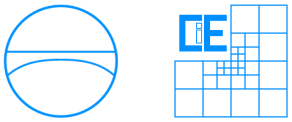


Faculty of Civil Engineering and Geodesy
Chair for Computation in Engineering
Prof. Dr. rer. nat. Ernst Rank

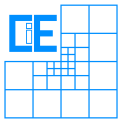
Partitioned Hierarchical Space-time FEM for Transient Heat Problems

Hayden Liu Weng

Master's thesis
for the Master of Science program Computational Mechanics

Author:	Hayden Liu Weng
Matriculation number:	03685577
Supervisor:	Prof. Dr.rer.nat. Ernst Rank
	Philipp Kopp, M.Sc.
	Nina Korshunova, M.Sc.
Date of issue:	01. October 2018
Date of submission:	31. March 2019

Involved Organisations



Chair for Computation in Engineering
Faculty of Civil Engineering and Geodesy
Technische Universität München
Arcisstraße 21
D-80333 München

Declaration

With this statement I declare, that I have independently completed this Master's thesis. The thoughts taken directly or indirectly from external sources are properly marked as such. This thesis was not previously submitted to another academic institution and has also not yet been published.

München, April 9, 2019

Hayden Liu Weng

Hayden Liu Weng
e-Mail: h.liu@tum.de

Acknowledgments

I would like to show my greatest appreciation to all of the people who made the completion of my thesis possible.

First of all, I would particularly like to thank Nina Korshunova for her continuous support and commitment throughout the realization of this thesis, devoting incredible amounts of time and energy to it. Her positive attitude and permanent guidance helped me to organize my thoughts, and her invaluable advice helped me stay on track with the the completion of this work. Moreover, I want to thank her for the great work environment she created for me, which made my work more enjoyable.

I would also like to express my gratitude to Philipp Kopp for the insights he gave me at all stages of this work, providing not only a starting point for my implementation, but also pointing me towards key ideas to many of the hurdles which I faced over the course of this thesis. Without his guidance and persistent help this thesis would not have been possible.

I am indebted to Prof. Ernst Rank, who gave me the opportunity to work on this topic, and who supported me through all of my thesis, posing interesting questions and ideas to have a better grasp of the bigger picture. I also want to thank him for supervising and examining my work.

I am deeply grateful to Dr.-Ing. Stefan Kollmannsberger and the entire Chair for Computation in Engineering research group. I have greatly benefited from their expertise and constructive review of my thesis, as well as for their words of encouragement.

Special thanks go to my Computational Mechanics colleagues, who have made this experience less isolating, sharing our individual highs and lows throughout the last few years, and specially the last few months. Pushing each other towards the completion of our individual projects was also a very gratifying experience in itself.

I would also like to thank my friends who are elsewhere in the world, for being there for me during every step of my life. They provided me both a chance to clear up my head for a bit, as well as words of encouragement when it was time to keep going. I am grateful for their patience and for keeping me motivated while writing this thesis.

Last but not least, I want to express my gratitude to my family. Without their unconditional support, I would not be anywhere close to where I am today. All of this has only been possible because of them.

Contents

Chapter 1

Introduction and Motivation

1.1 Motivation and objectives

Over the last few years, techniques such as additive manufacturing (AM) have been gaining popularity and rapidly changing the industrial world. The method provides the ability to rapidly generate three-dimensional objects with arbitrarily complex shapes, something not previously possible through traditional manufacturing techniques. In addition to this, the method provides a high geometrical accuracy without the need of bulky or expensive molds, facilitating the production and improving the functionality and mechanical properties of multiple key elements across different industries [11]. Among others, the aerospace, automotive, biomedicine, and electronics industries have all started to adopt the process for its flexibility not just for fast prototyping at reduced costs, but also for the production of the final parts themselves.

