UKAEA-CCFE-PR(26)407

Wiera Bielajewa, Michelle Tindall, Perumal Nithiarasu

A novel, finite-element-based framework for sparse data solution reconstruction and multiple choices

A novel, finite-element-based framework for sparse data solution reconstruction and multiple choices

Wiera Bielajewa, Michelle Tindall, Perumal Nithiarasu

This is a preprint of a paper submitted for publication in
International Journal of Numerical Methods for Heat

A novel, finite-element-based framework for sparse data solution reconstruction and multiple choices

Wiera Bielajewa[a], Michelle Baxter (née Tindall)[b], Perumal Nithiarasu[a,*]

[a] *Zienkiewicz Institute for Modelling, Data and AI, Swansea University, Bay Campus, Swansea, SA1 8EN, United Kingdom (UK)*
[b] *Culham Science Centre, United Kingdom Atomic Energy Authority (UKAEA), Abingdon, OX14 3DB, United Kingdom (UK)*

Abstract

Digital twinning is gaining widespread popularity across various areas of engineering, and indeed it offers a capability of effective real-time monitoring and control, which are vital for cost-intensive experimental facilities, particularly the ones where extreme conditions exist. Sparse experimental measurements collected by various diagnostic sensors are usually the only source of information available during the course of a physical experiment. Consequently, in order to enable monitoring and control of the experiment (digital twinning), the ability to perform inverse analysis, facilitating the full field solution reconstruction from the sparse experimental data in real time, is crucial. Such solution reconstruction might be necessary to control a system, if a parameter to be controlled cannot be directly derived from the sparse measurements alone, as oftentimes is the case, for instance maximum temperature within a test piece.

This paper shows for the first time that it is possible to directly solve inverse problems, such as solution reconstruction, where some or all boundary conditions (BCs) are unknown, by purely using a finite-element (FE) approach, without needing to employ any traditional inverse analysis techniques or any machine learning models, as is normally done in the field. This novel and efficient FE-based inverse analysis framework employs

[*] Corresponding author

Email addresses: wiera.bielajewa@swansea.ac.uk (Wiera Bielajewa), michelle.baxter@ukaea.uk (Michelle Baxter (née Tindall)), p.nithiarasu@swansea.ac.uk (Perumal Nithiarasu)

a conventional FE discretisation, splits the loading vector into two parts corresponding to the known and unknown BCs, and then defines a loss function based on that split. In spite of the loading vector split, the loss function preserves the element connectivity. This function is minimised using a gradient-based optimisation; and the near real-time operation for heat conduction in a stainless steel plate is achieved.

Furthermore, this paper presents a novel modification of the aforementioned approach, which allows it to generate a range of different solutions satisfying given requirements in a controlled manner. Controlled multiple solution generation in the context of inverse problems and their intrinsic ill-posedness is a novel notion, which has not been explored before. This is done in order to potentially introduce the capability of semi-autonomous system control with intermittent human intervention to the workflow. Having access to a variety of feasible alternatives during the experiment can augment the human decision-making process and assist the operator in evaluating and selecting the most suitable course of action.

Keywords:
Solution reconstruction, inverse analysis, multiple choices, sparse data, digital twinning, finite element method

1. Introduction

Digital twins and digital twinning are becoming increasingly popular in various engineering fields, which is a trend driven by the desire to have a dynamic virtual representation of a physical system in order to extract as much useful real-time information as possible from the physical asset. A digital twin is a dynamic virtual representation of a physical system, which is continuously updated from real-time data in order to fully and accurately simulate the current behaviour of its physical counterpart [1]. Whereas, digital twinning refers to the process of creating and maintaining digital twins, which becomes important when the dynamic control of the physical system is introduced. When applied to the experimental facilities, digital twinning broadly aims to enhance the sparse experimental measurements collected from the various diagnostic sensors, i.e. provide system monitoring. It also aims to ensure the ability to reach the required experimental conditions within the optimal time frame, i.e. maintain system control. Therefore, the first step towards having a fully-fledged digital twinning process capable of fulfilling these two objectives involves developing an efficient inverse analysis framework, which would be able to directly integrate the experimental data into the simulation in real time in order to obtain a full solution. Full solution might be necessary to achieve desired system control, in cases when a parameter to be controlled cannot be directly derived from the sparse measurements alone, for example maximum temperature within a test piece. This paper presents an inverse analysis framework based on the Finite Element Method (FEM) as a compelling alternative to existing approaches.

In general, a forward problem can be defined as using the applicable model of a system to ascertain the effects of the given causes. Specifically, for the transient problems, the boundary conditions (BCs), the initial conditions (ICs), the material properties, and possibly other system parameters should be established to solve the forward problem. Contrarily, inverse problems can be classified into two categories:

(i) Deriving the system parameters from the observed causes and effects.
(ii) Deriving the causes from the observed effects.

The first category is a traditional definition of an inverse problem [2]; whereas, fundamentally, the second category is an inverse reconstruction of the full solutions using the available sparse data within a domain, i.e. measurements, with system parameters assumed to be known. The physical experiments represent the most common source of the sparse data observed within a domain. The main focus of this paper is on an inverse thermal field reconstruction, which is a task belonging to the second category. Such reconstruction is an important and often an essential part of digital twinning.

Consequently, the focus of this paper is twofold:

a. Developing a framework for inversely reconstructing the full temperature field from the sparse experimental measurements with speed as close to real time as possible.
b. Understanding how to generate multiple viable options satisfying given requirements, which could be used for semi-autonomous system control with a human-in-the-loop (HuIL) element [3].

These two points are discussed in some detail below.

From a historical perspective inverse problems focused largely on parameter estimation for differential equations [4, 5, 6], with some of the conventional methods being functional analytic regularisation as well as statistical regularisation [2, 4]. Possibly the most notable example of statistical regularisation is Bayesian inversion [4]. Another way to solve such inverse problems is to use a search-and-optimisation-based approach, for example the Particle Swarm Optimisation (PSO) algorithm [7]. More comprehensive reviews of the various methods to solve inverse problems are provided by Tamaddon-Jahromi et al. [8] and Arridge et al. [4]. However, the aforementioned methods tend to quickly become computationally intractable and generally lack sufficient flexibility.

Increasingly in many engineering fields, machine learning (ML) is being applied to everything with problems ranging from manufacturing [9, 10] to aerospace [11, 12]; and, indeed, it offers a combination of unique benefits, such as accuracy, efficiency, flexibility, and scalability [13], all of which are desirable in many engineering applications. Nevertheless, in the context of inverse analysis and digital twinning, ML models tend to have a few drawbacks.

Data-driven ML models usually require a significant amount of training data in order to provide sufficient accuracy and wide enough applicability. Although some ML model types, such as Gaussian Process Regression (GPR) [14], generally require far less training data than other ML models such as Neural Networks (NNs) [15, 16], Long Short-Term Memory (LSTM) [17], and Transformers [18, 19], still a significant amount of data is needed. Collecting significant amounts of data poses a considerable challenge for complex engineering applications. This issue can be resolved in two ways: (1) Use the experimental data for the training or (2) Run a high number of standard forward simulations, FEM simulations for instance, to generate the necessary training data [8, 18, 20, 21, 22]. The former is rarely possible in engineering, as the experimental data is usually sparse and there is not enough to be able to effectively train a data-driven ML model. The latter seems to be more promising on the surface; however, it assumes the possession of a completely accurate, verified, and trustworthy FEM model, which is also a complicated problem of its own to develop and verify, especially when dealing with extreme environments. Additionally, training ML model on selected cases of the forward simulations could bias the ML model to predict certain solutions, which might not necessarily match the physical experiment.

On the other hand, there are physics-based ML models, such as Physics-Informed Neural Networks (PINNs), which do not rely on training data, as they are trained using relevant partial differential equations (PDEs) [23, 24]. This resolves the issue of potential solution bias which data-driven models are susceptible to, but creates another issue. One of the primary reasons why ML is such an attractive option for digital twinning is its high inference speed after the model is trained, which could potentially allow it to function in real time. However, PINNs require the continuous training to be performed as the sensor measurements are inputted, which slows the model down considerably. Furthermore, the integration of such physics-based ML models within existing industry simulation workflows, which are typically developed using the conventional simulation software, is not a straightforward process.

It is generally accepted that it is very challenging to directly apply FEM to inverse problems, where some or all BCs are unknown, which traditionally limit its application to forward problems only. However, the developed FEM-based inverse analysis framework demonstrate that it is possible to modify the FE workflow in such a way as to allow it solve an inverse problem and accurately reconstruct a solution.

The secondary focus of the present work is to generate multiple viable solutions in a controlled manner. The challenge of inverse problems is that they are ill-posed, meaning that the solution is non-unique [2]. However, this difficulty can be harnessed and transformed into an advantage, as this ill-posedness is surprisingly desirable when it comes to the second objective of achieving control: it can be exploited to generate multiple options (solutions) for the user.

The remainder of the paper is organised into following sections. Section 2 details the method derivations for linear transient problem; moreover, it describes the extension of the method allowing the generation of multiple solution options fitting certain total energy criterion. Section 3 presents and interprets the results for several thermal field reconstruction cases; furthermore, several solution options are produced and analysed in order to demonstrate the application of the multiple solution generation process. Finally, Section 4 summarises the significance of the results in the context of the objectives set in Section 1, and also provides the main advantages of the presented framework.

2. Methodology

This section provides a description of the methodology. The framework introduced in the present work is based upon the ODIL (Optimizing a DIscrete Loss) approach, which was successfully applied to fluid flow problems [25] and glioma radiotherapy planning [26] and has undergone substantial adaptations for this paper. Additionally, the framework is adapted to ensure compatibility with the measurements from the physical experiments. Figure 1 outlines a general workflow with the details provided in the relevant sub-sections.

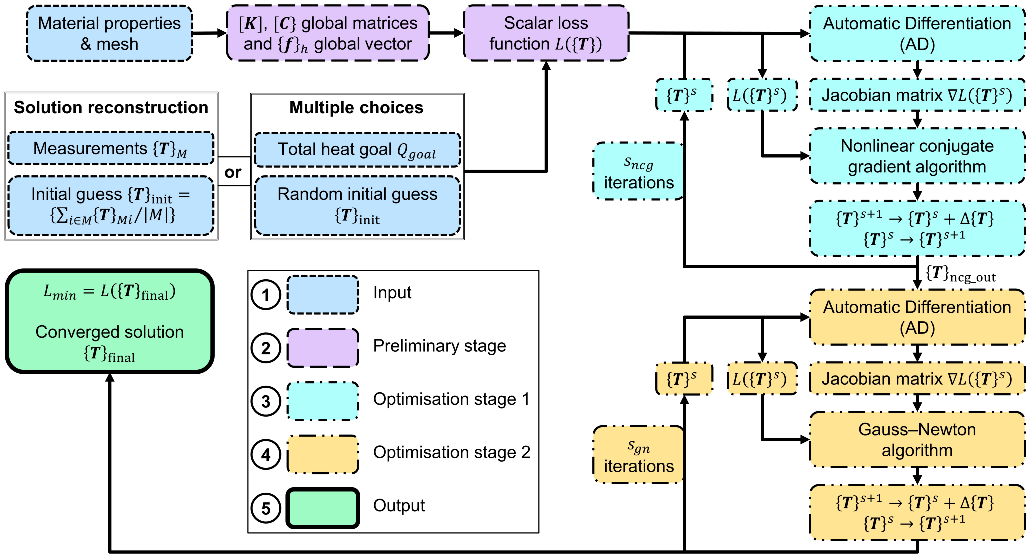
Material properties & mesh
$[K]$, $[C]$ global matrices and $\{f\}_h$ global vector
Scalar loss function $L(\{T\})$
Automatic Differentiation (AD)
$\{T\}^s$
$L(\{T\}^s)$
Jacobian matrix $\nabla L(\{T\}^s)$
Solution reconstruction
Measurements $\{T\}_M$
Initial guess $\{T\}_{\text{init}} = \{\sum_{i\in M}\{T\}_{Mi}/|M|\}$
or
Multiple choices
Total heat goal Q_{goal}
Random initial guess $\{T\}_{\text{init}}$
s_{ncg} iterations
Nonlinear conjugate gradient algorithm
$\{T\}^{s+1} \to \{T\}^s + \Delta\{T\}$
$\{T\}^s \to \{T\}^{s+1}$
$\{T\}_{\text{ncg\_out}}$
$L_{min} = L(\{T\}_{\text{final}})$
Converged solution $\{T\}_{\text{final}}$
1 Input
2 Preliminary stage
3 Optimisation stage 1
4 Optimisation stage 2
5 Output
Automatic Differentiation (AD)
$\{T\}^s$
$L(\{T\}^s)$
Jacobian matrix $\nabla L(\{T\}^s)$
s_{gn} iterations
Gauss–Newton algorithm
$\{T\}^{s+1} \to \{T\}^s + \Delta\{T\}$
$\{T\}^s \to \{T\}^{s+1}$

Figure 1: Summary of the workflow for the solution reconstruction and multiple solution generation. For the steady-state problem only $[\boldsymbol{K}]$ global matrix and $\{\boldsymbol{f}\}_h$ global vector are computed, and the whole procedure is repeated only once; whereas for the transient problem, it is repeated every time step and then $\{\boldsymbol{T}\}_{\text{final}}$ refers to the solution at one time step. For solution reconstruction the initial temperature distribution $\{\boldsymbol{T}\}_{\text{init}}$ is a uniform temperature equal to the average measurement value for steady state; $\{\boldsymbol{T}\}_{\text{init}}$ used for each time step is a uniform temperature equal to the average time-step measurement value.

