 Least square method

The problem of minimizing J with respect to $X = [\alpha, \beta]^T$ is quadratique problem of which the uniqueness of the solution is depending on whether the Hessian is definite positive. formulated as follows:

$$\text{Hessian} = \begin{pmatrix} G(t_1, x) & \text{Loss}(T_1) \\ \vdots & \vdots \\ G(t_n, x) & \text{Loss}(T_n) \end{pmatrix}^T @ \begin{pmatrix} G(t_1, x) & \text{Loss}(T_1) \\ \vdots & \vdots \\ G(t_n, x) & \text{Loss}(T_n) \end{pmatrix}$$

and the first order optimal condition is:

$$\text{Hessian} @ X - b = 0 \quad \text{with} \quad b = \begin{pmatrix} G(t_1, x) & \text{Loss}(T_1) \\ \vdots & \vdots \\ G(t_n, x) & \text{Loss}(T_n) \end{pmatrix}^T @ \begin{pmatrix} P_1 \\ \vdots \\ P_n \end{pmatrix}$$

Which we solve using the conjugate gradient method. The obtained results are as

expected and illustrated in the figure 4.6.

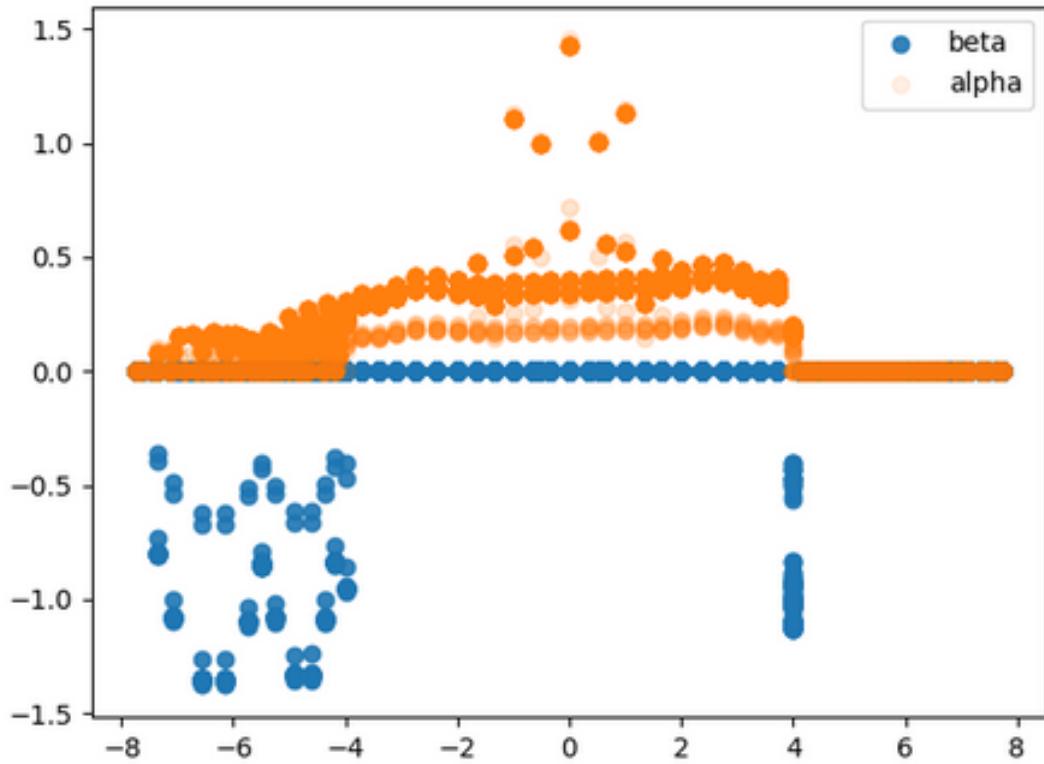


Figure 4.6: The respective heat source and losses weights over Y axis of the component.

β is negative as weights of the losses and α is positive. In this part the model is reduced to the affected heat zone and heat exchange is not considered between -4 and 4. additionnaly , the magnitude of weights correlate reasonably with the finite element diemensions e.g. bigger volumes in the middle.

Moreover, the predicted external load and model match perfectly as shown in the figure 4.7.

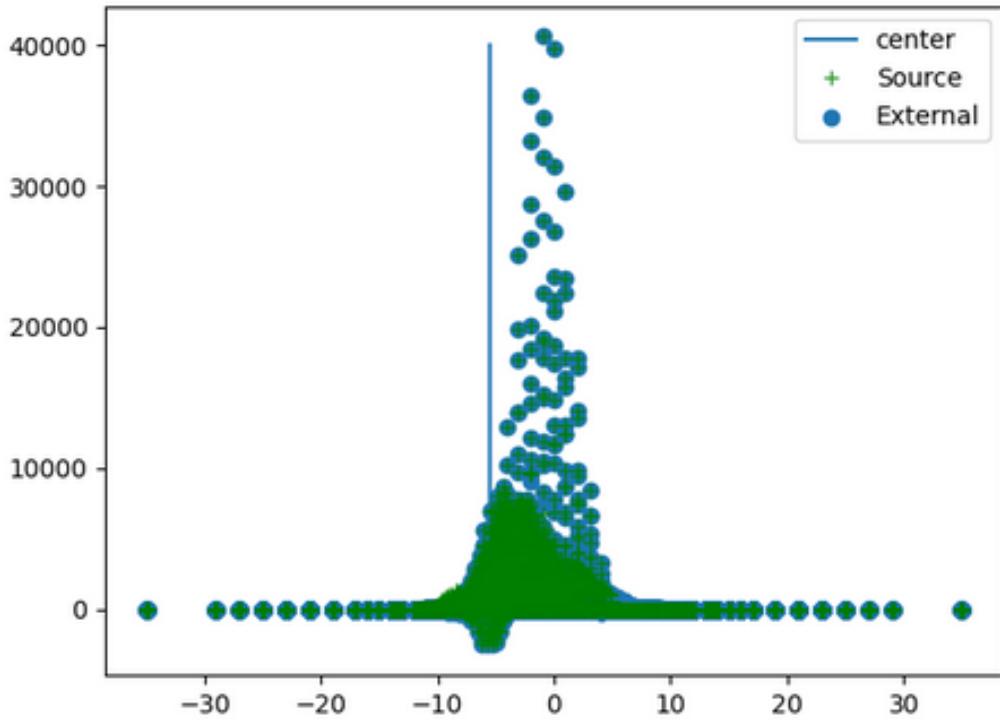


Figure 4.7: The External heat source predicted model (in green) and the generated External load (in blue).

4.2.3 Internal loads parametrisation

Regarding the internal loads, the proposed model in order to explicitly formulate the temperature dependant terms based on the given Jacobian or (conductivity matrix) K and the internal load vector I is:

$$I = \gamma_t k(T) K @ T$$

Where γ_t is time depedant parameter (including the loss variation for each time step) and $k(T)$ is given temperature dependant conductivity. It is constant value defined for each temperature interval.

The parameter γ_t is obtained for each frame by applying the least square method as explainind in the previous case. The figure 4.8 illustrates the results for frame 1 on the training model.

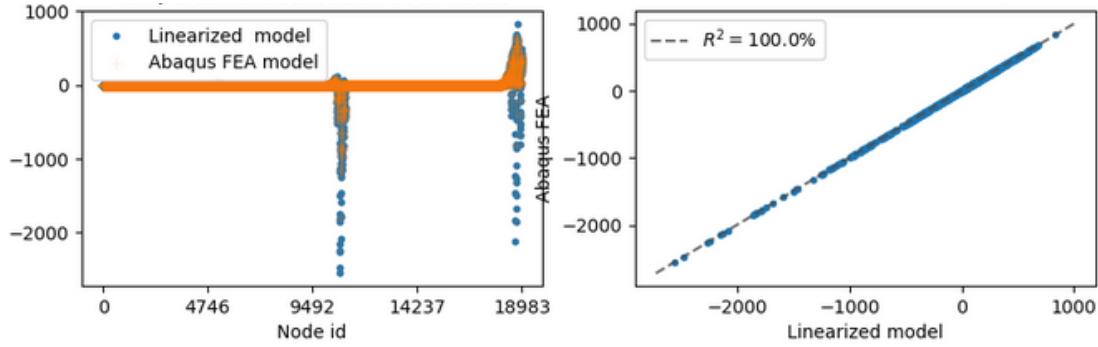


Figure 4.8: The predicted and linearized internal loads models and the corresponding R^2 evaluation on the study case model.