Among the additive manufacturing processes, selective laser melting (SLM) is one of the most prominent in industrial settings. In this process, CAD data from the target parts is transformed into two-dimensional layer information which have to be added onto the final object. These layers are formed employing a high energy density laser beam which fully melts the material powder. Therefore, dense parts with properties close to the bulk material can be created this way [21]. However, the high energy densities from the SLM process also has negative side effects: the final product might still present a certain degree of pore defects, material spattering, or residual powder. Moreover, non-connected layers might occur due to denudation [17]. In contrast, the lower heat densities of the analogous selective laser sintering process (SLS), where fusion occurs only at the molecular level, provides full control over the porosity of the product [22].

Since the quality of the produced object is dependent on the material, the size, and the geometry of the target specimen, it is fundamental to be able to estimate the process variables before entering the formal production stage of a part. Currently, this is usually carried out by trial and error, which ends up being a fairly expensive procedure until satisfactory results are obtained. Instead of this, it would be of interest to use numerical simulations to quantify the involved variables and better control the production output of the additive manufacturing processes.

The current simulations of the SLM processes performed via the Finite Element Method

(FEM) are very computationally expensive and complex, a result of the physical phenomena occurring during the additive manufacturing process. First of all, the phase of the simulated material differs: the individual powder particles in the region under the laser beam are melted, whereas the material from the previous layers is already solidified and cooling down. Secondly, the high rates of heat application and cooling have an effect on the stress state of the object, which, in turn, leads to residual stresses and deformation over the entirety of the specimen, generating geometrical inaccuracies [25]. Indeed, the transient temperature field is intimately related to specimen residual stresses, resulting microstructure, fatigue life, and shape distortions, as well as the melt pool characteristics [32].

Capturing these thermal effects numerically requires the use of *multi-phase* models across *multiple scales*. Moreover, the high dependency on temperature of the physical properties leads to *material nonlinearies* in the simulated problem. This is further emphasized by the very local nature in space (the laser size is various orders of magnitude smaller than the overall specimen) and in time (the laser beam's traveling speed must be high enough to cover a new layer before the previous one cools excessively) of the laser trajectory, which generates high temperature gradients which must be able to be resolved.

The two main research focuses to reduce this computational effort are the development of new physical models which can better describe the physical behavior intrinsically, and the improvement of existing numerical approaches used to compute the results. Within the first area, the usage of two-dimensional models has been the most popular, but it does not account for non-connected layers and is not accurate for more complex geometries [6]. Alternatively, Eulerian approaches have also been developed to reduce the computational effort, but yield lower precision results [37]. In addition to these, studies substituting traditional time-stepping schemes in favor of a space-time approach in the analysis of the heat equation in [15, 1] suggest that a similar strategy might be applied in the additive manufacturing scenario, which is one of the ideas the proposed method employs.

On the other hand, research in the second area has focused in adaptive meshing algorithms. In these approaches, error estimators provide information on the current region of interest to be refined, while simultaneously coarsening the rest of the domain. Studies in these methods have been performed in [3] for shock analysis through finite differences, and in [18] for the multiscale $hp-d$ method introduced by Rank in [27]. The multi-level hp–finite element method introduced in [35], which follows a similar concept as the multiscale $hp-d$, will be considered for the current work.

The present thesis aims to develop a superposition-based space-time method which addresses the difficulties of additive manufacturing process simulations while reducing the overall computational effort.

The Petrov-Galerkin space-time approach will be combined with the multi-level hp–FEM and resolved in a partitioned manner in the present work. The concept of this method is to apply the multi-level hp–approach, which decomposes the full space-time domain into different levels representing the local and global scales, and then solve them separately whilst accounting for the coupling between the different levels. The space-time approach has previously been used in [13] for the solution of problems in elastodynamics, in [23] for the Navier-Stokes equations, as well as in [34, 1] for heat problems, among others. Similarly, the multi-level hp–approach has been used for both linear and non-linear problems in conjunction with varying physical models [16, 36].

As previous work with a partitioned solution of an approach with refinement by superposition by Korshunova [18] was performed with the $hp-d$ method in one-dimension and linear elements, the aim of this work is to extend this concept to higher orders and higher dimensions, whilst applying the multi-level hp–method and a space-time discretization. This work will verify the applicability of the different techniques when used in tandem. The resulting method should provide a less computationally expensive approximation of the additive manufacturing process due to the local refinement both in space and time, allowing the separate solution of local effects and the larger scales.