2.1. Linear transient problem

The general linear 3D transient heat conduction equation without a heat source and with isotropic material properties is given as [27]:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right) = \rho c_p \frac{\partial T}{\partial t} \tag{1}$$

where x, y, and z are the spatial coordinates, T is the temperature, k is the thermal conductivity, while ρ and c_p are density and specific heat, respectively, which are assumed to be temperature-independent; t is time.

Additionally, general expressions describing Neumann BCs are:

$$
\begin{aligned}
k\frac{\partial T}{\partial x}l + k\frac{\partial T}{\partial y}m + k\frac{\partial T}{\partial z}n + q = 0 \quad \text{on} \quad \Gamma_q \\
k\frac{\partial T}{\partial x}l + k\frac{\partial T}{\partial y}m + k\frac{\partial T}{\partial z}n + h(T - T_a) = 0 \quad \text{on} \quad \Gamma_h
\end{aligned} \tag{2}
$$

where l, m, and n are the direction cosines of the surface normals, q is the time-dependent heat flux, while h is a convection heat transfer coefficient, T_a is ambient temperature, and Γ_q and Γ_h are the boundary surfaces where these BCs are applied. Dirichlet BCs are omitted in the present paper, since they are not physically representative of the experimental setting. Nonetheless, this approach can be used in conjunction with Dirichlet BCs.

The standard Galerkin weighted residual method is employed to discretise Eq. 1 and Eq. 2 in space [27]; and fully implicit time discretisation scheme is used. Consequently, the following global system of equations is obtained:

$$
[\boldsymbol{C}] \left\{ \frac{\{\boldsymbol{T}\}^{n+1} - \{\boldsymbol{T}\}^{n}}{\Delta t} \right\} + [\boldsymbol{K}] \{\boldsymbol{T}\}^{n+1} = \{\boldsymbol{f}\}^{n+1} \tag{3}
$$

with

$$
[\boldsymbol{K}] = \int_{\Omega} [\boldsymbol{B}]^{T} [\boldsymbol{D}] [\boldsymbol{B}] \, \mathrm{d}\Omega + \int_{\Gamma_h} h \, [\boldsymbol{N}]^{T} [\boldsymbol{N}] \, \mathrm{d}\Gamma \tag{4}
$$

$$
[\boldsymbol{C}] = \int_{\Omega} \rho c_p \, [\boldsymbol{N}]^{T} [\boldsymbol{N}] \, \mathrm{d}\Omega \tag{5}
$$

$$
\{\boldsymbol{f}\} = -\int_{\Gamma_q} q \, [\boldsymbol{N}]^{T} \, \mathrm{d}\Gamma + \int_{\Gamma_h} h T_a \, [\boldsymbol{N}]^{T} \, \mathrm{d}\Gamma \tag{6}
$$

where $\{\boldsymbol{T}\}$ and $\{\boldsymbol{N}\}$ are the temperature and shape function vectors, respectively. $[\boldsymbol{C}]$ and $[\boldsymbol{K}]$ are the global mass and stiffness matrices, respectively, and $\{\boldsymbol{f}\}$ is a global loading vector. Superscript n signifies the n^{th} time step; Ω is the discretised domain.

The loading term $\{\boldsymbol{f}\}$ can be split into two following terms corresponding to the known (convection) and unknown (applied heat flux) BCs - global convection $\{\boldsymbol{f}\}_h$ and applied heat flux loading vectors $\{\boldsymbol{f}\}_q$, respectively:

$$
\begin{aligned}
\{\boldsymbol{f}\} &= \{\boldsymbol{f}\}_q + \{\boldsymbol{f}\}_h \\
\{\boldsymbol{f}\}_q &= -\int_{\Gamma_q} q\,[\boldsymbol{N}]^T\,\mathrm{d}\Gamma \\
\{\boldsymbol{f}\}_h &= \int_{\Gamma_h} hT_a\,[\boldsymbol{N}]^T\,\mathrm{d}\Gamma
\end{aligned} \tag{7}
$$

Eq 7 represents a start of the novel deviation of the developed inverse analysis approach from the standard FEM workflow used for forward problems. The global applied heat flux vector $\{\boldsymbol{f}\}_q$ is assumed be unknown.

2.2. Inverse analysis

The following assumptions are made for the inverse analysis:

- The temperature values are known at points belonging to the measurement set $M \in \Omega$.
- The materials properties are known; they include the general thermal conductivity matrix $[\boldsymbol{D}]$ and the convection heat transfer coefficient h.
- The ambient temperature T_a is known.
- The exact locations of Γ_h and Γ_q on the boundary are known, and $\Gamma = \Gamma_h \cup \Gamma_q$, where Γ is domain (Ω) boundary.

Given that M number of temperatures $\{\boldsymbol{T}\}_M$ are known from measurements and the heat flux value on the boundary Γ_q is unknown, the next step is to define a scalar loss function L of $\{\boldsymbol{T}\}^{n+1}$. The dependency of $\{\boldsymbol{f}\}_q$ on the temperature distribution $\{\boldsymbol{T}\}$ is as follows (Eqs. 3 and 7):

$$
\{\boldsymbol{f}\}_q^{n+1} = [\boldsymbol{C}]\left\{\frac{\{\boldsymbol{T}\}^{n+1} - \{\boldsymbol{T}\}^n}{\Delta t}\right\} + [\boldsymbol{K}]\{\boldsymbol{T}\}^{n+1} - \{\boldsymbol{f}\}_h^{n+1} \tag{8}
$$

Then $L\left(\{\boldsymbol{T}^{n+1}\}\right)$ is a sum of three terms:

(a) Residual term
$c_1 \sum_{i\in\Omega\setminus\Gamma_q}\left([\boldsymbol{C}]_i\left\{\frac{\{\boldsymbol{T}\}^{n+1}-\{\boldsymbol{T}\}^n}{\Delta t}\right\} + [\boldsymbol{K}]_i\{\boldsymbol{T}\}^{n+1} - \{\boldsymbol{f}\}_{hi}^{n+1}\right)^2$
where $[\boldsymbol{C}]_i$ and $[\boldsymbol{K}]_i$ are the i^{th} rows of the global $[\boldsymbol{C}]$ and $[\boldsymbol{K}]$ matrices,

respectively, which correspond to the i^{th} node in the domain; while $\{\boldsymbol{f}\}_{hi}$ is the i^{th} element in the global $\{\boldsymbol{f}\}_h$ vector. This term enforces the fact that, in the absence of the volumetric heating, the elements of loading vector $\{\boldsymbol{f}\}_q$ corresponding to the nodes inside the domain and the nodes where only the convection is applied should be zero (see Appendix A).

(b) Measurement term $c_2 \sum_{i \in M} \left(\{\boldsymbol{T}\}_i^{n+1} - \{\boldsymbol{T}\}_{Mi}^{n+1} \right)^2$
This term directly incorporates the temperature measurements into the simulation.

(c) Regularisation (smoothing) term

$$c_3 \left[\sum_{i \in \Gamma_q \backslash \Gamma_{edge}} \left(\{\boldsymbol{f}\}_{qi}^{n+1} - \frac{\sum_{i \in \Gamma_q \backslash \Gamma_{edge}} \{\boldsymbol{f}\}_{qi}^{n+1}}{\left| \Gamma_q \backslash \Gamma_{edge} \right|} \right)^2 + \right.$$

$$+ \sum_{i \in \Gamma_{edge} \backslash \Gamma_{corners}} \left(\{\boldsymbol{f}\}_{qi}^{n+1} - \frac{\sum_{i \in \Gamma_{edge} \backslash \Gamma_{corners}} \{\boldsymbol{f}\}_{qi}^{n+1}}{\left| \Gamma_{edge} \backslash \Gamma_{corners} \right|} \right)^2 +$$

$$\left. + \sum_{i \in \Gamma_{corners}} \left(\{\boldsymbol{f}\}_{qi}^{n+1} - \frac{\sum_{i \in \Gamma_{corners}} \{\boldsymbol{f}\}_{qi}^{n+1}}{\left| \Gamma_{corners} \right|} \right)^2 \right]$$

$\{\boldsymbol{f}\}_q^{n+1}$ is calculated using Eq. 8. Γ_{edge} represents the 3D edge of the surface where the heat flux q is applied, i.e. it is $\partial \Gamma_q$ and also, referring to the last point in the aforementioned assumptions list, it is equal to $\Gamma_q \cap \Gamma_h$. $\Gamma_{corners}$ represents sharp features on Γ_{edge} if such features exist and if they are discretised using one element, i.e the node on a corner belongs only to one element. An example of Γ_{edge} and $\Gamma_{corners}$ is given in Table 2. This term ensures that there is a smooth temperature transition on the boundary of the domain where the heat flux q is applied.

In the above equations, c_1, c_2, and c_3 are weighting coefficients signifying the relative importance of each term; the selection of the appropriate values is discussed in Section 3. Mesh, as well as $[\boldsymbol{C}]$, $[\boldsymbol{K}]$, and $\{\boldsymbol{f}\}_h$ can be obtained using virtually any commercial or open-source FEM software, such as ANSYS [28] or Code_Aster [29]; the latter is used in this paper. The component in the form of $[\boldsymbol{C}] \left\{ \left(\{\boldsymbol{T}\}^{n+1} - \{\boldsymbol{T}\}^n \right) / \Delta t \right\}$, which signifies heat accumulation over time, becomes zero for steady-state problems.

The regularisation (smoothing) term prevents unphysically large and abrupt temperature variations on the surface from appearing. It does so through indirectly ensuring the smooth spatial variation of temperature's first derivative normal to the surface. The first derivative of temperature in the direction normal to the boundary directly correlates with the heat flux through the boundary $q = -k(\partial T/\partial n)$ [27], which in turn defines the applied heat flux loading vector $\{\boldsymbol{f}\}_q$. The regularisation term limits the deviation of each element of $\{\boldsymbol{f}\}_q$ from the mean value of all $\{\boldsymbol{f}\}_q$ elements. It is necessary to apply the regularisation term only to Γ_q, where the implicit degrees of freedom in the form of the unknown applied heat flux are situated.

The regularisation term is split into three parts: the first one corresponding to the nodes on $\Gamma_q \setminus \Gamma_{edge}$, the second one corresponding to $\Gamma_{edge} \setminus \Gamma_{corners}$, and the third one to $\Gamma_{corners}$. When the applied heat flux varies gradually on Γ_q relative to the finite element size, a rapid change in the loading vector $\{\boldsymbol{f}\}_q$ values usually occurs between the nodes belonging to $\Gamma_q \setminus \Gamma_{edge}$, the nodes belonging to $\Gamma_{edge} \setminus \Gamma_{corners}$ and $\Gamma_{corners}$ if such nodes exist. Therefore, in order to avoid the complications related to this sudden transition, it is beneficial to separate the regularisation term into aforementioned groups. Four-node quadrilateral (linear) elements on Γ_q are used in conjunction with the regularisation term defined above; however, a slightly different node grouping within this loss function term might be necessary if three-point (linear) triangles, particularly unstructured ones, are to be used for Γ_q surface discretisation. Additionally, certain adjustments would need to be made for the meshes with mesh refinement on Γ_q in order to account for the potential presence of varying element size.

Lastly, it should be noted that such regularisation term definition might not necessarily be an optimal choice for all potential heat flux distributions. However, considering a specific experimental arrangement, reasonable assumptions usually can be made regarding a general shape of the heat flux distribution produced by a given heat-generating element; meaning that some knowledge can be derived regarding the relative magnitudes of heat fluxes without knowing their specific values. For example, if the heat flux is produced by an induction coil of known geometry located at a certain position relative to a test piece, then some plausible assumptions can be made regarding the approximate locations of regions of the maximum and minimum heat fluxes. This information can subsequently be used to customise the regularisation term.

In order to select the correct distribution out of the infinite number of possible distributions fitting the residual term, the measurement and regularisation terms are employed in conjunction, thus making the inverse problem approximately well-posed. The heat flux associated with a particular temperature distribution is implicitly calculated within the regularisation term using the node equations excluded from the residual term (part (a) of the loss function and Appendix A) to obtain the loading vector values corresponding to the nodes on Γ_q.

While the current paper utilises only linear analysis, the extension of this framework to nonlinear material properties naturally follows. It is slightly simplified by the fact that unlike stress-strain curve, thermal material properties, k, ρc_p, and h, tend to steadily increase with temperature without displaying a softening region. Therefore, the potential additional solution non-uniqueness stemming purely from the nonlinear material properties [30] might be avoided.