Actually two models are used for this study. The training model with very low heat source intensity and thus low internal loads (see figure D.4) and the other model for validation (see figure D.5). The validation is displayed in figure 4.9 for the first frame using the weights obtained by the training model.

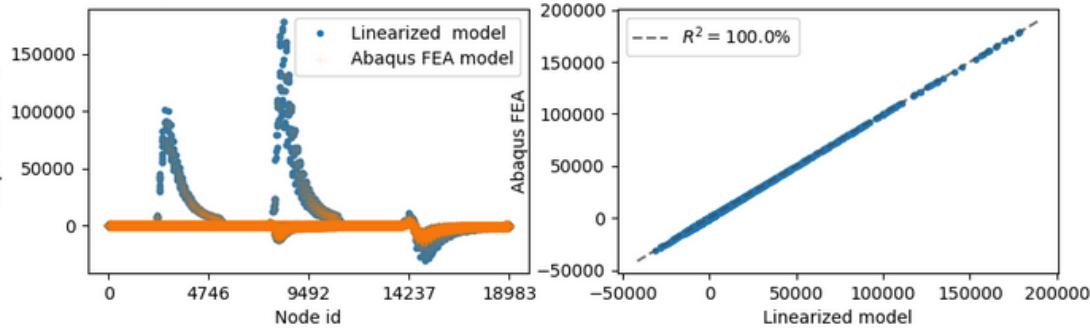


Figure 4.9: The predicted and linearized internal loads models and the corresponding R^2 evaluation on a validation case.

The results are as expected because the temperature variation of the losses is constant or even can be neglected because of the magnitude of the radiation term T^3 with respect to the Stephan-Boltzmann constant *i.e.* $5.67 \cdot 10^{-11}$. at thmeasurements position we are not supposed to reach higher temperature than a melting temperature at the welding zone e.g. $\sim 10^3$. With such magnitude, the radiation term variation (T^3) times the Stephan-Boltzmann coefficient (times the emissivity *i.e* 0.8) can be neglected. which leads to almost a perfect score for both training and validation models.

4.2.4 Inertia parametrisation

The inertia or the internal energy term is modeled as follows:

$$C_T = \epsilon c(T) C$$

The same procedure as before, is applied to this model both training model (figure 4.10) and validation model at both first (figure 4.11) and last frame (figure 4.12) are displayed to illustrate the obtained results.

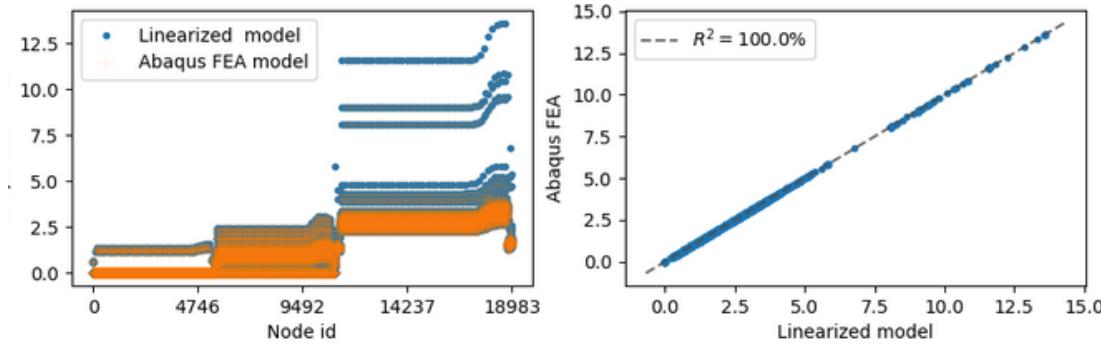


Figure 4.10: The predicted and linearized internal energies models and the corresponding R^2 evaluation on the study case model.

Here, is the validation models

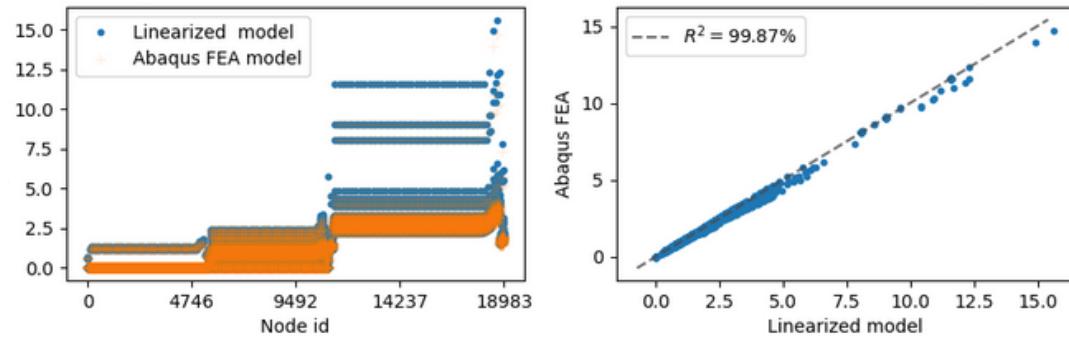


Figure 4.11: The predicted and linearized internal energies models and the corresponding R^2 evaluation on the validation case at frame 1 (first).

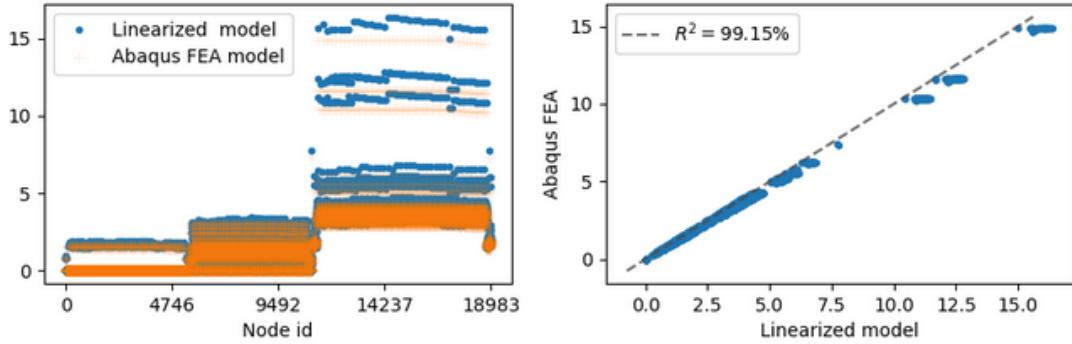


Figure 4.12: The predicted and linearized internal energies models and the corresponding R^2 evaluation on the validation case at frame 74 (last).

With a score of more than 90%, the model of the internal energy can be definitely integrated in the elaborated white box model for the welding process at a specific geometry (1.2.4).

Based on the shown study, the following model:

$$c(T)\epsilon_t C \frac{\partial T}{\partial t} = \gamma_t k(T) K @ T_{t+\Delta t} + \beta Loss(T) + \alpha G(x, t) \quad (4.6)$$

can be used for solving the inverse problem instead of using the time consuming ABAQUS/Standard to solve the forward problem. The adopted solver for such problem is the Newton method, which is explained in section ??.

4.3 Inverse problem

4.3.1 Approximated Gradient descent

This approach is identical to the first proposed tool, only ABAQUS/Standard solver is substituted with the proposed reduced model. The advantage of such model compared to deep learning models is interpretability and the user can choose the measurements positions, the time interval, the boundary conditions etc in more flexible way. Moreover, material parametric study is possible thanks to the explicit formula 4.7. The figure 4.13 represents the optimization loop based on the proposed model.

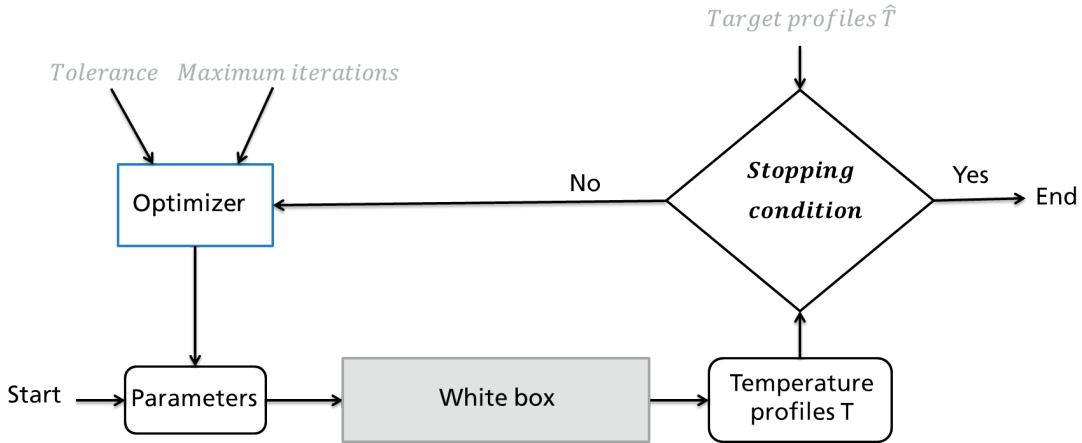


Figure 4.13: A schema on the proposed white box based optimisation tool to solve the inverse problem.