1.2 Outline

A general outline of the present thesis is now provided to briefly describe the contents of the upcoming chapters.

In the first place, the mathematical and numerical fundamentals corresponding to the nonlinear heat transfer problem are introduced in Chapter 2. This includes the strong formulation of the governing equations, and its transition into the weak form, lowering the continuity requirements for its solution. Having introduced the mathematical model to be used, Section 2.2 presents the Finite Element Method and its nuances, such as the basis functions employed, the application of boundary (and initial) conditions, and the relevance of the L^2 projection to it. Furthermore, ways to deal with the nonlinear aspects of the heat transfer problem, as well as its transient nature, are summarized in Sections 2.3 and 2.4 respectively.

Secondly, given that specific regions of the domain will be of special interest, it is necessary to consider refinement of the simulation space. Section 3.1 deals with the refinement of the finite element mesh, where both the traditional replacement (Subsection 3.1.1) and the superposition (Subsection 3.1.2) refinements are considered. However, as the treatment of the time variable differs according to the method applied (time-stepping and space-time formulations), special considerations regarding its refinement are presented separately in Section 3.2.

Following this, Chapter 4 deals with the implementation of the partitioned multi-level hp–method applying both, the time-stepping, and space-time versions. This includes linear stationary, linear transient, nonlinear stationary, and nonlinear transient formulations resolved traditionally, followed by the linear and nonlinear versions of the Space-Time formulation. Each individual section includes an overview of the formulation's development first, before proceeding to address additional considerations regarding the implementation specific to each of the considered cases.

The proposed formulations are then validated against problems with analytic solutions and are also verified against the AdhoC++ in-house FEM code implementation of the multi-level method in Chapter 5. The chapter follows the case-by-case structure of the previous, providing concrete examples, convergence studies, and comments on the computational costs. For the Space-time formulations, an additional comparison with the corresponding time-stepping transient versions (linear and nonlinear, respectively) are also included.

Finally, the project conclusion and outlook are presented alongside a summary in Chapter 6.

Chapter 2

The Finite Element method for heat transfer problems

The present chapter presents the fundamentals of the Finite Element Method as well as some additional considerations relevant to its application. The mathematical model of the problem, in both its strong and weak forms, is considered in Section 2.1 before proceeding onto its discretization and the numerics involved in its solution. As this work is limited to heat transfer problems, the formulations of the FEM will be shown only for the considered case. Additionally, requirements for existence and uniqueness of the solution to the posed problem, as well as conditions for the convergence to said solution are excluded, since they are beyond the scope of the current work. They can be, however, found in [33], [12], or [38], and are also developed in [36] among others.

2.1 Mathematical model for heat transfer problems

First of all, it is necessary to define the physics involved in the simulated problem. In the context of additive manufacturing, this corresponds to the heat transfer equation, where the unknown primary variable is the scalar temperature field T. This equation describes the conservation of energy throughout a defined domain Ω, meaning that the changes in its value must be balanced with its flux over the domain boundary $\Gamma = \delta\Omega$ and existing sources/sinks. This can be represented by the following boundary value problem (BVP):

$$\begin{cases} \rho(\mathbf{x},T)c(\mathbf{x},T)\dfrac{\partial T}{\partial t} - \nabla\cdot\left(k(\mathbf{x},T)\nabla T\right) = f\left(\mathbf{x}\right) \quad \text{in } \Omega, & \text{(2.1a)} \\ T = T_0 \quad \text{in } \Gamma_D, & \text{(2.1b)} \\ -k_n \left.\dfrac{\partial T}{\partial n}\right|_{\Gamma_N} = q_0 \quad \text{in } \Gamma_N, & \text{(2.1c)} \end{cases}$$

where ρ corresponds to the material density, c to its heat capacity, and k to the thermal conductivity, whereas t defines the time variable, ∇ the gradient operator, and f the heat source of the problem. For convenience, since parameters $\rho(\mathbf{x},T)$ and $c(\mathbf{x},T)$ are always evaluated together at once, the arguments are dropped from the former to yield $\rho c(\mathbf{x},T) :=$

$\rho(\mathbf{x}, T)c(\mathbf{x}, T)$ for a simplified notation.