2.3. Minimisation

After the loss function $L\left(\{\boldsymbol{T}\}^{n+1}\right)$ is defined, the main objective becomes the minimisation of L, and $\{\boldsymbol{T}\}$ corresponding to the minimum of L would be the solution for the $n+1$ time step, i.e. the reconstructed temperature field at this time step. It should be emphasised that the minimum of L might not be necessarily zero, as, depending on its definition, the regularisation term might not be required to be exactly zero for the correct solution. The function minimisation is a common optimisation problem, but it is a cornerstone in the ML training process, and hence it is advantageous to borrow some of the methods frequently used in ML to minimise L.

Figure 1 summarises a general workflow for the solution reconstruction for each time step. In this paper two iterative gradient-based optimisation algorithms are used consecutively: the first one is nonlinear conjugate gradient method (Stage 1) for s_{ncg} iterations [31] and the second one is Gauss–Newton method (Stage 2) for s_{gn} iterations [32, 33]. The output from the nonlinear conjugate gradient method $\{\boldsymbol{T}\}_{\text{ncg\_out}}$ is an input to Gauss–Newton method. The nonlinear conjugate gradient method utilises a first-order derivative matrix (Jacobian matrix) $\nabla L\left(\{\boldsymbol{T}\}\right)$, whilst the Gauss–Newton method makes use of the approximation of the second-order derivative matrix (Hessian matrix) $\nabla^2 L\left(\{\boldsymbol{T}\}\right)$ based on $\nabla L\left(\{\boldsymbol{T}\}\right)$. Similar to the ML training process, Automatic Differentiation (AD) is employed in order to efficiently calculate the exact Jacobian matrix at each iteration [25, 34]. The initial temperature distributions used for each time step is chosen to be a uniform temperature equal to the average time-step measurement value. At the beginning of each iteration $L\left(\{\boldsymbol{T}\}^s\right)$ and $\nabla L\left(\{\boldsymbol{T}\}^s\right)$ are calculated for the temperature distribution outputted by the previous iteration $\{\boldsymbol{T}\}^s$, then $\Delta\{\boldsymbol{T}\}$ is computed using either nonlinear conjugate gradient method (during

Stage 1) or Gauss–Newton method (during Stage 2). Finally, the temperature distribution for the next iteration becomes $\{\boldsymbol{T}\}^{s+1} = \{\boldsymbol{T}\}^{s} + \Delta\{\boldsymbol{T}\}$.

An iterative minimisation of the residual term combined with the measured temperature values and the regularisation term will provide a temperature distribution over the entire domain including boundaries. The temperature distribution along the heat flux boundary Γ_q can now be used to compute the heat flux.

To ensure a high computational speed the majority of matrix additions, multiplications, as well as AD are performed on Graphical Processing Unit (GPU) with the help of PyTorch, which is a highly optimised ML library [35].

2.4. Multiple choice generation

In the previous sub-sections, the regularisation term is used to overcome the inherent ill-posedness of the inverse problem ensuring that the loss function converges to the correct solution; whereas this section demonstrates how ill-posedness can be exploited in order to generate multiple solutions fitting certain predefined requirements. A scenario can be considered where some desired values of the time-dependent total heat $Q_{\text{goal}}(t)$ should be achieved, which can be done by applying the heat flux on Γ_q in various ways. It is assumed that the measurements are not available, as the aim of this procedure is to produce potential future options which can be used for control, rather than to reconstruct the correct solution from the obtained measurements.

For the transient problem, the loss function is defined as a sum of the following three terms, two of which have been used previously for the solution reconstruction:

1. Residuals term
   $$c_1 \sum_{i\in\Omega\setminus\Gamma_q} \left([\boldsymbol{C}]_i \left\{\frac{\{\boldsymbol{T}\}^{n+1}-\{\boldsymbol{T}\}^{n}}{\Delta t}\right\} + [\boldsymbol{K}]_i \{\boldsymbol{T}\}^{n+1} - \{\boldsymbol{f}\}_{hi}^{n+1}\right)^2.$$
2. Regularisation (smoothing) term
   $$c_3 \Bigg[\sum_{i\in\Gamma_q\setminus\Gamma_{edge}} \left(\{\boldsymbol{f}\}_{qi}^{n+1} - \frac{\sum_{i\in\Gamma_q\setminus\Gamma_{edge}}\{\boldsymbol{f}\}_{qi}^{n+1}}{\left|\Gamma_q\setminus\Gamma_{edge}\right|}\right)^2 +$$
   $$+ \sum_{i\in\Gamma_{edge}\setminus\Gamma_{corners}} \left(\{\boldsymbol{f}\}_{qi}^{n+1} - \frac{\sum_{i\in\Gamma_{edge}\setminus\Gamma_{corners}}\{\boldsymbol{f}\}_{qi}^{n+1}}{\left|\Gamma_{edge}\setminus\Gamma_{corners}\right|}\right)^2 +$$
   $$+ \sum_{i\in\Gamma_{corners}} \left(\{\boldsymbol{f}\}_{qi}^{n+1} - \frac{\sum_{i\in\Gamma_{corners}}\{\boldsymbol{f}\}_{qi}^{n+1}}{|\Gamma_{corners}|}\right)^2 +$$

$$+ \left(\frac{\sum_{i \in \Gamma_{edge} \setminus \Gamma_{corners}} \{\boldsymbol{f}\}_{qi}^{n+1}}{\left|\Gamma_{edge} \setminus \Gamma_{corners}\right|} - 2 \cdot \frac{\sum_{i \in \Gamma_{corners}} \{\boldsymbol{f}\}_{qi}^{n+1}}{|\Gamma_{corners}|} \right) \Bigg] .$$

The dependency of $\{\boldsymbol{f}\}_q^{n+1}$ on the temperature distribution $\{\boldsymbol{T}\}$ is is given by Eq. 8.

3. Total heat term $c_4 \left(\sum_{i \in \Gamma} \left([\boldsymbol{C}]_i \left\{ \frac{\{\boldsymbol{T}\}^{n+1} - \{\boldsymbol{T}\}^n}{\Delta t} \right\} + [\boldsymbol{K}]_{\text{base\_}i} \{\boldsymbol{T}\}^{n+1} \right) - Q_{\text{goal}}^{n+1} \right)^2$

where

$$[\boldsymbol{K}]_{\text{base}} = \int_{\Omega} [\boldsymbol{B}]^T [\boldsymbol{D}] [\boldsymbol{B}] \, \mathrm{d}\Omega \tag{9}$$

and $[\boldsymbol{K}]_{\text{base\_}i}$ is the i^{th} row of the $[\boldsymbol{K}]_{\text{base}}$ matrix.

The total heat term serves to enforce the required total heat values for the solution as it progresses through time. c_4 represents the relative importance of the total heat specification; a lower c_4 value means the generated solution is allowed to deviate from the desired total heat values more, and vice versa. It can be used to control the type of the solution generated. The randomisation of the initial temperature values allows the generation of different solutions for the same c_4 value, which is explained in greater detail in this section.

The regularisation term includes an additional fourth component, which limits the magnitude of the deviation between the average value of $\{\boldsymbol{f}\}_q$ elements on $\Gamma_{edge} \setminus \Gamma_{corners}$ and the average value of $\{\boldsymbol{f}\}_q$ elements on $\Gamma_{corners}$. The separation of the regularisation term into three parts, which is beneficial for the solution reconstruction, becomes less effective when generating multiple solutions, as there is no measurement term creating a connection between the three regions. Therefore, if the fourth component is absent from the regularisation term, the generated solutions tend to exhibit sharp temperature variations between the corners and the rest of Γ_q. The fourth component serves to constrain the applied heat flux on the corners without overly constraining the problem overall, so that it becomes well-posed.

Figure 1 summarises a general workflow for generating multiple solutions, here the total heat term replaces the measurement term used for the solution reconstruction. A similar approach of combining two gradient-based optimisation algorithms, conjugate gradient and Gauss–Newton (Subsection 2.3), is adopted for the loss function minimisation. The only difference with the solution reconstruction process is the initial temperature distributions used for each time step. In theory, any gradient-free optimisation algorithm can be used as a wrapper for the two aforementioned gradient-based algorithms in order to find multiple local minima of the loss function landscape, and basin-hopping [36] was initially trialled for this purpose. Nevertheless, just randomised assignment of the temperature values at each node works well while keeping the adherence to the runtime specification for the digital twinning. The random initial temperature values are produced using probability density function of the uniform distribution over the half-open interval $[T_{min}, T_{max})$ [37]. A new random set of values for the initial guess vector $\{\boldsymbol{T}\}_{\text{init}}$ is produced for every time step using the same T_{min} and T_{max} values for the whole run; the random $\{\boldsymbol{T}\}_{\text{init}}$ corresponding to each time step are generated prior to the run. It should be noted that T_{min} and T_{max} values limit only the initial guess, and they do not limit the final converged temperature distribution.

Finally, while only the total heat requirement is considered in the present work, it is easy enough to create a loss function term enforcing any other given requirement, such as the required maximum temperature on Γ_q.

3. Results and discussion

The example considered in this paper is an air-cooled stainless steel plate subjected to the linearly increasing uniform surface heating given by:

$$q(t) = \frac{600000}{180} t \tag{10}$$

Figure 2 shows a general schematic of the plate with the applied BCs described in Tables 1 and 2, while Table 3 provides the material properties. This particular geometry and BC types are selected as they represent a generally realistic experimental setup. The simulation is run for 180s; the initial conditions represent a steady state, room temperature of 20°C. For this case Γ_{edge} is ABCD edge shown in Figure 2 and also equal to $\Gamma_q \cap \Gamma_h$; $\Gamma_{corners}$ consists of the nodes A, B, C, and D (Table 2).

3.1. Thermal field reconstruction

The reference solutions, to which the inversely reconstructed solutions are compared, are generated in Code_Aster [29] by applying the BCs shown in Table 1. In order to inversely reconstruct the solution from the sparse measurements, the steady-state loss function is firstly defined (Subsection 2.2) and used to reconstruct the steady-state solution, which is

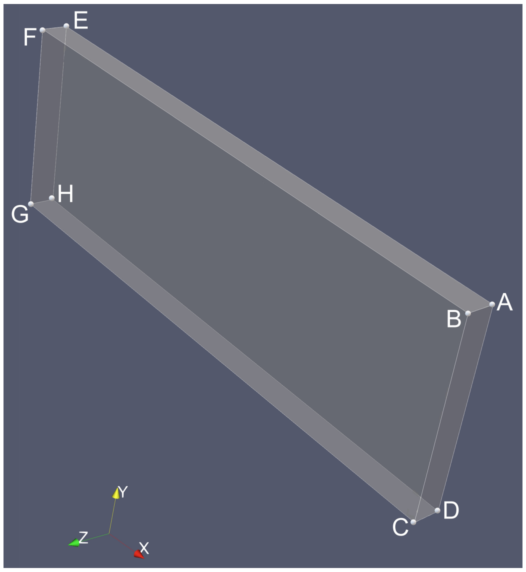

Figure 2: Labels for the applied BCs (Table 1).

Table 1: Applied BCs; Figure 2 shows the labels.

Surface	**BC**
ABCD	Uniform heat flux
BFGC BFEA AEHD DHGC EFGH	Plate-air convection

Table 2: Defined sets; Figure 2 shows the labels.

Set	**Location**
Γ_q	Surface ABCD
Γ_{edge}	Edge ABCD
$\Gamma_{corners}$	Nodes A, B, C, D

utilised as a starting point for the subsequent transient calculations. Then, the transient loss function can be defined (Sub-section 2.2) and used to reconstruct the solution at each time step.

Table 4 gives the descriptions of the six cases considered in the present sub-section. Δt_{ref} and Δt_{rec} are time step sizes used for generating the reference solution in Code_Aster and for the solution reconstruction process, respectively.

Parameters c_1, c_2 are equal to 1.0 for the solution reconstruction; conversely, c_3 might need a separate adjustment, the process of which is detailed in Appendix B. The locations of c_3 value regions resulting in the lowest relative errors seem to be dependent on the number of measurements as well as on Δt_{rec}; consequently, it can be calibrated prior to the experiment. Moreover, the only meaningful values c_3 might adopt are between 0.0 and 1.0, as any value above 1.0 means that c_1 and c_2 can be adjusted in order to keep c_1, c_2, and c_3 below 1.0. Generally, values of c_3 above 0.5 seem to yield acceptable errors.

One structured mesh consisting of 5,196 linear quadrilateral and hexahedral elements (for surface and volume discretisation, respectively) is used for the reference solutions as well as for all six cases of the solution reconstruction. During the calculation of elemental matrices and vectors, four Gauss (integration) points are used for the surface integration of quadrilaterals, and eight Gauss points are used for the volume integration of hexahedrals. No mesh refinement is used in the present paper. Two options for the measurement placement are considered: 15 and 9 measurements; Figure 3 shows the locations for each one. Furthermore, these measurements are experimentally attainable when thermocouple (TC) measurements and

Table 3: Material properties.