In this method, obviously the gradient is only approximated numerically, which impacts the accuracy of the solution.

4.3.2 Pontryagin Maximum Principle

The Pontryagin Maximum Principle (PMP) for nonlinear set of ODEs, allow an exact calculation of the gradient which leads to faster and more accurate results. The uniqueness is guaranteed by introducing the adjoint state but only for local minimima. Again, we recall that a unique global solution of the optimal problem is related to the strict convexity character of the objective function. In this we see the problem as an inverse bilevel problem of which we solve the upper level first and then conclude on the lower problem.

We consider the control $G \in \mathbb{R}^k$, where C is assumed to be invertible, $c(T)! = 0$ and $\epsilon! = 0$ and

$$T^* = \frac{dT}{dt} = \frac{1}{c(T)\epsilon_t} C^{-1} [\gamma_t k(T) K @ T(t) + \beta Loss(T) + \alpha G(t)] = f(t, T(t), G(t)) \quad (4.7)$$

Let $f : \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ and $f^0 : \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$ be C^1 of which the control is measurable and bounded, defined on an interval $[0, t^e] \subset \mathbb{R}^+$ and having its values in $U_{ad} \in \mathbb{R}^m$. We consider the objective function of the optimal control problem

to be:

$$\begin{aligned}
& \text{Minimize } C(G_t) = \int_0^{t^e} \|T(t) - \hat{T}\|_2^2 = \int_0^{t^e} f^0(s, T(s), G(s))ds \\
& G \in U_{ad} \subset \mathbb{R}^\times \\
& T \text{ solve 4.7} \\
& \hat{T} = \text{Measurements data}
\end{aligned}$$

We define the hamiltonia H of the optimal control system 4.3.2 on $\mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^+ \times \mathbb{R}^m \mapsto \mathbb{R}$ by:

$$H(t, T, p, p^0) = \langle p, f(t, T, G) \rangle + p^0 f^0(t, T, G) \quad (4.8)$$

The theorem of PMP, in its generalized form for nonlinear case¹, states that if there is an optimal control $G \in U_{ad}$ associated to the trajectory $T(\cdot)$, then:

$$\exists p(\cdot) : [0, t^e] \mapsto \mathbb{R}^n$$

absolutely continuous called the adjoint state. and a real number $p^0 \leq 0$, so that $(p(\cdot), p^0)$ is not trivial, and $\forall t \in [0, t^e]$ almost every where:

$$\begin{aligned}
T^*(t) &= \frac{\partial H(t, T, p, G)}{\partial p} \\
p^*(t) &= -\frac{\partial H(t, T, p, G)}{\partial T}
\end{aligned}$$

And we have the maximisation condition:

$$H(t, T, p, G) = \max_{v \in U_{ad}} H(t, T, p, v)$$

¹Explained for instance in the lecture of optimal control of Prof. Yannick Privat [37].

Chapter 5

Summary and achievements

5.1 Summary

First of all, in chapter 1, a brief introduction to welding processes is given. The context as well as the problematic are defined. In order to understand the inverse problem, the forward problem which is solving the nonlinear transient heat equation is introduced and the well-posedness of Galerkin equation was investigated in chapter 2. Also, the ABAQUS/Standard solver for such problems is examined in section 2.3. The reconstruction of such solver is detailed in the "white box" chapter 4. Besides that, several regression models based on either physical or mathematical correlations were elaborated to reconstruct the discretized Partial Differential Equation (PDE). The used method to determine the parameters of the obtained set of Ordinary Differential Equations (ODEs) is mainly the (Preconditionned) conjugate gradient. The number of ODEs is reduced to the heat affected zone and the forward problem is solved using Newton's method. Regarding the inverse problem, which is reconstructing the heat source parameters given temperature profiles at measurement position, there are four proposed solutions. The solutions are categorized as either "black box" approach, where the forward problem is solved either by means of deep learning models or using ABAQUS/Standard solver, or the "white box" approach where a quicker solver but equivalent to ABAQUS/Standard is proposed. In the "black box" approach, an optimization problem based tool is developed. The theoretical examination of such problem is inspected in section 3.1 followed by numerical experiments where the input data is given by a chosen target constellation of parameters. Additionally, the tool is documented as HTML files and hosted in Github (see 1.4.1) and tested by colleagues. One additional "black box" feedforward artificial

neural network model was proposed namely Multilayer Perceptron (MLP) in section 3.2. In chapter 4, we present the reduced transient nonlinear model, with the different established subregression models and we validate each assumption with ABAQUS/Standard results. The solution to the inverse problem is then proposed with two methods. The gradient descent based method is similar to the "black box" developed optimization tool with the main difference to consider similar but faster solver instead of ABAQUS/- Standard. The second method is based on the Pontryagin maximum principle (PMP) and the adjoint state applied to the proposed nonlinear ODEs. Both methods lead to local minima and the uniqueness is only guaranteed by introducing the adjoint state.

5.2 Achievements

With the aid of the approaches and ideas presented in this thesis, it is possible to identify an applicable set of parameters and reconstruct the heat source according to measurements data. The theoretical part of this thesis consists of proving the existence of a solution to such problem. Also, justifying the uniqueness in case of strong convexity with validated assumptions. In fact, the temperature map $T : G \mapsto T_G$ with respect to the heat source density G should be:

+ **Lipschitz continuous**

+ **Affine linear**

In case of non convexity of the cost function, due to strong nonlinearity local minima can still be found by the gradient descent algorithm. One of the key factors to ensure convexity is the position of the measurements and the observation time interval. In the cooling phase where temperatures are low and have small variation, at further (but still in the heat affected zone) measurement positions, the assumptions were demonstrated numerically to be valid.

The concrete achievements of the thesis in addition to the theoretical study are listed as follows:

- Gradient-based optimization tool: Tested, validated and documented (Sphinx webpage generator).
- Multilayer Perceptron (MLP) tool: A decision tool trained with 512 images (1D) for up to three parameters to identify.

- Convolutional neural network tool: A decision tool trained with 648 images (2D) for geometrical parameters i.e. b, c, a_f, a_r .
- Model order reduction tool: A reduced model to solve both forward and inverse problems.
- Graphical User Interface: Providing the user with a friendly platform to choose the tool and launch the parameter identification (ongoing).

5.3 Comparison

To give both qualitative and quantitative idea about the four proposed tools in their current state:

- Gradient-based optimization tool: ABACAL
- Multilayer Perceptron tool: MLP
- Model order reduction tool: MOR

The following criteria were considered:

- Applicability: The ability to be applied to realistic data for instance
- Portability: The ability to be executed as embedded application
- Rapidity: How fast it identifies the parameters
- Accuracy: How close the target profiles are to the obtained profile with the identified parameters

The table 5.1 summarizes the comparison using the determination coefficient R^2 statistic for both applicability and accuracy, the executions profiling for rapidity and categorical variable for portability. The latter criterion is the most trivial because of the fact that ABACAL tool is dependent to ABAQUS FEA which is a commercial software requiring high computing capabilities. The minimum processor required is equivalent to Intel Pentium 4 or AMD processors supporting SSE2 instruction set. While the other proposed tools can also be embedded in microcontrollers which can interact with sensors and actuators of various kinds.

Table 5.1: Illustrative comparison of the four proposed tools for one single parameter identification i.e Q.

	ABACAL	MLP	MOR
Applicability (R^2 score)	-	-	-
Accuracy (R^2 score)	> 99%	> 90%	-
Rapidity (in minutes)	> 20 for one iteration	< 1	< 1 for one iteration
Portability (Yes or no)	0	1	1

For notation we use $>$ to mean "approximatively bigger than" and $<$ to mean "approximatively smaller than". The presented comparison is only illustrative and can be further elaborated with enough validation data. R^2 scores can be checked in the following sections: in 3.16 for MLP and in 3.7 for ABACAL tool. It is based on the forward problem computing time for the ABACAL and MOR tools. For the NN models, it is commonly known that prediction is made in seconds, however the training phase takes more time. The expected result would be that the ABACAL tool which is based on ABAQUS/Standard software is the most accurate but when it comes to rapidity ANN is the most suitable tool and portable tool at the expense of accuracy. The MOR based tool is expected to be a good compromise between all considered criteria. It might also offer the possibility to consider material property parametric study and can be enhanced to be more competitive in terms of accuracy (improving the regression models) or in terms of rapidity (including other methods of order reduction).