In this case, the partial differential equation (PDE) from Equation 2.1a is accompanied by the boundary terms from Equation 2.1b and 2.1c, where Γ is split into Γ_D and Γ_N (with $\Gamma = \Gamma_D \dot{\cup} \Gamma_N$) to represent the Dirichlet and Neumann boundaries respectively (see Figure 2.1). Here T_0 represents the prescribed temperature, k_n the thermal conductivity in the normal direction to the boundary Γ_N, and q_0 the prescribed flux.

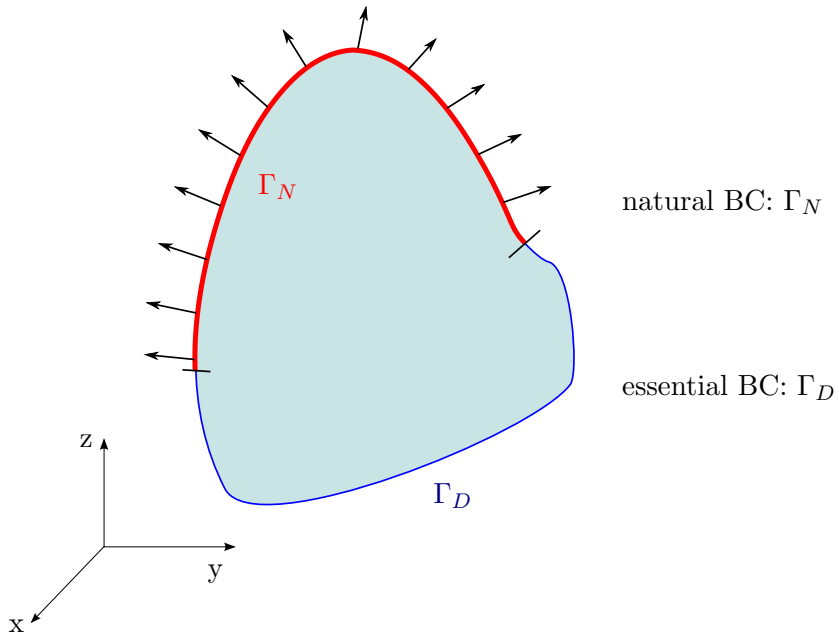


Figure 2.1: Boundary conditions applicable, defined on domain Ω

Moreover, these boundary conditions suffice to model the typical interactions in heat transfer problems: conduction, convection, and radiation. While the Dirichlet boundary condition handles direct conduction with an object at prescribed temperature T_0, the Neumann condition can be used to represent both convection and radiation boundary conditions depending on the value given to q_0 [2]. For instance, convection would correspond to the following heat flux:

$$q_0 = h\left(T_\infty - T|_{\Gamma_N}\right), \tag{2.2}$$

with h being the convection coefficient, T_∞ the environmental temperature, and $T|_{\Gamma_N}$ the temperature at the boundary. Analogously, radiation would be represented by:

$$q_0 = \kappa\left(T_s - T|_{\Gamma_N}\right), \tag{2.3}$$

where κ corresponds to the radiation coefficient and T_s is the radiating source temperature.

In addition to boundary conditions, initial conditions are also necessary whenever time-dependency is involved, providing a starting state from which the remainder of the numerical analysis should proceed. This would correspond to

$$T|_{t=t_0} = T_{\text{init}} \quad \text{at } t = t_0. \tag{2.4}$$

Setting together Equations 2.1a to 2.1c and Equation 2.4 leads to the transient heat transfer initial boundary value problem (IBVP). In should be noted that, in this case, $f, T_o,$ and q_0 would now also be time dependent.