Parameter	**Value**	**Unit**
Thermal conductivity k (stainless steel Grade 91)	25.84	$W/(m\,^{\circ}\mathrm{C})$
Air-steel convection heat transfer coefficient h	135.00	$W/(m^2\,^{\circ}\mathrm{C})$
Density ρ (stainless steel Grade 91)	7760.00	$kg/(m^3)$
Specific heat c_p (stainless steel Grade 91)	416.80	$J/(kg\,^{\circ}\mathrm{C})$
Atmospheric temperature T_a	20.00	°C

* All values are given for the room temperature of 20°C.

possibly infrared (IR) camera surface recordings are available [38, 39]. None of the measurements are located directly on $\Gamma_q \setminus \Gamma_{edge}$, as it is generally not possible to place any diagnostic sensors directly under the heat-generating element, for example an induction coil [38]. Furthermore, none of the measurements are placed close to EFGH side (Figure 2), since, with the location of the heating element being known, this region would not provide any useful information for a significant part of the simulation in the beginning as the temperatures measured there would simply remain nearly constant.

The accuracy is judged by calculating the errors of the reconstructed solution relative to the reference solution, while the computational speed is evaluated by recording the time it takes to reconstruct the temperature field at each time step, i.e. the time step runtime. Table 5 summarises the overall relative and absolute errors averaged in space and time as well as the overall maximum relative and absolute errors for each case, whereas Figure 4 compares the average time step runtime for local GPU (NVIDIA GeForce RTX 3060) and supercomputer GPU (NVIDIA A100) with the time step size Δt_{rec} for each case. Table 5 also provides relative and absolute differences between two reference solutions used in this paper, the first one with Δt_{ref} equal to 1s and the second one with Δt_{ref} equal to 0.1s. Moreover, Figures 5

Table 4: Description of six cases considered for the stainless steel plate.

Case No.	Δt_{ref} [s] [a]	Δt_{rec} [s] [b]	No. of measurements	s_{ncg} [c]	s_{gn} [d]	c_1	c_2	c_3
1	1.0	1.0	15	2	1	1.0	1.0	1.0
2	1.0	1.0	9	2	1	1.0	1.0	1.0
3	0.1	1.0	15	2	1	1.0	1.0	1.0
4	0.1	1.0	9	2	1	1.0	1.0	1.0
5	0.1	2.0	15	2	1	1.0	1.0	1.0
6	0.1	2.0	9	2	1	1.0	1.0	0.5

[a] Δt_{ref} is a time step size used for generating the reference solution in Code_Aster.

[b] Δt_{rec} is a time step size used for the solution reconstruction process.

[c] s_{ncg} is a number of nonlinear conjugate gradient algorithm iterations (Figure 1).

[d] s_{gn} is a number of Gauss–Newton algorithm iterations (Figure 1).

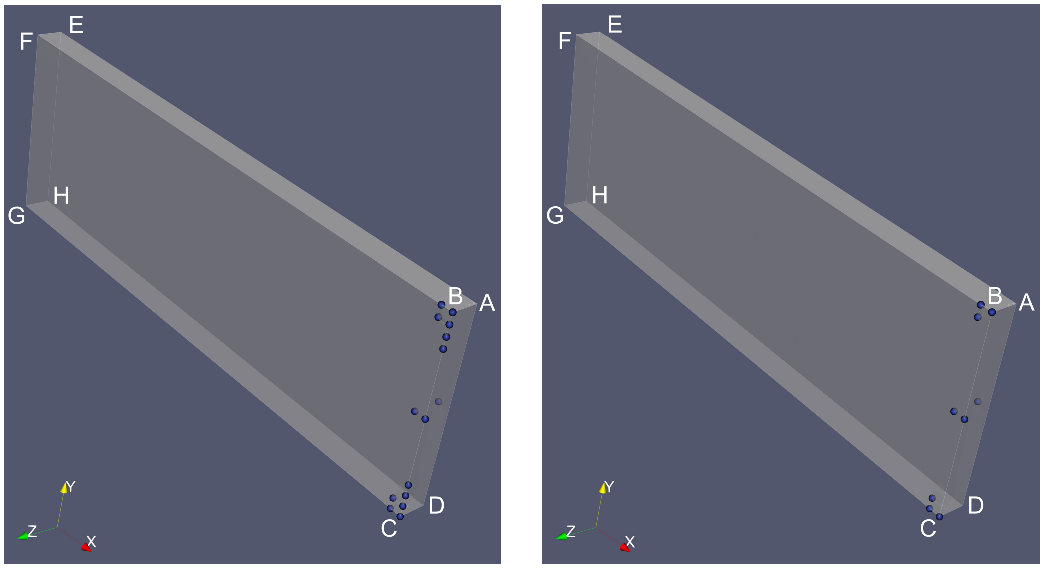

Figure 3: Two options for the measurement placement considered: 15 (left) and 9 (right) measurements.

and 6 provide a more detailed view of the relative and absolute errors as they develop with time for Cases No. 1 and 2, respectively; the errors on these figures are averaged in space only. Similar figures for Cases No. 3-6 can be found in Appendix C. Since all calculations are performed on the same mesh, the space averaging is simply done using the corresponding nodal temperature values.

3.1.1. Number of measurements and errors

Firstly, the correlation between the number of measurements and the error distribution is considered. The six cases can be divided into three pairs, in which all parameters are the same apart from the number of measurements, excluding c_3 in some cases as it is calibrated based on the number of measurements and Δt_{rec} (Appendix B): Cases No. 1 and 2, Cases No. 3 and 4, and Cases No. 5 and 6 (Table 4). Unsurprisingly, the overall maximum relative errors increase with the decrease in the number of measurements for Cases No. 1 and 2, Cases No. 3 and 4 (Table 5); on the other hand, for Cases No. 5 and 6 the maximum relative error decreased by 0.08% with the decrease in the number of measurements.

Figures 7 and 8 display the spatial relative error distributions for the time instance with the maximum overall relative error for Cases No. 1 and 2, respectively; the time instance is 63s for both of them. It can be seen that the areas with the highest errors are around $\Gamma_q \setminus \Gamma_{edge}$ (Figure 2, Table 2), which is to be expected as no measurements are available there and no relationship between T and q is known a priori. However, it can be observed that the errors are still within the 2% range, and Figures 7 and 8 show an excellent match between the reconstructed and reference solutions.

As shown in Figures 5 and 6, as well as in Figures C.17 and C.18, and Figures C.19 and C.20 in Appendix C, the average and maximum relative errors do not increase continuously with time, but in fact peak at the time instance between 0 and 180s and then start to gradually decline. This signifies that the measurements do provide a limiting effect on the error accumulation in time.

Table 5: Average and maximum relative and absolute solution reconstruction errors for six cases considered for the stainless steel plate.

Case No.	Average relative error [%]	Maximum relative error [%]	Average absolute error [°C]	Maximum absolute error [°C]
1	0.09	1.84	0.05	2.56
2	0.09	2.01	0.05	2.78
3	0.14	1.77	0.06	2.54
4	0.14	1.98	0.06	2.83
5	0.21	1.75	0.08	1.88
6	0.21	1.67	0.09	2.58
Ref. diff. [a]	0.15	1.33	0.07	1.0

[a] Reference difference row - this row corresponds to the relative and absolute difference between the two reference solutions, the first one with Δt_{ref} equal to 1s and the second one with Δt_{ref} equal to 0.1s.

3.1.2. Δt_{ref}, Δt_{rec}, and errors

The second set of interesting results is related to the time step ratio (TSR), which can be defined as follows:

$$\text{TSR} = \frac{\Delta t_{\text{rec}}}{\Delta t_{\text{ref}}} \tag{11}$$

For Cases No. 1 and 2 TSR is equal to 1.0, for Cases No. 3 and 4 it is equal to 10, while for Cases No. 5 and 6 to 20; six cases can be divided into two groups, in which all parameters are the same apart from the TSR: Cases No. 1, 3, 5 with 15 measurements, and Cases No. 2, 4, 6 with 9 measurements (Table 4). Figure 9 visualises the results from Table 5 for these two groups. It is evident that the average and maximum errors increase with a much lower rate than the corresponding TSR, especially considering the relative difference between two reference solutions used in this paper, the first one with Δt_{ref} equal to 1s and the second one with Δt_{ref} equal to 0.1s. There is 0.15% average and 1.33% maximum relative difference between two reference solutions (Table 5); however, it is apparent from Figure 9 that errors do not increase by nearly these amounts as TSR progresses from 1 to 20, indeed the

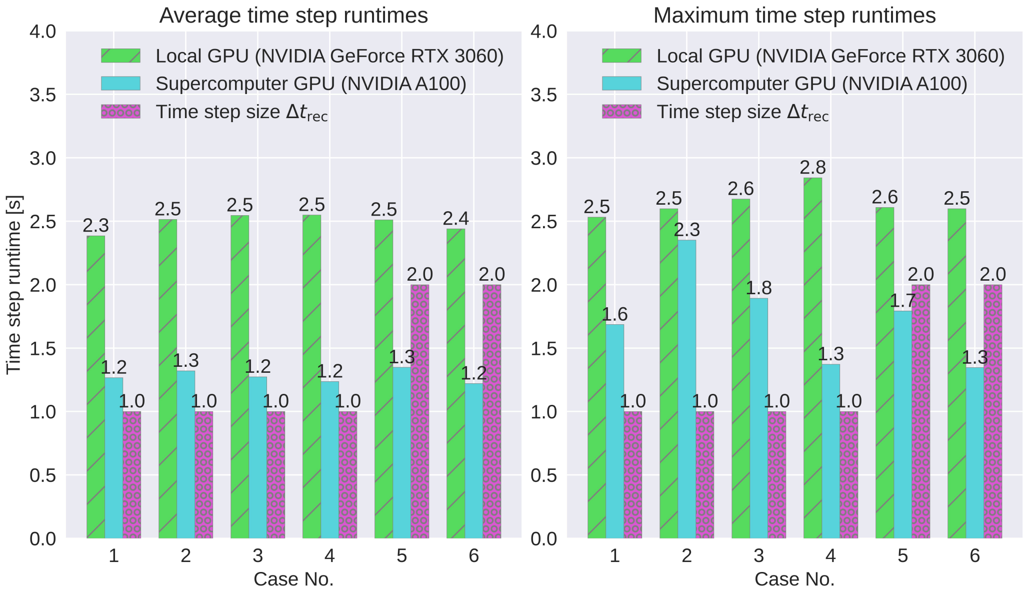

Figure 4: Average and maximum time step runtimes for all cases, compared with the time step size Δt_{rec} used; one GPU is used for the local and supercomputer calculations.

maximum relative error remain nearly constant. A tenfold increase in TSR resulted in 1.6 times increase in relative average error; however, the maximum relative errors decrease with increasing TSR for both sets of measurements. This observation is highly beneficial for two reasons:

1. It is yet another proof that the measurements incorporated into the algorithm provide a limiting effect on the solution reconstruction errors.
2. The time step size can be increased to decrease the runtime, to meet real time requirements, since it can be reasonably increased with only minor effect on the reconstruction.

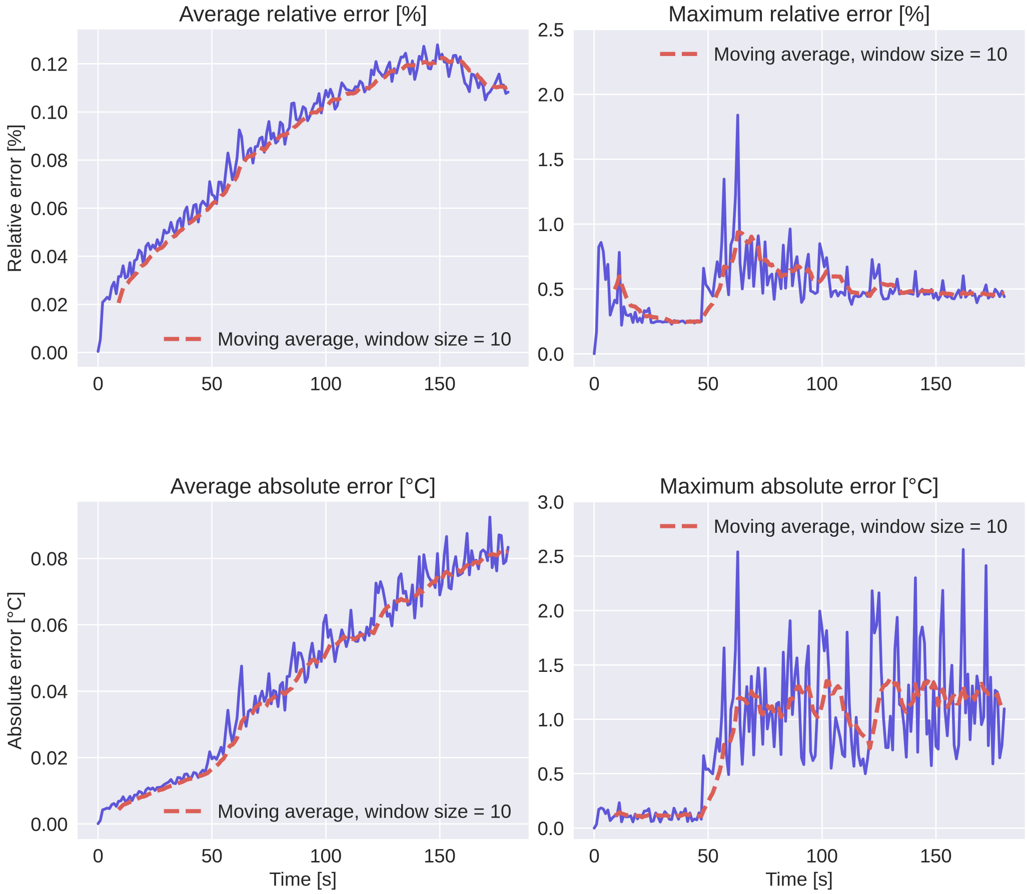

Figure 5: The dependence of relative and absolute errors on time for Case No. 1.