Chapter 6

Outlook

Before further developing the proposed tool, it is recommended to make a static documentation for the other tools as it is done for the ABACAL tool (ABAQUS/Standard based optimization tool). The figure 6.1 illustrates such type of documentation.

module.Gradient_descent.Gradient_Loss($L_1, Q_1, Q_2, \text{target}$)
Returns the gradient of the cost function estimated two initial parameter set. The partial derivatives are approximated by backward difference.
Given the direction and the orthogonal basis:
 $h^i = (0, \dots, h_i, \dots, 0), e_{i \in (1, \dots, n)}$
With:
 $h_i = \langle Q_2 - Q_1, e_i \rangle_{R^n}$
We approximate the partial derivatives of $J(x_0, \dots, x_i, \dots, x_n)$ by backward difference as follows:
$$\partial_{x_i} J = \frac{J(Q_1 + h^i) - J(Q_1)}{h_i}$$

For more explanations one can check the following wikipedia pages: [here](#) or [here](#).

Warning: When Q_1 is equal to Q_2 a default value of the gradient is set in the method. It can be changed in following line:

```
>>> if Q1[i] == Q2[i]:  
>>> grad[i] = 0.1
```

Example:

```
>>> Q1 = np.array([1,2,3])  
>>> Q2 = np.array([1,1,2,1,3,1])  
>>> Targets = Gradient_descent.Load_profile(targets_file,nb_col_profiles)  
>>> L1.profile1 = Gradient_descent.Load_profile(Q1,Targets)  
>>> grad = Gradient_Loss(L1,Q1,Q2,target)
```

Figure 6.1: A screenshot of the generated documentation for ABACAL tool.

The possibility to combine the proposed tools is a conceivable paradigm, especially for materials with properties having strong temperature variation. In such case, at least the assumption of linear-affinity with respect to the heat source intensity is not valid. An idea to be examined, is to provide several modes of such nonlinearity and according to the temperature level we can identify the mode of linearity to use. In this thesis, the optimization loop software is based on the Euler backward scheme approximation of the

derivative and thus we could avoid the issue of such problematic. On the other hand, the other proposed neural networks can provide a first guess for the optimization tool.

In order to have better prediction of the proposed neural network models, more varied training data is necessary. The procedure is already established and scripts for data acquisition are parallelized and pre-processing is unchangeable, however the architecture of the proposed neural network could be modified and more hidden layer might be possible with more data. Another idea to be inspected is using the white box outcome in order to build more adapted architecture to such welding type e.g. Tee joint welding or even the geometries of the components to be welded.

The validation of the proposed tools was mainly done numerically. However, to have a reliable portable tool which can be implemented in the test rig for instance, realistic data needs to be acquired and fed to the tool. Also, material properties needs to be taken into account as an additional parameter identification.

From the user point of view, the proposed tools should be easy to manipulate. The figure 6.2 illustrate the proposed GUI for the ABAQUS/Standard based optimization tool.

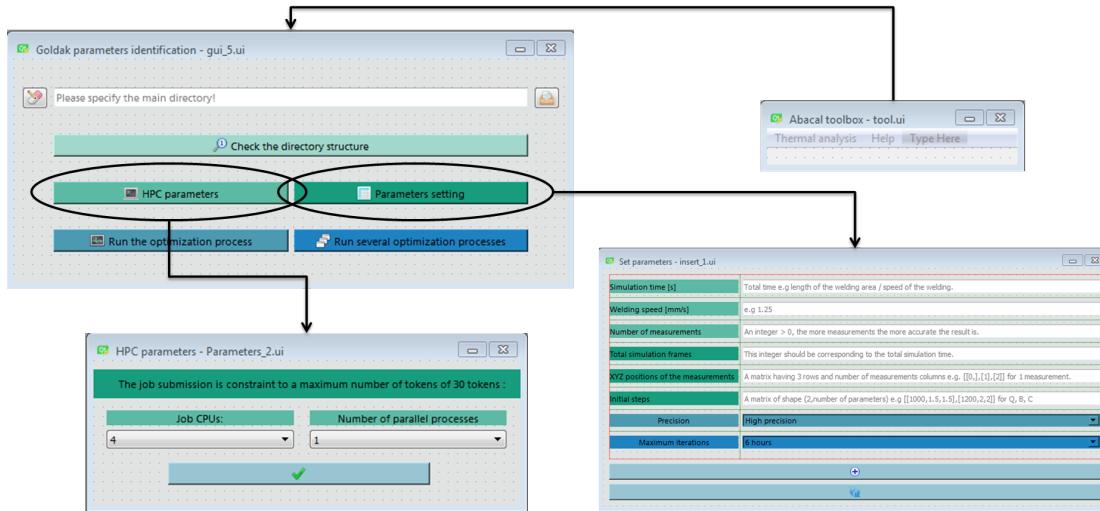


Figure 6.2: A screenshot of the designed GUI dialog windows.

The GUI may be further developed and more features can be added in order to make it usable internally in other research groups facing similar problematics. Once a robust, rapid and memory effective software is validated (with realistic data), it would be even more efficient if it is directly used in the lab where the experiments are driven and include it in the real time data acquisition loop.

Convolutional Neural Networks based approach is ongoing and due to time constraint no results are presented. Actually, they were designed to map image data to an output variable. They have proven so effective that they are the go-to method for any type of prediction problem involving image data as an input. In this section the data are representing two channels of black and white images. The figure 6.3 exemplify the type of data used for training.

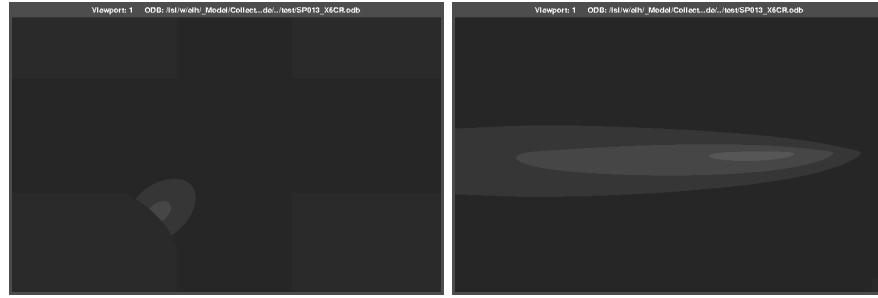


Figure 6.3: Type of the training data to build the CNN model.

The idea behind such type of data is the fact that the identified parameters such as B and C are describing the shape of the weld bead which might be "easily" recognized by the CNN models. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. CNN is also computationally efficient. It uses special convolution and pooling operations and performs parameter sharing.

The data collection was made based on a parallelized and automatized workflow following the steps below for each process:

1. Change the parameters set
2. Run FEA jobs and wait for job to end
3. Postprocessing: Extract relevant data e.g temperature profiles
4. Postprocessing the images: grey scale, zoom, remove and set background and the most important is unify the scale for all images and name them according to time of extraction.

Appendix A

Repositories

A.1 Gantt chart

This https://github.com/youness-elh/Master_Gantt/tree/main is a link of the Gantt chart detailing the different thesis packages and the corresponding milestones is presented. The Excel template has been downloaded from [42].

A.2 Source code

This the Github repository https://github.com/youness-elh/Master_thesis_packages for the following thesis coding packages:

- Gradient-based optimization tool: Tested, validated and documented (Sphinx webpage generator).
- Multilayer Perceptron (MLP) tool: A decision tool trained with 512 images (1D) for up to three parameters to identify.
- Convolutional neural network tool: A decision tool trained with 648 images (2D) for geometrical parameters i.e. b, c, a_f, a_r .
- Model order reduction tool: A reduced model to solve both forward and inverse problems.
- Graphical User Interface: Providing the user with a friendly plateform to choose the tool and launch the parameter identification (ongoing).

A.3 Documentation

The documentation of each package is included in the notebooks except for the ABAQUS-/Standard based optimization tool which was elaborated in more detailed way. Here is the link of the generated documentation the full documentation is hosted on Github and can be consulted via the following Github: https://youness-elh.github.io/Abacal_doc/.