3.1.3. Runtimes

Figure 4 compares the average and maximum time step runtimes with the time step sizes Δt_{rec} used for the solution reconstruction. For the digital twinning application it is essential that the calculation speed is as close as possible to real time; thus, the time step runtime should be equivalent to or less than the time step size. While the runtimes are significantly shorter than the values typically found in literature for the physics-based methods [24, 25], the utilisation of the local GPU is still not quite enough to match the time step size. On the other hand, the supercomputer GPU runtimes are in the range of 1.2-1.8s with Δt_{rec} being equal to 2s (Cases No. 5 and 6); hence, the solution reconstruction on supercomputer happens in near real

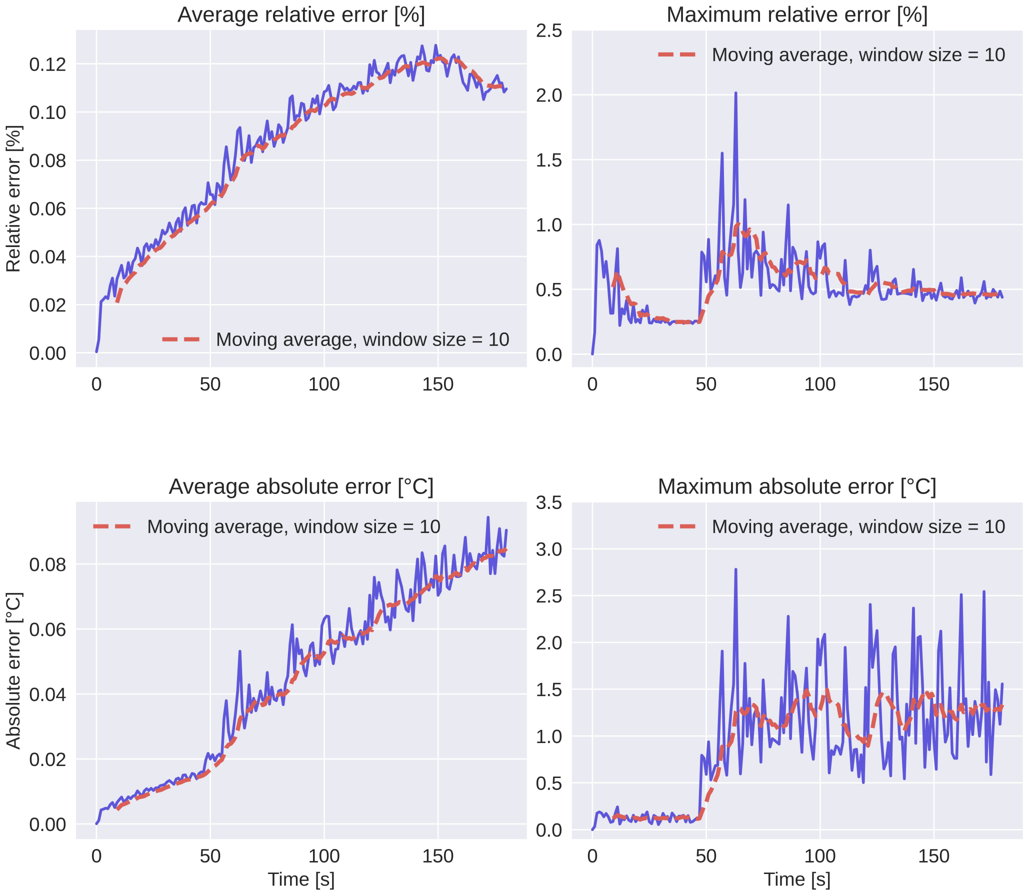

Figure 6: The dependence of relative and absolute errors on time for Case No. 2.

time with 1-2s fixed delay.

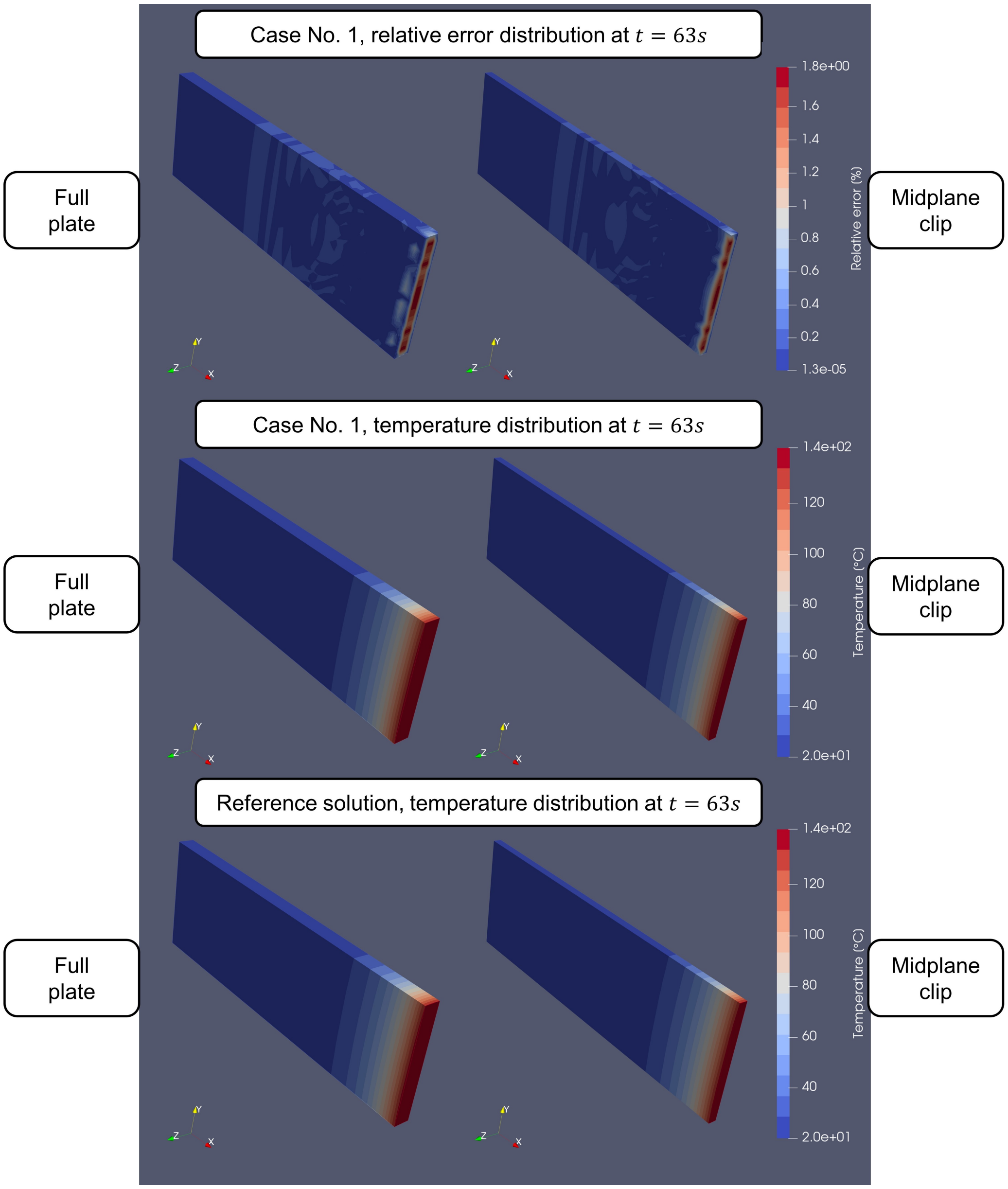

Figure 7: The relative error and temperature distributions for the 63s time instance of Case 1, together with the reference solution temperature distribution; this time instance corresponds to the maximum overall relative error for Case 1.

This changes for nonlinear and more complex problems. For meshes

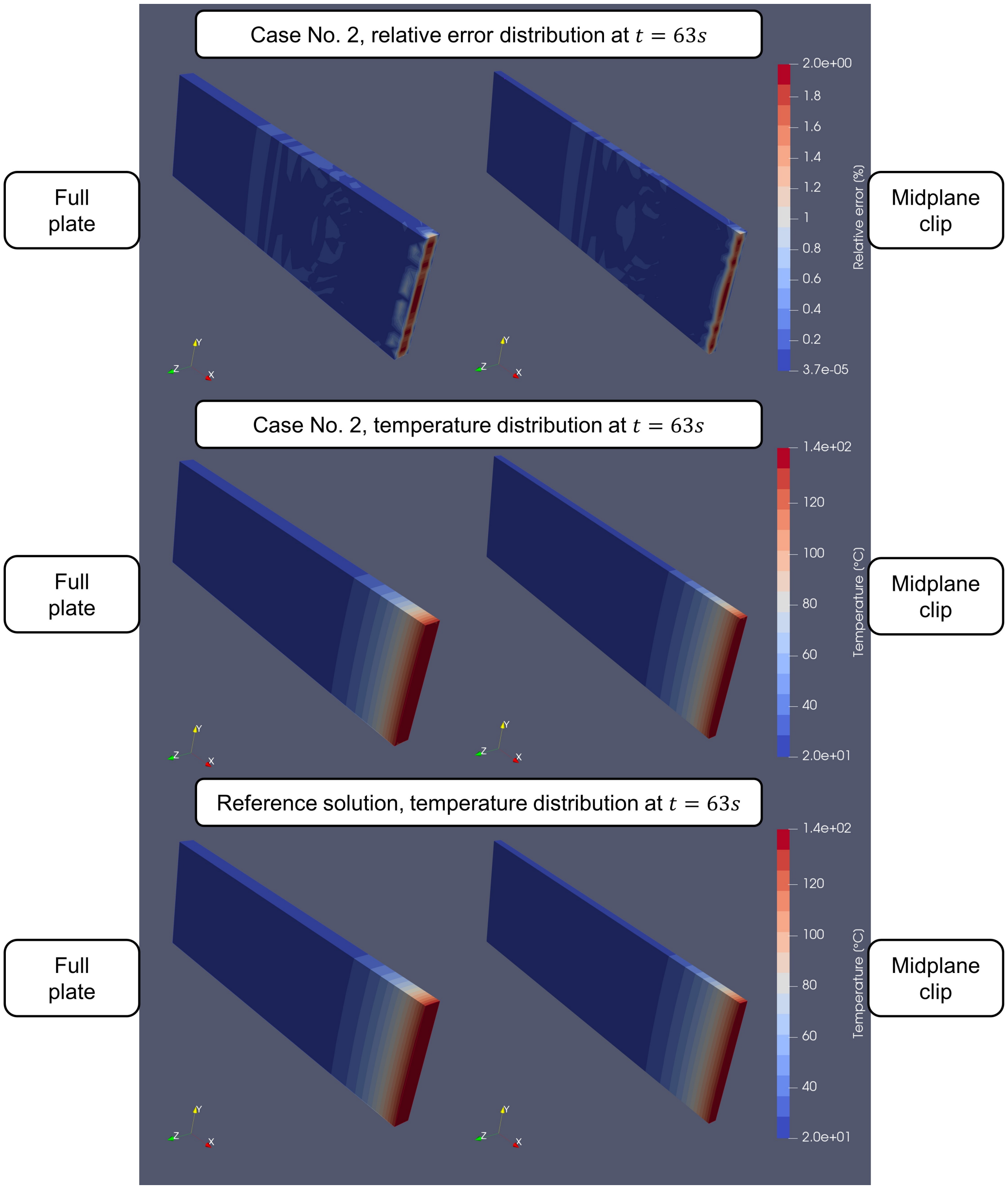

Figure 8: The relative error and temperature distributions for the 63s time instance of Case 2, together with the reference solution temperature distribution; this time instance corresponds to the the maximum overall relative error for Case 2.

with a higher number of nodes several GPUs can be utilised concurrently to sufficiently reduce runtimes, the process which might not be dissimilar to how ML models can be trained on multiple GPUs.

3.2. Multiple choices

In this sub-section it is assumed that the desired time-dependent total heat Q_{goal} is as follows:

$$Q_{\text{goal}}(t) = -0.03t^2 + 1.7t \tag{12}$$

The transient loss function is defined as described in sub-section 2.4.

Table 6 presents the selected parameter values. It is decided to use Δt_{rec} equal to 2s for generating multiple solutions, as this value coupled with the current time step runtimes allows to emulate a near real-time operation with a constant delay as it is shown in sub-section 3.1. Three sets of randomised initial guesses are produced corresponding to the three different generated solutions for each value of c_4, with each set containing the number of initial guess vectors $\{\boldsymbol{T}\}_{\text{init}}$ equal to the number of time steps. The simulation is run for 30s.