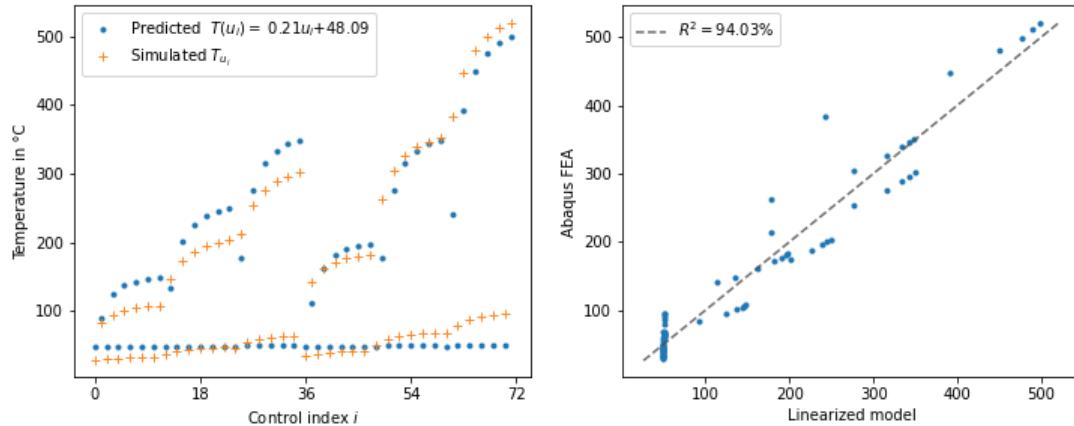
Appendix B

Regression models

B.1 Convexity hypothesis

Here more illustrative figure on the assumption regarding the state variable linearity with respect to the control.

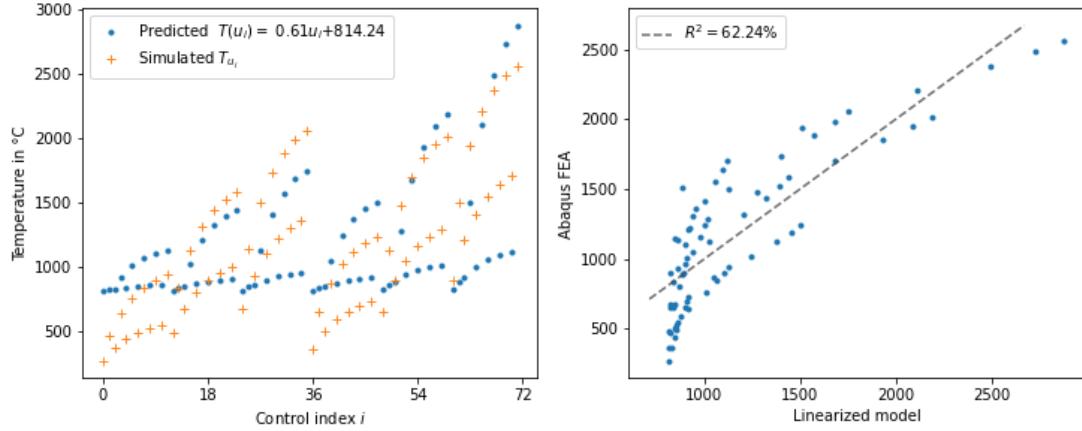
Figure B.1: Illustrative case where the tempererute is affine linear with respect to u in the beggining of the heating phase.



The position of the measurements is $x, y, z = 10, -4, 10$ and time frame for the figure above is 40 which corresponds to the beggining of the heating phase. At that phase, the temperature are also relatively low and the variation of T_u is not steep with respect to u . In the opposite to the figure below where the variation is steep and the assumption is less valid. The choice of the position is also important, because if it is chosen very far

then the control is null as it is not fully distributed to the domain. In the other hand when choosing the position of the sensors close to the welding zone, the radiation losses are not neglected and we lose the linear property. The other practical reason to not put the sensors close is to avoid being damaged from the high temperature at melting point for instance.

Figure B.2: Illustrative case where the temperature is affine linear with respect to u in the end of the heating phase.



Appendix C

Solution uniqueness

C.1 Convexity hypothesis

Numerical experiment to identify 3 parameters:

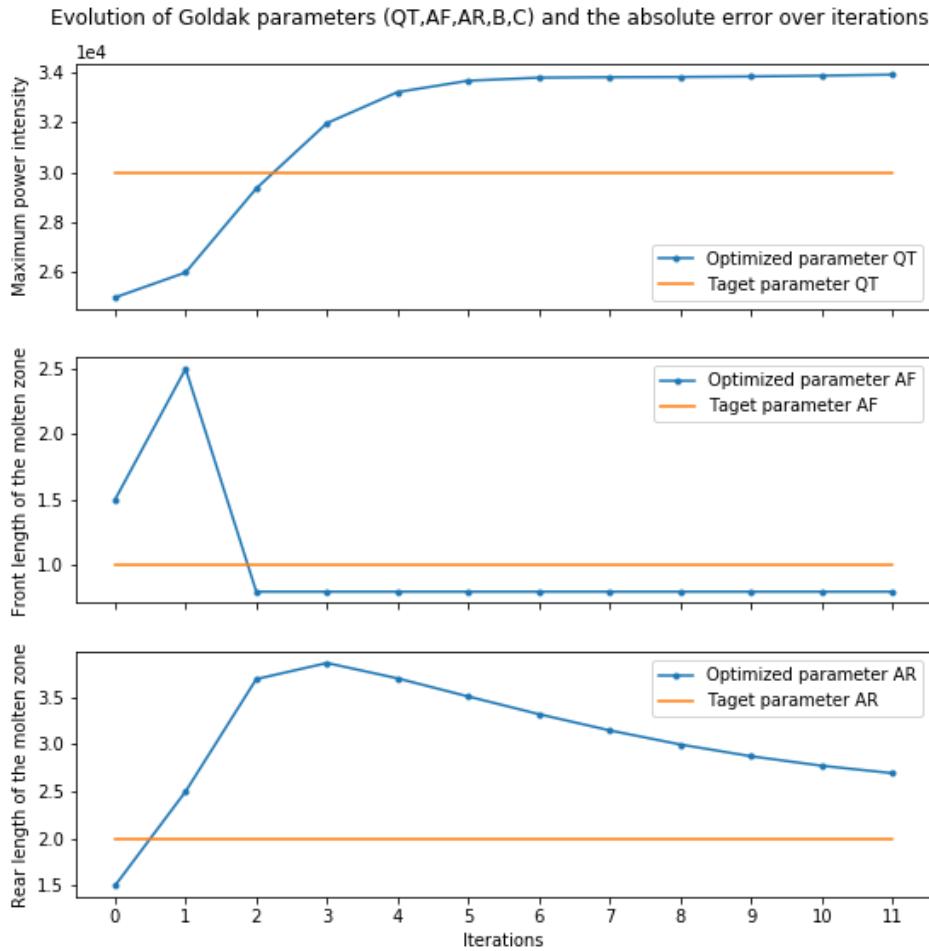


Figure C.1: The result of the gradient descent tool to identify three parameters.

The stopping criteria is the square absolute error $\sum_{t=i}^{76} \|T - \hat{T}\|_2^2 < 1e-1$. The regulation coefficients $1e - 18$, the numerical data are smooth and no need for regulation.

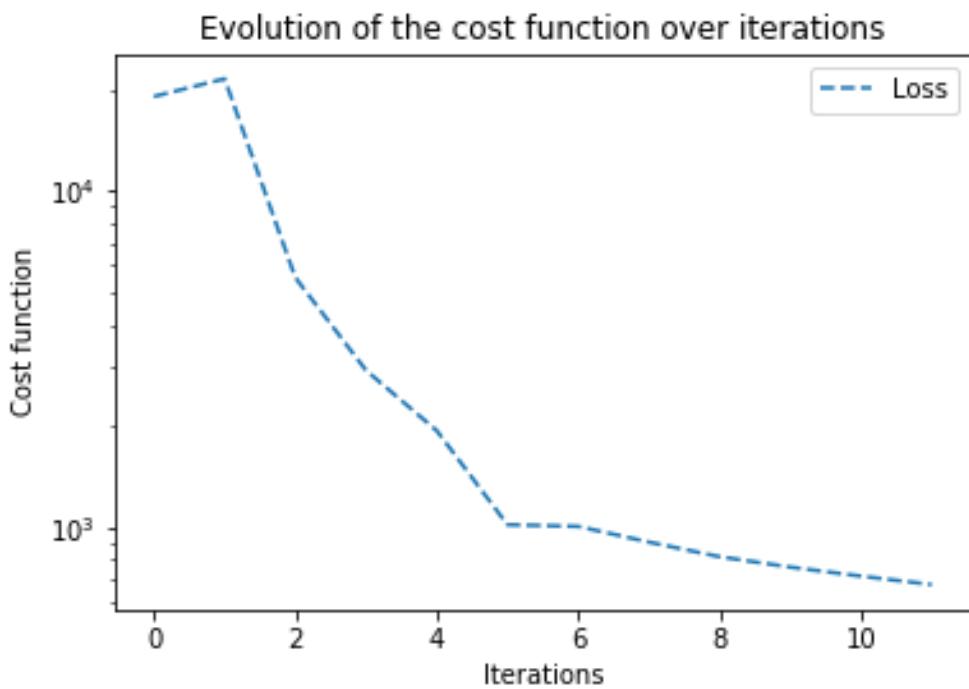


Figure C.2: The cost function for the identification of three parameters.