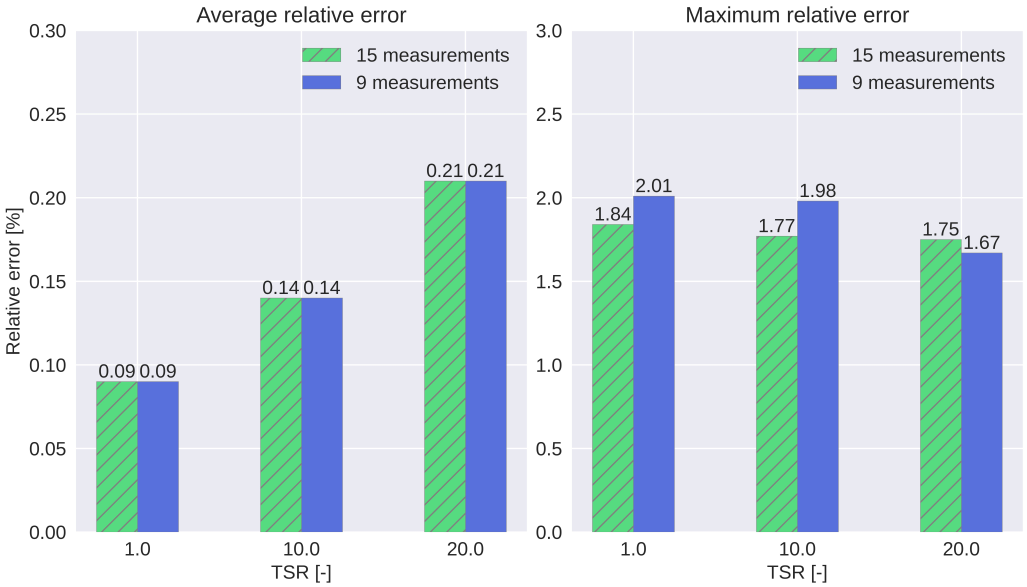

Figure 9: The dependence of the average and maximum relative errors on TSR; 15 measurements are used in Cases 1, 3, and 5 (1, 10, and 20 TSR, respectively), while 9 measurements are used in Cases 2, 4, and 6 (1, 10, and 20 TSR, respectively).

3.2.1. Total heat

Figure 10 shows a total heat variation with time for each of the nine solution options; it is compared with the total heat goal $Q_{\text{goal}}(t)$ represented by Eq. 12. Table 7 summarises the overall total heat relative and absolute errors averaged in time as well as the overall total heat maximum relative and absolute errors for each option. Figure 11 compares the average and maximum total heat relative errors for three values of c_4; whereas, Figure 12 showcases the temperature distributions for nine solution options at 26s time instance.

It can be observed that the average and maximum errors decrease with the c_4 increase, and a close match between the desired total heat temporal variation and the actual total heat variation is easily achieved. Nevertheless, some distinct differences between the temperature distributions are visible in Figure 12. Furthermore, this figure exemplifies what is mentioned in Section 2, viz., the value of c_4 can potentially be used to control the general type or shape of the solution generated. And indeed, for each of the three considered values of c_4 the solutions generated from random initial guess sets display similar features. Additionally, it is worth noting that Solution

Table 6: Parameter values used for generating multiple solution options.

Opt. No.	Δt_{rec} [s] [a]	T_{min} [°C]	T_{max} [°C]	s_{ncg} [c]	s_{gn} [d]	c_1	c_3	c_4	Random set No.
1	2.0	20	100	2	1	1.0	0.1	0.11	1
2	2.0	20	100	2	1	1.0	0.1	0.11	2
3	2.0	20	100	2	1	1.0	0.1	0.11	3
4	2.0	20	100	2	1	1.0	0.1	0.10	1
5	2.0	20	100	2	1	1.0	0.1	0.10	2
6	2.0	20	100	2	1	1.0	0.1	0.10	3
7	2.0	20	100	2	1	1.0	0.1	0.14	1
8	2.0	20	100	2	1	1.0	0.1	0.14	2
9	2.0	20	100	2	1	1.0	0.1	0.14	3

[a] Δt_{rec} is a time step size used for generating a solution option.

[c] s_{ncg} is a number of nonlinear conjugate gradient algorithm iterations (Figure 1).

[d] s_{gn} is a number of Gauss–Newton algorithm iterations (Figure 1).

options No. 4 and 6 as well as Solution options No. 7 and 9 look identical in Figure 12; however, they differ significantly at earlier time instances, whilst adhering to the same solution shape.

Table 7: Average and maximum relative and absolute total heat errors for nine solution options considered for the stainless steel plate.

Opt. No.	Average relative error [%]	Maximum relative error [%]	Average absolute error [W]	Maximum absolute error [W]
1	1.69	9.68	0.15	0.32
2	1.76	9.14	0.16	0.34
3	1.64	8.33	0.15	0.30
4	2.20	10.00	0.20	0.55
5	2.48	12.10	0.22	0.49
6	2.19	12.35	0.19	0.41
7	0.43	2.49	0.04	0.08
8	0.45	2.32	0.04	0.08
9	0.42	2.37	0.04	0.08

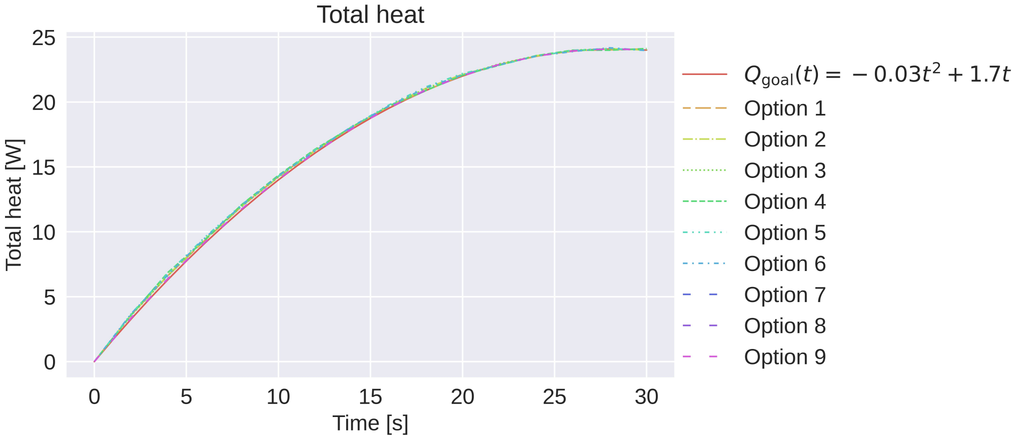

Figure 10: The dependence of the total heat on time for various solution options; the total heat goal Q_{goal} corresponds to Eq. 12.

3.2.2. Runtimes

As is required for real time control of an experiment, the simulation can run close to real time. Figure 13 compares the average and maximum time step runtimes with the time step size Δt_{rec} of 2s used to generate these solutions. As with the solution reconstruction cases in Sub-section 3.1, the simulation on the local GPU is not quite able to operate in near real time; however, the simulation run on the supercomputer GPU attains excellent runtimes of approximately 1.2s per time step when the time step size is 2s.

4. Conclusions

In conclusion, this paper introduces an inverse analysis framework, which relies purely on standard FE discretisations, and highlights its potential by analysing its application to a transient heat conduction in a stainless steel plate, which represents experimental scenario. After the initial FE discretisation the presented workflow deviates significantly from a standard FE workflow used to solve forward problems.

The results in this paper address the problem of inverse solution reconstruction from sparse measurements in near real time, which is directly

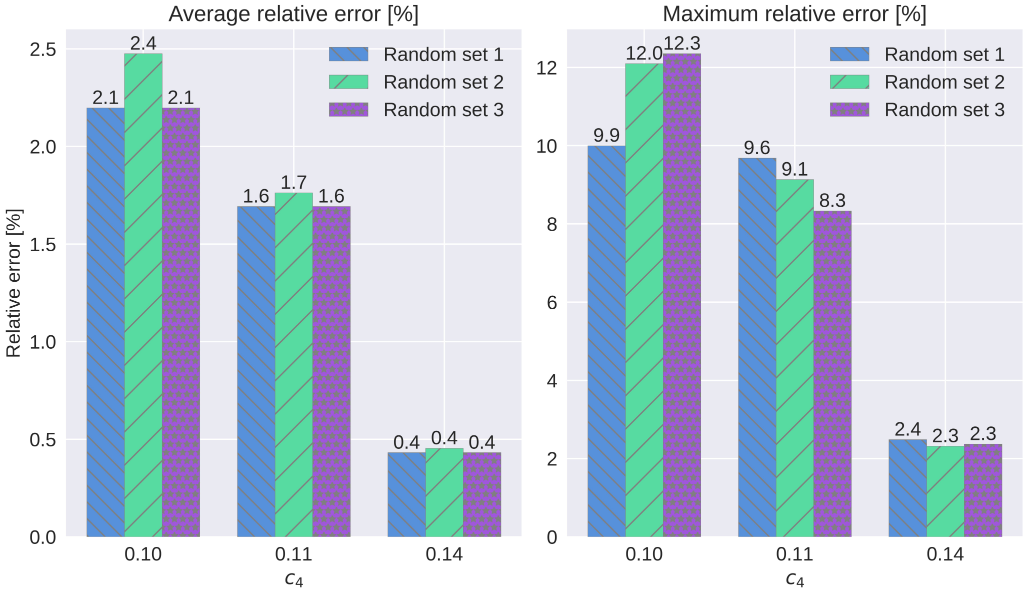

Figure 11: The dependence of the relative total heat errors on c_4.

related to the system monitoring objective (Section 1). Furthermore, they showcase a novel approach for producing multiple options fitting certain requirements in a controlled manner, a functionality essential for the semi-autonomous system control objective (Section 1). This is an innovative way of taking advantage of the intrinsic inverse problem ill-posedness. The demonstrable advantages of the presented methodology include the following:

1. It combines the best characteristics of the existing data-driven and physics-based approaches, with or without ML. No copious amounts of

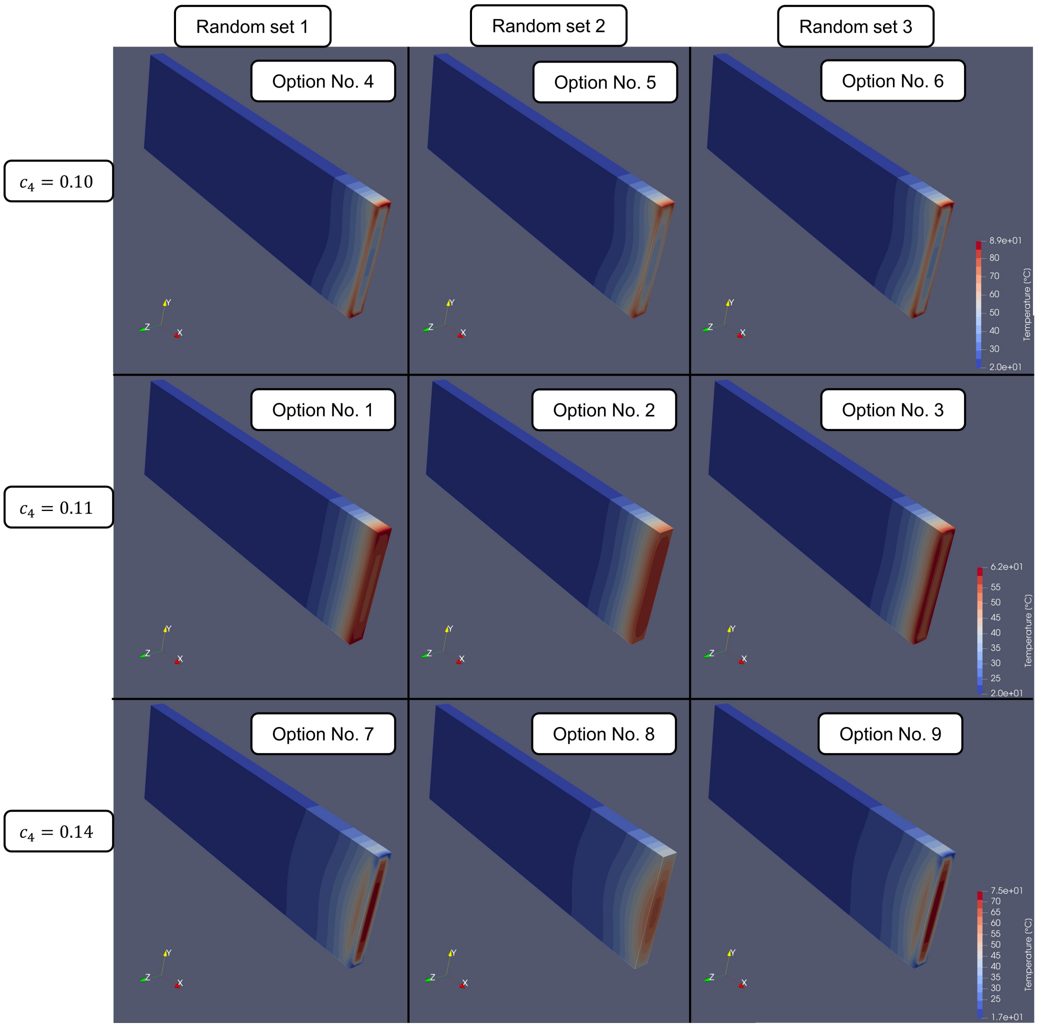

Figure 12: Temperature distributions for nine solution options at 26s time instance.

training data are required to accomplish acceptable accuracy, while the current runtimes allow for the near real-time operation with a consistent 1-2s offset.