Numerical experiment to identify 4 parameters:

Evolution of Goldak parameters (QT,AF,AR,B,C) and the absolute error over iterations

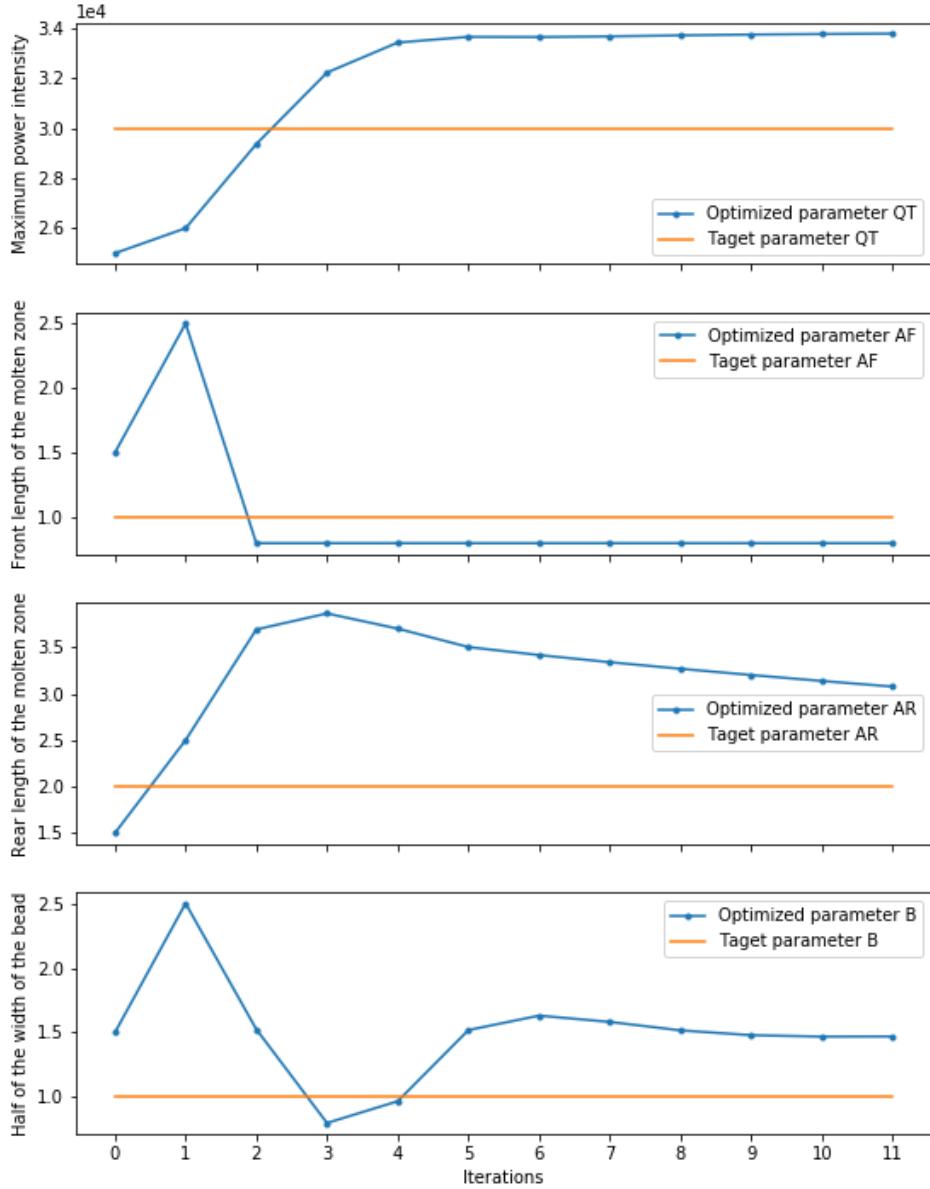


Figure C.3: The result of the gradient descent tool to identify four parameters.

The step computing is not displayed. The stopping criteria is the square absolute error $\sum_{t=i}^{76} \|T - \hat{T}\|_2^2 < 1e-1$. The regulation coefficients $1e-18$, the numerical data are smooth and no need for regulation.

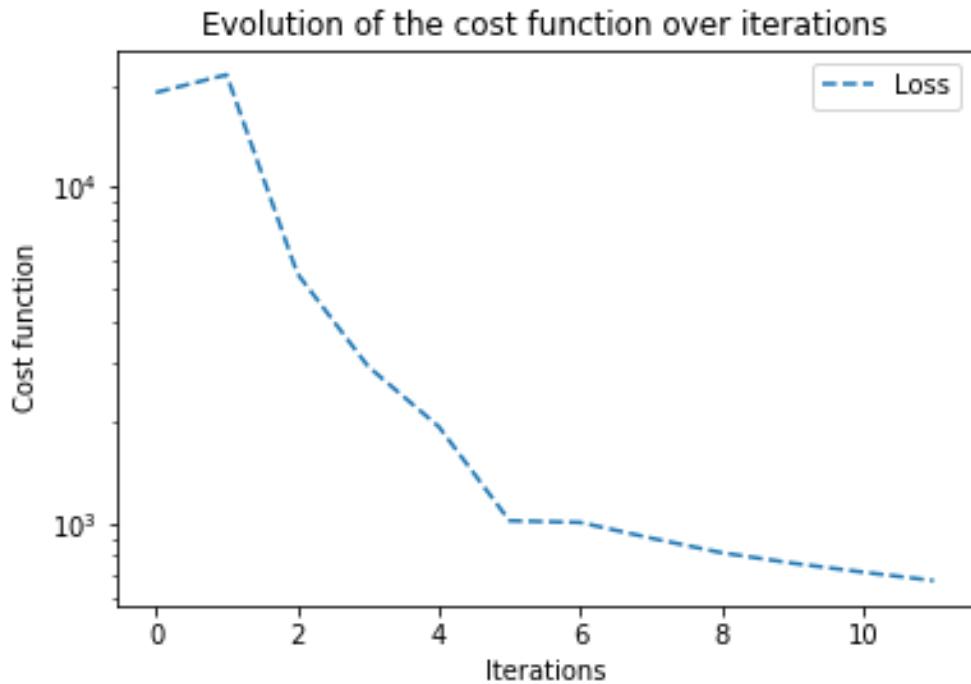


Figure C.4: The loss function for the identification of four parameters.

Appendix D

White box

D.1 Conjugate gradient solution

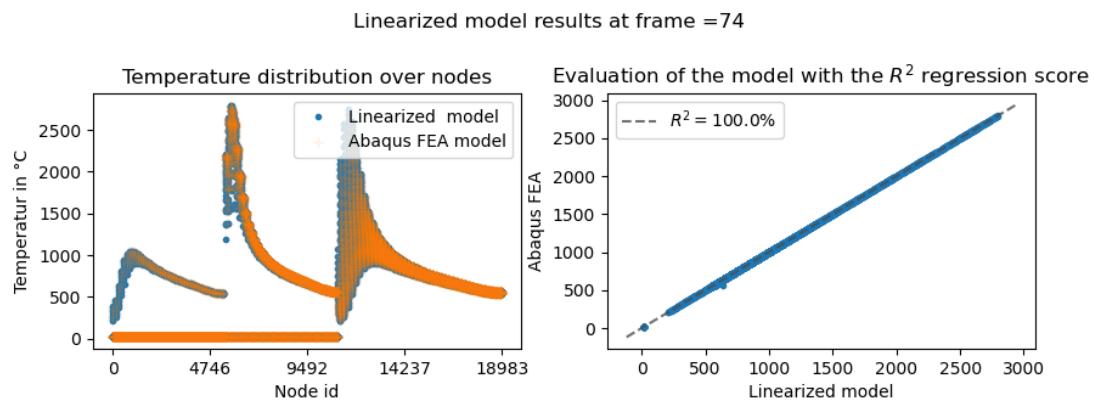


Figure D.1: The linearized model solution compared to ABAQUS solution for frame 74 using Conjugate gradient solver.

The relative error is very high and the conjugate gradient solver applied to the linearized model is giving a precise solution as it is illustrated in the figure 4.3.