2. It only makes use of the domain information that might be available in the real experimental setup.
3. It utilises the traditional industry-standard simulation method for solid materials, which significantly simplifies the process of its integration into the existing simulation workflows in the industry. $[\boldsymbol{C}]$, $[\boldsymbol{K}]$, $\{\boldsymbol{f}\}_h$, and mesh can be generated using any preferred software, while the framework's output can be post-processed and analysed again using any preferred software; and also it can be directly compared with any existing forward simulations. Moreover, this means that all the new FEM-related developments can be potentially employed to enhance its performance; for example, such developments might include novel meshing techniques [40].

While there are still some important considerations to be dealt with, such as temperature-dependency of the material properties and more complex geometries, the current results are encouraging, indicating significant potential for the digital twinning applications.

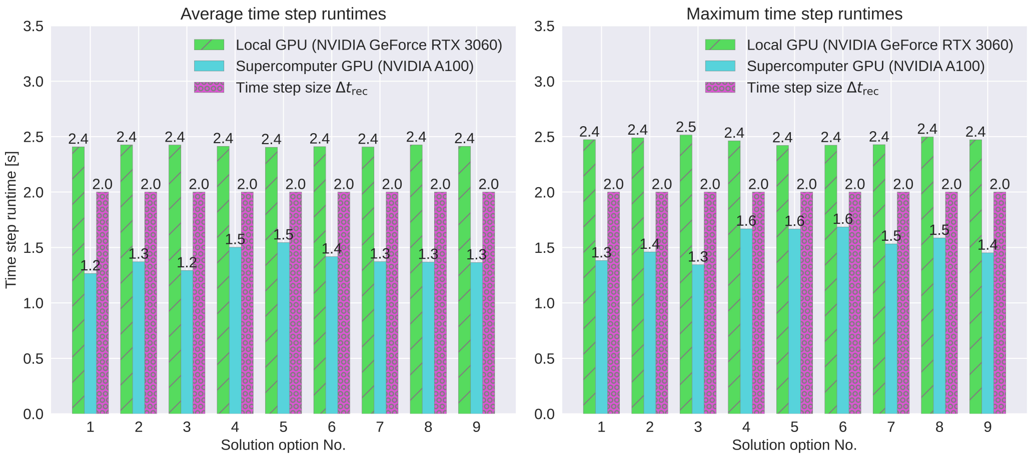

Figure 13: Average and maximum time step runtimes for all generated solution options, compared with the time step size Δt_{rec} used; one GPU is used for the local and supercomputer cases.

5. Acknowledgements

This work has been part-funded by the EPSRC Energy Programme [grant number EP/W006839/1]. The authors acknowledge the support of Supercomputing Wales and AccelerateAI projects, which is part-funded by the European Regional Development Fund (ERDF) via the Welsh Government for giving them access to NVIDIA A100 40GB GPUs. Additionally, the authors would like to thank Lloyd Fletcher and Adel Tayeb from UKAEA for the fruitful discussions regarding the physical testing of the heat exchange components and the diagnostic tools used during the experiments.

6. Declarations

The source codes along with datasets used in the current paper are available from the corresponding author upon request.

References

[1] J.-F. Yao, Y. Yang, X.-C. Wang, X.-P. Zhang, Systematic review of digital twin technology and applications, Visual Computing for Industry, Biomedicine, and Art 6 (2023) 10. doi:10.1186/s42492-023-00137-4.

[2] A. Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, Society for Industrial and Applied Mathematics, 2004.

[3] W. Li, D. Sadigh, S. S. Sastry, S. A. Seshia, Synthesis for human-in-the-loop control systems, in: E. Ábrahám, K. Havelund (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 470–484.

[4] S. Arridge, P. Maass, O. Öktem, C.-B. Schönlieb, Solving inverse problems using data-driven models, Acta Numerica 28 (2019) 1–174. doi:10.1017/S0962492919000059.

[5] D. Lehký, Z. Keršner, D. Novák, Framepid-3pb software for material parameter identification using fracture tests and inverse analysis, Advances in Engineering Software 72 (2014) 147–154. doi:https://doi.org/10.1016/j.advengsoft.2013.10.001, special Issue

dedicated to Professor Zdeněk Bittnar on the occasion of his Seventieth Birthday: Part 2.

[6] M. Paruch, A. Piasecka-Belkhayat, A. Korczak, Identification of the ultra-short laser parameters during irradiation of thin metal films using the interval lattice boltzmann method and evolutionary algorithm, Advances in Engineering Software 180 (2023) 103456. doi:https://doi.org/10.1016/j.advengsoft.2023.103456.

[7] A. Bangian-Tabrizi, Y. Jaluria, An optimization strategy for the inverse solution of a convection heat transfer problem, International Journal of Heat and Mass Transfer 124 (2018) 1147 – 1155. doi:10.1016/j.ijheatmasstransfer.2018.04.053.

[8] H. R. Tamaddon-Jahromi, N. K. Chakshu, I. Sazonov, L. M. Evans, H. Thomas, P. Nithiarasu, Data-driven inverse modelling through neural network (deep learning) and computational heat transfer, Computer Methods in Applied Mechanics and Engineering 369 (2020) 113217. doi:https://doi.org/10.1016/j.cma.2020.113217.

[9] H. Tercan, T. Meisen, Machine learning and deep learning based predictive quality in manufacturing: a systematic review, Journal of Intelligent Manufacturing 33 (2022) 1879–1905. doi:10.1007/s10845-022-01963-8.

[10] H.-W. Chiu, C.-H. Lee, Prediction of machining accuracy and surface quality for cnc machine tools using data driven approach, Advances in Engineering Software 114 (2017) 246–257. doi:https://doi.org/10.1016/j.advengsoft.2017.07.008.

[11] S. Le Clainche, E. Ferrer, S. Gibson, E. Cross, A. Parente, R. Vinuesa, Improving aircraft performance using machine learning: A review, Aerospace Science and Technology 138 (2023) 108354. doi:https://doi.org/10.1016/j.ast.2023.108354.

[12] S. Szrama, T. Lodygowski, Aircraft engine remaining useful life prediction using neural networks and real-life engine operational data, Advances in Engineering Software 192 (2024) 103645. doi:https://doi.org/10.1016/j.advengsoft.2024.103645.

[13] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, 2016.

[14] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, Adaptive Computation and Machine Learning series, MIT Press, 2005.

[15] B. Mehlig, Machine Learning with Neural Networks: An Introduction for Scientists and Engineers, Cambridge University Press, 2021. doi:10.1017/9781108860604.

[16] G. Yagawa, H. Okuda, Neural networks in computational mechanics, Archives of Computational Methods in Engineering 3 (1996) 435 – 512. doi:10.1007/BF02818935.

[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–80. doi:10.1162/neco.1997.9.8.1735.

[18] W. Bielajewa, M. Tindall, P. Nithiarasu, Comparative study of transformer- and lstm-based machine learning methods for transient thermal field reconstruction, Computational Thermal Sciences: An International Journal 16 (2024). doi:10.1615/ComputThermalScien.2023049936.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[20] R. Lewis, Simulation driven machine learning methods to optimise design of physical experiments and enhance data analysis for testing of fusion energy heat exchanger components, Phd thesis, Swansea University, 2023.

[21] Y. Zhang, Z. Gong, W. Zhou, X. Zhao, X. Zheng, W. Yao, Multi-fidelity surrogate modeling for temperature field prediction using deep convolution neural network, Engineering Applications of Artificial Intelligence 123 (2023) 106354. doi:https://doi.org/10.1016/j.engappai.2023.106354.

[22] F. Zhu, J. Chen, D. Ren, Y. Han, Transient temperature fields of the tank vehicle with various parameters using deep

learning method, Applied Thermal Engineering 230 (2023) 120697. doi:https://doi.org/10.1016/j.applthermaleng.2023.120697.

[23] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. doi:https://doi.org/10.1016/j.jcp.2018.10.045.

[24] P. Sharma, L. Evans, M. Tindall, P. Nithiarasu, Stiff-pdes and physics-informed neural networks, Archives of Computational Methods in Engineering 30 (2023) 2929–2958. doi:10.1007/s11831-023-09890-4.

[25] P. Karnakov, S. Litvinov, P. Koumoutsakos, Flow reconstruction by multiresolution optimization of a discrete loss with automatic differentiation, The European Physical Journal E 46 (2023) 59. doi:10.1140/epje/s10189-023-00313-7.

[26] M. Balcerak, I. Ezhov, P. Karnakov, S. Litvinov, P. Koumoutsakos, J. Weidner, R. Z. Zhang, J. S. Lowengrub, B. Wiestler, B. Menze, Individualizing glioma radiotherapy planning by optimization of a data and physics informed discrete loss, 2023. arXiv:2312.05063.

[27] P. Nithiarasu, R. W. Lewis, K. N. Seetharamu, Fundamentals of the Finite Element Method for Heat and Mass Transfer, 2 ed., Wiley, 2016.

[28] ANSYS, Inc., Ansys: Engineering simulation software, Available at https://www.ansys.com/, 1970–2024. [Software].

[29] Électricité de France (EDF), Code_Aster: Open source finite element solver, analysis of structures and thermomechanics for studies and research, Open source on https://code-aster.org/, 1989–2024. [Software].

[30] O. C. Zienkiewicz, R. L. Taylor, The Finite Element Method: Its Basis and Fundamentals, volume 2, 4 ed., McGraw-Hill Book Company (UK) Limited, 1989.

[31] J. Nocedal, S. J. Wright, Numerical Optimization, Springer Series in Operations Research and Financial Engineering, 2 ed., Springer New York, 2006. doi:10.1007/978-0-387-40065-5.

[32] Å. Björck, Numerical Methods for Least Squares Problems, Society for Industrial and Applied Mathematics, 1996. doi:10.1137/1.9781611971484.

[33] S. Gratton, A. S. Lawless, N. K. Nichols, Approximate gauss–newton methods for nonlinear least squares problems, SIAM Journal on Optimization 18 (2007) 106–132. doi:10.1137/050624935.

[34] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: A survey, J. Mach. Learn. Res. 18 (2017) 5595–5637.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance deep learning library, Curran Associates Inc., Red Hook, NY, USA, 2019.

[36] D. J. Wales, J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms, The Journal of Physical Chemistry A 101 (1997) 5111–5116. doi:10.1021/jp970984n.

[37] R. E. Walpole, R. H. Myers, S. L. Myers, K. Ye, Probability & Statistics for Engineers & Scientists: MyStatLab Update, 9 ed., Pearson Boston, Boston, 2017.

[38] D. Hancock, Employing Additive Manufacturing for Fusion High Heat Flux Structures, Phd thesis, University of Sheffield, 2018.

[39] T. Barrett, M. Bamford, B. Chuilon, T. Deighan, P. Efthymiou, L. Fletcher, M. Gorley, T. Grant, T. Hall, D. Horsley, M. Kovari, M. Tindall, The chimera facility development programme, Fusion Engineering and Design 194 (2023) 113689. doi:https://doi.org/10.1016/j.fusengdes.2023.113689.

[40] X. Zou, S. B. Lo, R. Sevilla, O. Hassan, K. Morgan, The generation of 3d surface meshes for nurbs-enhanced fem, Computer-Aided Design 168 (2024) 103653. doi:https://doi.org/10.1016/j.cad.2023.103653.

Appendix A. 1D steady-state example

This appendix briefly analyses a 1D steady-state example in order to more clearly convey the idea behind the residual loss function term introduced in Section 2. Figure A.14 shows a discretised domain comprised of three nodes and two elements; heat flux q is applied to Node 1, while the domain-atmosphere convection is assumed to take place at Node 3. The specific numbers are not given here as they are not necessary for the understanding of the general concept.