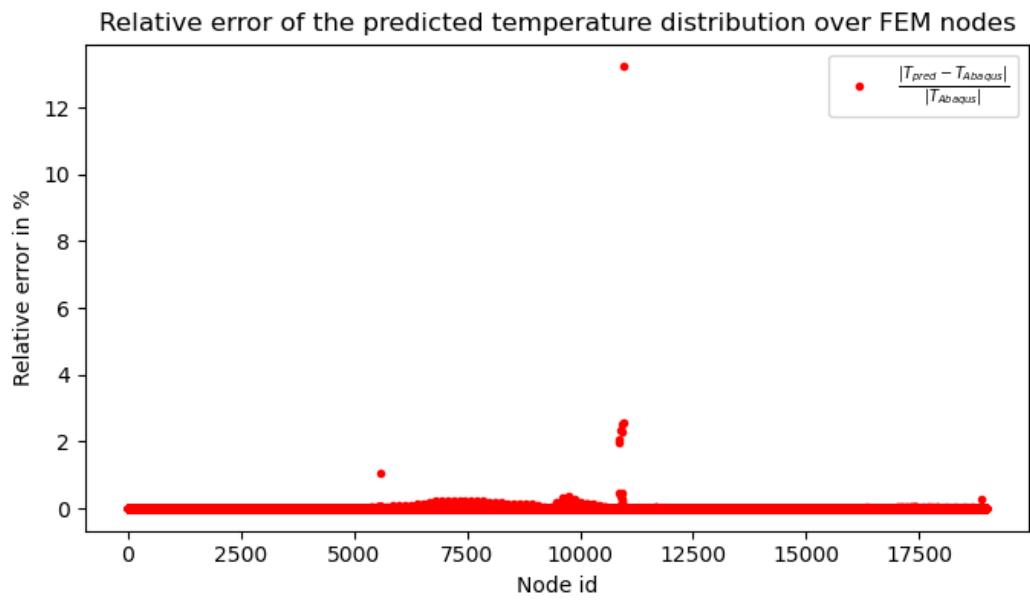


Figure D.2: The linearized model solution compared to ABAQUS solution for frame 74 using Conjugate gradient solver.

D.2 Heat source code

```

def Heat_source(self,frame,center=[5.496,-5.501,-50.01],Angle=45.,speed=5.,QF = 1.,QR = 0.833, Q=30000.,\
B=2.,C=6.,AF=3.33,AR=6.66):

    #model coordinates and time discritisation
    TT= np.array([self.Convert_frame_totime(frame)])
    Node_ids = self.All_nodes_Frame.index
    TT = TT
    #time diemension
    n_t = len(TT)
    n_x = len(Node_ids)
    COORDs = self.All_coordinates_Frame.loc[Node_ids]
    XX = np.array(COORDs['Y']).reshape(n_x,1)+ np.zeros(n_t)
    YY = np.array(COORDs['Z']).reshape(n_x,1)+ np.zeros(n_t)
    ZZ = np.array(COORDs['X']).reshape(n_x,1)+ np.zeros(n_t)

    #center coordinates
    X0 = center[1]
    Y0 = center[2]
    Z0 = center[0]
    #angle to rad
    PI = np.pi
    AY = Angle * PI/180.
    #Shift coordinates
    XL = XX - X0
    YL = YY - speed * TT - Y0
    ZL = ZZ - Z0
    #project coordinates of shape (n_x,n_t)
    XD = np.sin( AY ) * ZL + np.cos( AY ) * XL
    ZD = np.cos( AY ) * ZL - np.sin( AY ) * XL
    YD = YL
    #print(XD.shape,YD.shape,ZD.shape)
    #efficiency factor
    QC = QR*np.ones((n_x,n_t))
    AC = AR*np.ones((n_x,n_t))
    QC[YD>0]= QF
    AC[YD>0]= AF
    #Heat source flux comptation Q*QC*exp(-(x/B)^2-(y/AC)^2-(z/c)^2)
    Flux = Q*QC*np.exp(-(XD/B)**2-(YD/AC)**2-(ZD/C)**2)
    ###Geometry constraint#####
    Flux[((ZZ>4.) & (XX>4.)) | ((ZZ<-4.) & (XX>4.))|((ZZ<-4.) & (XX<-4.))] = 0. #out of the component

    self.Flux_model = csr_matrix(Flux).reshape(n_x,n_t)
    self.Flux_model_list.append(self.Flux_model)
    del XD
    del YD
    del ZD
    del QC
    del AC
    print('\t\t Heat source is computed using the Goldak model!')

```

Figure D.3: Snippet of the heat source code.

D.3 Internal loads

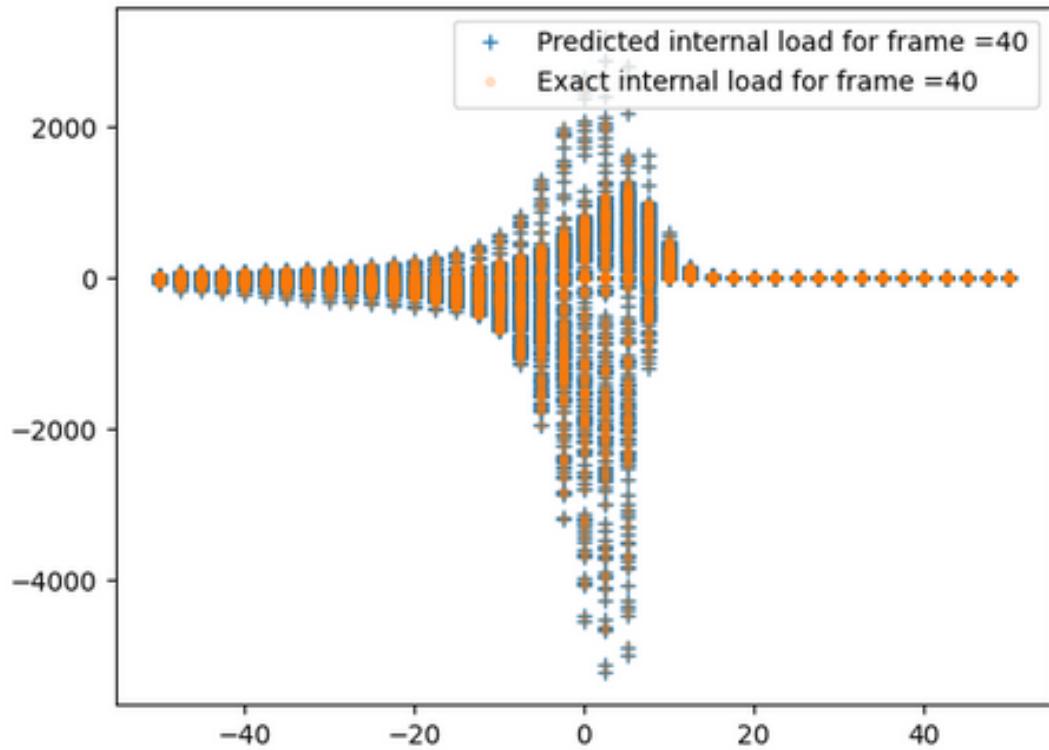


Figure D.4: The predicted and linearized internal loads of a training model over the moving axis at frame 40 (\sim half).

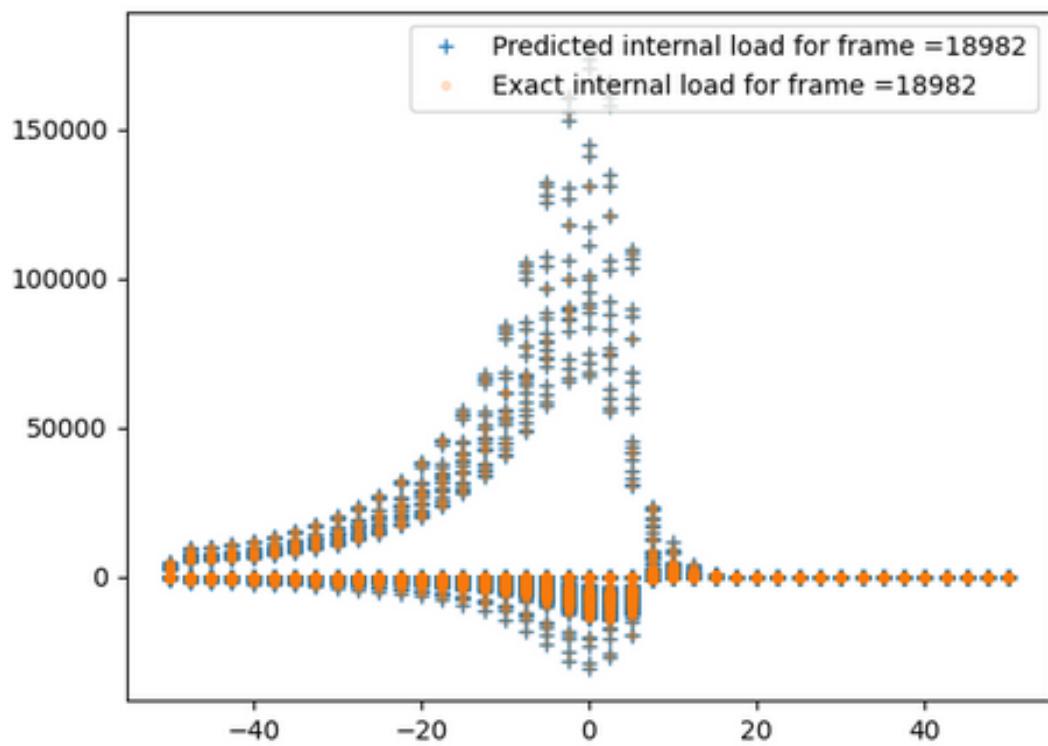


Figure D.5: The predicted and linearized internal loads of a validation model over the moving axis at frame 40 (\sim half).