The three equations corresponding to each node are written as follows:

$$\begin{aligned}
\text{Node 1:}\quad & [\boldsymbol{K}]_{11}\, T_1 + [\boldsymbol{K}]_{12}\, T_2 + [\boldsymbol{K}]_{13}\, T_3 = \{\boldsymbol{f}\}_{q1} + \{\boldsymbol{f}\}_{h1} \\
\text{Node 2:}\quad & [\boldsymbol{K}]_{21}\, T_1 + [\boldsymbol{K}]_{22}\, T_2 + [\boldsymbol{K}]_{23}\, T_3 = \{\boldsymbol{f}\}_{q2} + \{\boldsymbol{f}\}_{h2} \\
\text{Node 3:}\quad & [\boldsymbol{K}]_{31}\, T_1 + [\boldsymbol{K}]_{32}\, T_2 + [\boldsymbol{K}]_{33}\, T_3 = \{\boldsymbol{f}\}_{q3} + \{\boldsymbol{f}\}_{h3}
\end{aligned} \tag{A.1}$$

Due to the mesh and hence $[\boldsymbol{K}]$ connectivity and known locations of applied BCs, Eq. A.1 can be reduced to the following:

$$\begin{aligned}
\text{Node 1:}\quad & [\boldsymbol{K}]_{11}\, T_1 + [\boldsymbol{K}]_{12}\, T_2 = \{\boldsymbol{f}\}_{q1} \\
\text{Node 2:}\quad & [\boldsymbol{K}]_{21}\, T_1 + [\boldsymbol{K}]_{22}\, T_2 + [\boldsymbol{K}]_{23}\, T_3 = 0 \\
\text{Node 3:}\quad & [\boldsymbol{K}]_{32}\, T_2 + [\boldsymbol{K}]_{33}\, T_3 = \{\boldsymbol{f}\}_{h3}
\end{aligned} \tag{A.2}$$

Finally, the residual term for this problem can be defined as:

$$\begin{aligned}
\text{Residual term} = c_1 \left([\boldsymbol{K}]_{21}\, T_1 + [\boldsymbol{K}]_{22}\, T_2 + [\boldsymbol{K}]_{23}\, T_3\right)^2 + \\
+ c_1 \left([\boldsymbol{K}]_{32}\, T_2 + [\boldsymbol{K}]_{33}\, T_3 - \{\boldsymbol{f}\}_{h3}\right)^2
\end{aligned} \tag{A.3}$$

In Eq. A.3, it can be seen that the equation associated with Node 1 where the unknown heat flux is applied is excluded. Nevertheless, the $[\boldsymbol{K}]$ matrix connectivity and thus the nodal heat flux balance are preserved, since T_1 is still present in the calculation. Node 1 equation is excluded from Eq. A.3 as $\{\boldsymbol{f}\}_{q1}$ is assumed to be unknown; consequently, $\{\boldsymbol{f}\}_{q1}$ becomes an implicit

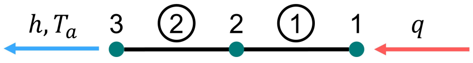

Figure A.14: The general 1D example with three nodes and two elements.

DoF, which are not directly adjusted during the optimisation procedure, together with the explicit DoF $\{\boldsymbol{T}\}$, which are directly adjusted during the optimisation procedure. However, undoubtedly, as DoF, $\{\boldsymbol{f}\}_{q1}$ and $\{\boldsymbol{T}\}$ are strongly interdependent.

Appendix B. Weighting coefficient selection for solution reconstruction

The weighting coefficient selection for solution reconstruction is mentioned in Section 3; this appendix provides a more detailed view of this process. Figures B.15 and B.16 show the average and maximum relative and absolute solution reconstruction errors, in space and time, for various values of c_3 for all solution reconstruction cases considered in this paper (Table 4). The cases are split into two groups: 15 measurements (Cases No. 1, 3, and 5 shown in Figure B.15), and 9 measurements (Cases No. 2, 4, and 6 shown in Figure B.16); nevertheless, a number of common features can be discerned:

1. The solutions converge for all values of c_3 belonging to $(0, 1]$, and the converged temperature distribution is generally reasonably close to the reference temperature distribution. The solutions diverge rapidly from the reference solution to infinity or to some finite temperature distribution vastly different from the reference temperature distribution for c_3 equal to zero, i.e. when no regularisation is applied, thus highlighting the importance of the regularisation term. For every set of measurements there usually exists a temperature distribution perfectly fitting the residual and measurement terms but having an extremely abrupt unphysical temperature variations on the surface; and in the absence of any limiting factor, such as the regularisation term, the optimisation process tends to converge to this unrealistic temperature distribution or not converge at all.
2. The shapes of the curves are similar between Cases No. 1 and 3, and likewise between Cases No. 2 and 4; the only significant difference is that the average curves are displaced in the direction of y axis. All these cases share the same value of Δt_{rec} that is equal to 1s. However, the values of TSR (Eq. 11) are different (1.0 for Cases No. 1 and 2, and 10.0 for Cases No. 3 and 4), which might explain the displacement of the average curves in y axis direction.

3. The shape of the curve for Case No. 5 noticeably differs from the aforementioned curves for Cases No. 1 and 3; the same phenomenon can be observed for Case No. 6 as compared with Cases No. 2 and 4. Examining the various parameter values in Table 4 indicates that this change might be caused by the transition of Δt_{rec} from 1s to 2s.

For Cases No. 1-5 the locations of the minima of average relative and maximum relative errors coincide; consequently, the value of 1.0 is selected (Table 4). The situation is is a bit more complex for Case No. 6, since the minima of average relative and maximum relative errors are reached at different values of c_3, albeit very close ones. The value of 0.5 is selected for this case as it corresponds to the minimum of the maximum relative error (Table 4).

Overall, the error dependency on c_3 seems to be defined by the number of measurements as well as Δt_{rec}; consequently, this parameter can be calibrated prior to the experiment. However, it should be noted that usually there are expansive intervals of c_3 where the relative errors vary only slightly, for example values of c_3 equal or above 0.5.

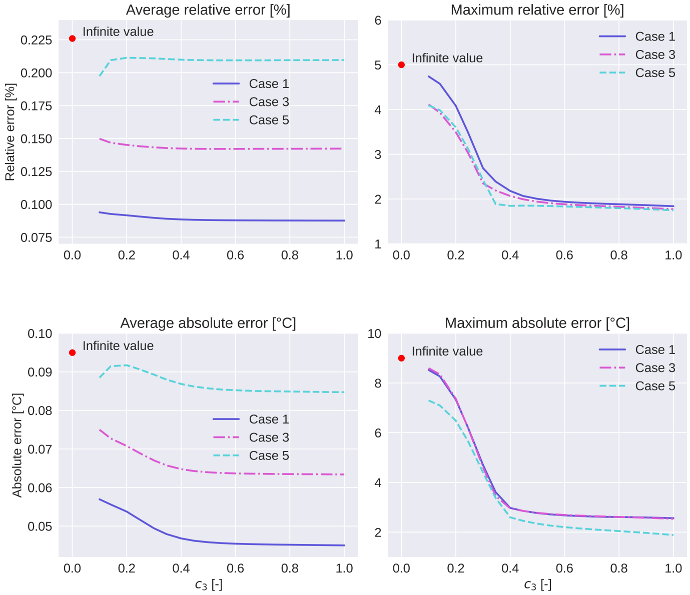

Figure B.15: Average and maximum relative and absolute solution reconstruction errors (in space and time) for various values of c_3 for Cases 1, 3, and 5 (Table 4).

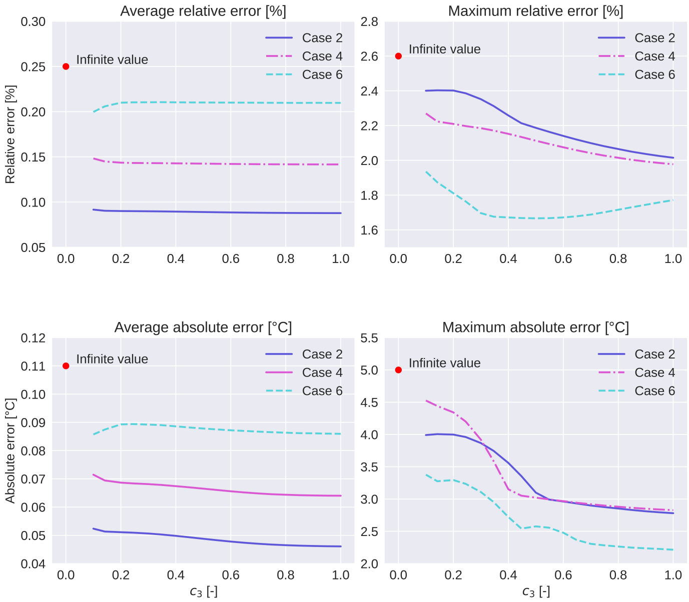

Figure B.16: Average and maximum relative and absolute solution reconstruction errors (in space and time) for various values of c_3 for Cases 2, 4, and 6 (Table 4).

Appendix C. Relative and absolute errors for Cases No. 3-6

The figures contained in this appendix, Figures C.17 to C.20, elaborate on how relative and absolute errors progress over time for Cases No. 3-6.

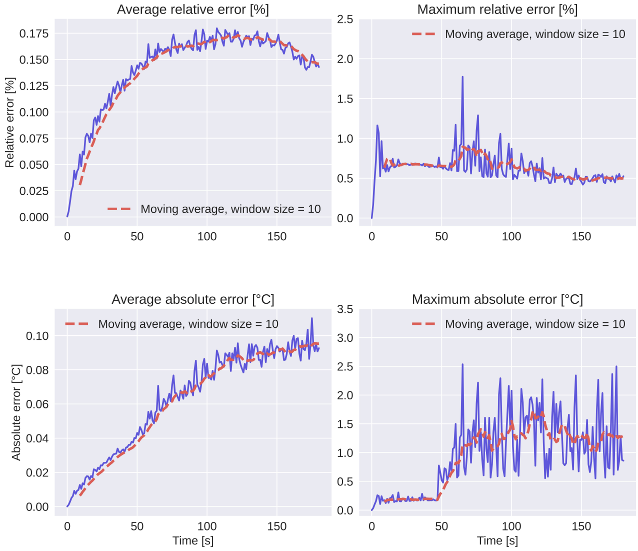

Figure C.17: The dependence of relative and absolute errors on time for Case No. 3.

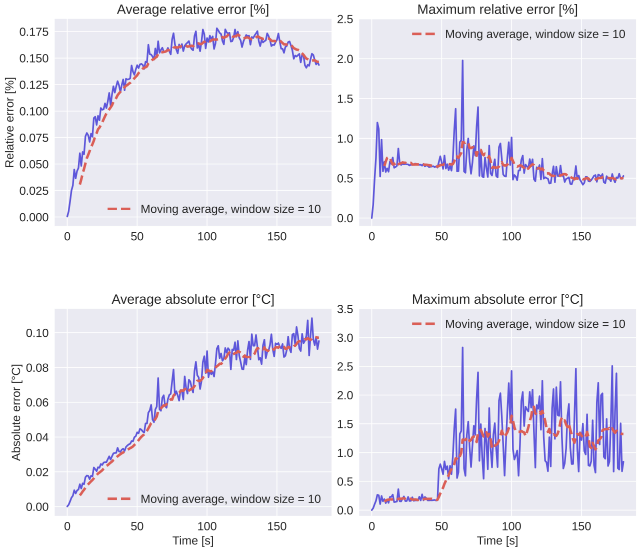

Figure C.18: The dependence of relative and absolute errors on time for Case No. 4.

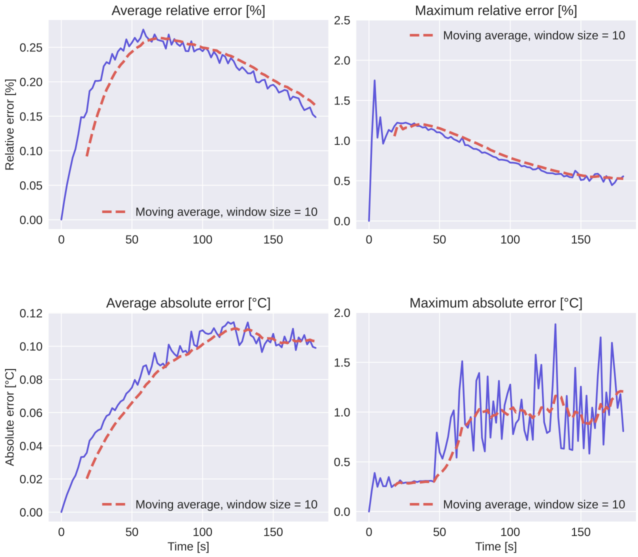

Figure C.19: The dependence of relative and absolute errors on time for Case No. 5.

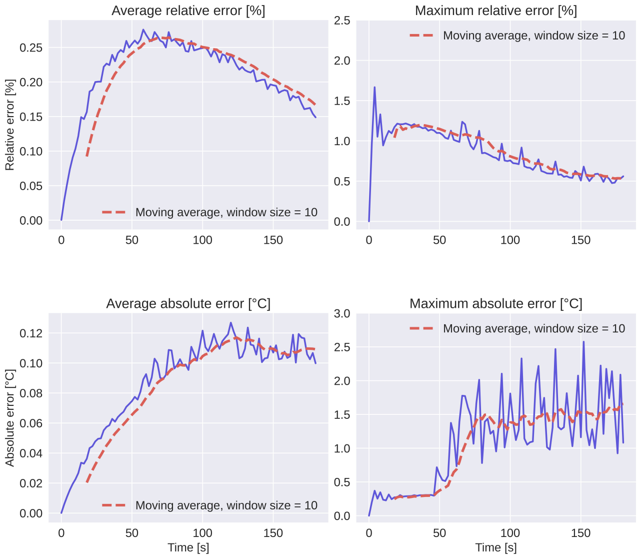

Figure C.20: The dependence of relative and absolute errors on time for Case No. 6.