Bibliography

- [1] American welding society. https://en.wikipedia.org/wiki/American_Welding_Society. Accessed: 2021-08-15.
- [2] Category:welding. <https://en.wikipedia.org/wiki/Category:Welding>. Accessed: 2021-08-15.
- [3] Fraunhofer institute for mechanics of materials webpage. https://en.wikipedia.org/wiki/Fraunhofer_Institute_for_Mechanics_of_Materials. Accessed: 2021-08-15.
- [4] Fraunhofer society description. https://en.wikipedia.org/wiki/Fraunhofer_Society. Accessed: 2021-08-15.
- [5] Generating thermal matrices. <https://abaqus-docs.mit.edu/2017/English/SIMACAEANLRefMap/simaanl-c-mtxgenerationperturbation.htm#:~:text=By%20default%2C%20the%20matrix%20generation%20procedure%20generates%20matrices%20from%20the,the%20matrix%20input%20contributions%20only>. Accessed: 2021-08-15.
- [6] Prof. Christophe Prud'Homme. Lectures material from <https://github.com/prudhomm>. Accessed: 2021-08-15.
- [7] Uncoupled heat transfer analysis. https://abaqus-docs.mit.edu/2017/English/SIMACAETHERefMap/simathe-c-uncoupledheat.htm#simathe-c-uncoupledheat-t-TimeIntegration-sma-topic5_simathe-c-uncoupledheat-eq7. Accessed: 2021-08-15.
- [8] Welding: Haz. https://en.wikipedia.org/wiki/Welding#Energy_beam. Accessed: 2021-08-15.
- [9] Welding joints. https://en.wikipedia.org/wiki/Welding_joint. Accessed: 2021-08-15.

- [10] F. J. Adams, R.A. Sobolev spaces. Elsevier Science, Oxford 2003.
- [11] S. Ambler. Just barely good enough models and documents: An agile best practice. <http://www.agilemodeling.com/essays/barelyGoodEnough.html>. Accessed: 2021-08-15.
- [12] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific J. Math., pages 1–3, 1966.
- [13] H. Brezis. Problemes unilateraux. J. Math. pure appl., 51:168, 1972.
- [14] D. Buckley. Solution of Nonlinear Transient Heat TransferProblems. Technical report, Florida International University, 2010.
- [15] G. Comini, S. D. Guidance, R. W. Lewis, and O. C. Zienkiewicz. Finite element solution on nolinear heat conduction problems with special reference to phase change. International Journal for Numerical Methods in Engineering, (8):613–624, 1974.
- [16] Courses (Oxford University). Finite element methods for pdes, 2019.
- [17] T. Daniel, F. Casenave, N. Akkari, and D. Ryckelynck. Model order reduction assisted by deep neural networks (rom-net). Advanced Modeling and Simulation in Engineering Sciences, 7, 12 2020.
- [18] S. Dempe. Bilevel optimization: theory, algorithms and applications. Optimization Online, 2018.
- [19] Dr. Vincent Vignon. Master 1 and 2, lectures, Retrieved from <https://irma-web1.math.unistra.fr/~vignon/liens/>, August 2021.
- [20] K. Eppler and F. Tröltzscher. Optimal Control Problems for the Nonlinear Heat Equation. Online Optimization of Large Scale Systems, 67:173–183, 2001.
- [21] L. C. Evans. Regularity properties for the heat equation subject to nonlinear boundary constraints. Nonlinear Analysis, 1(6):593–602, 1977.
- [22] J. Goldak, A. Chakravarti, and M. Bibby. A new finite element model for welding heat sources. Metal. Trans., 15B:299–305, 1984.
- [23] R. M. Gower. Convergence Theorems for Gradient Descent.
- [24] S. Gros and M. Diehl. Numerical Optimal Control (Draft). 2020.

- [25] N. Hempel, T. Nitschke-Pagel, K. Dilger, S. Moroz, and I. Varfolomeev. Berechnung von eigenspannungen in mehrlagenrohrschweißverbindungen und quantifizierung des einflusses auf die lebensdauer bei schwingbeanspruchung. Abschlussbericht, IGF-Vorhaben 17619 N der Forschungsvereinigung Schweißen und verwandte Verfahren e.V., 2016.
- [26] V. Karkhin, A. Pittner, C. Schwenk, and M. Rethmeier. Simulation of inverse heat conduction problems in fusion welding with extended analytical heat source models. (January 2014):1–8, 2011.
- [27] M. Knauer and C. Büskens. Hybrid Solution Methods for Bilevel Optimal Control Problems with Time Dependent Coupling. Recent Advances in Optimization and its Applications in Engineering. Springer, Berlin, Heidelberg, 2010.
- [28] J. L. LIONS and E. MACENES. Non-homogeneous Boundary Value Problems and Applications I. Springer, New York 1972.
- [29] S. Maddox. Fracture and Fatigue of Welded Joints and Structures. Woodhead Publishing Series in Welding and Other Joining Technologies, pages 168–207, 20.
- [30] M. Marion and R. Temam. Nonlinear Galerkin methods: The finite elements case. Numerische Mathematik, 57(1):205–226, 1990.
- [31] I. M. Martynenko. HEAT-CONDUCTION EQUATION. 79(3):26–29, 2006.
- [32] P. Mehlitz and G. Wachsmuth. Bilevel Optimal Control: Existence Results and Stationarity Conditions. Springer Optimization and Its Applications, 161:451–484, 2020.
- [33] M. A. Michael. Lifetime assessment of steel structures. (Head of Department, Energy and industry, consultancy), retrieved from <https://forcetechnology.com/en/services/lifetime-assessment>. Accessed: 2021-08-15.
- [34] J. T. Murphy. AN ELEMENTARY PROOF OF RADEMACHER'S THEOREM. III:117–128, 1985.
- [35] G. Phanikumar. Lesson on heat sources as part of the course on analysis and modelling of welding offered as a mooc from nptel, iit madras. <https://www.youtube.com/watch?v=cbfqHdcFroY>. Accessed: 2021-08-15.
- [36] A. Pittner. Doctoral thesis, Technische Universität Berlin, Fakultät V - Verkehrs- und Maschinensysteme, 2011.

- [37] Prof. Yannick Privat. Master 2, optimal control, Retrived from <https://irma.math.unistra.fr/~privat/cours/controleOpt.php>, August 2021.
- [38] G. Pólya. How to solve it, 2016.
- [39] W. Rudorffer, F. Dittmann, M. Wächter, I. Varfolomeev, and A. Esderts. Modellierung von schweißnähten zum nachweis der ermüdungsfestigkeit mit dem örtlichen konzept. Schlussbericht zum IGF-Vorhaben Nr. 20.025 N, 2021.
- [40] G. E. Schneider. A physical aproach to the finite-difference solution of the conduction equation in generalized coordinates. Numerical Heat Transfer, Part A: An international journal of computation and methodology, 1(5):1–19, 1982.
- [41] I. Varfolomeev, S. Moroz, D. Siegele, K. Kadau, and C. Amann. Study on fatigue crack initiation and propagation from forging defects. Metal. Trans.", Procedia Structural Integrity, 7:359–367, 2017.
- [42] J. Wittwer. Excel gantt chart template. <https://www.vertex42.com/ExcelTemplates/excel-gantt-chart.html>.
- [43] L. C. Wrobel and M. H. Aliabadi. Boundary Element Method. John Wiley and Sons, pages 1–19, 2002.