The weak form of the heat equation

Through variational calculus, it is possible to reduce the continuity requirements for the IBVP defined previously. This is done by considering first the trial solution and weighting (or test) function spaces ($\mathbf{X}$ and $\mathbf{V}$ respectively) as subspaces of the Hilbert space $H^1(\Omega)$. Functions in $\mathbf{X}$ are required to follow the boundary conditions, whereas the weighting, or virtual temperature, functions in $\mathbf{V}$ are zero at the Dirichlet boundary [12], namely:

$$\mathbf{X} = \left\{ T |\, T \in H^1(\Omega),\ T = T_0\ \forall \mathbf{x} \in \Gamma_D \right\} \tag{2.5a}$$

$$\mathbf{V} = \left\{ v |\, v \in H^1(\Omega),\ v = 0\ \forall \mathbf{x} \in \Gamma_D \right\}. \tag{2.5b}$$

Multiplying Equation 2.1a by a test function, and integrating over the domain leads to:

$$\int_\Omega \rho c\,(\mathbf{x}, T)\,\frac{\partial T}{\partial t} v\ d\Omega - \int_\Omega \nabla \cdot (k\,(\mathbf{x}, T)\,\nabla T)\,v\ d\Omega = \int_\Omega f\,(\mathbf{x})\,v\ d\Omega \quad \forall v \in \mathbf{V}. \tag{2.6}$$

Integrating by parts, this is:

$$\int_\Omega \rho c\,(\mathbf{x}, T)\,\frac{\partial T}{\partial t} v\ d\Omega + \int_\Omega \nabla v\ k\,(\mathbf{x}, T)\,\nabla T\ d\Omega = \int_\Omega f\,(\mathbf{x})\,v\ d\Omega + \int_{\Gamma_N} k\,(\mathbf{x}, T)\,\nabla T\ v\ d\Gamma, \tag{2.7}$$

where the additional term corresponds to the boundary conditions. Since only Dirichlet and homogeneous Neumann conditions, under which the last term cancels out, are considered, it will be omitted in the remainder of this work.

In addition, Equation 2.1a can also be rewritten into the form:

$$\text{Find } T \in \mathbf{X} \text{ s. t. } \left\langle \rho c\,(\mathbf{x}, T)\,\frac{\partial T}{\partial t}, v \right\rangle - \mathscr{A}\,(T, k(\mathbf{x}, T), v) = \mathscr{F}\,(v) \quad \forall v \in \mathbf{V}. \tag{2.8}$$

where $\langle \cdot, \cdot \rangle$ indicates the inner product, $\mathscr{A}\,(T, k(\mathbf{x}, T), v)$ defines a nonlinear form depending on $k(\mathbf{x}, T)$, the thermal conductivity, which is in turn dependent on the temperature T (leading to nonlinear material coefficients), and $\mathscr{F}\,(v)$ is a linear functional [8].

2.2 Finite element discretization

Now that the weak form of the heat equation is available, the finite element method aims to provide a numerical approximation to the continuous solution. To this effect, finite dimensional subspaces of the previously defined trial solution and weighting function spaces $\mathbf{X}^h \subseteq \mathbf{X}$ and $\mathbf{V}^h \subseteq \mathbf{V}$ are considered, such that the problem can be restated as follows:

$$\text{Find } T^h \in \mathbf{X^h} \text{ s. t. } \left\langle \rho c\left(\mathbf{x}, T^h\right) \frac{\partial T^h}{\partial t}, v^h \right\rangle - \mathscr{A}\left(T^h, k(\mathbf{x}, T^h), v^h\right) = \mathscr{F}\left(v^h\right) \quad \forall v^h \in \mathbf{V}^h. \tag{2.9}$$

Here, the trial solution and the test function are approximated by their finite dimensional counterparts. These can be expressed in terms of a basis of the corresponding subspaces such that:

$$T \approx T^h = \sum_{i=1}^{n_{\text{DoF}}} N_i^T \hat{T}_i, \tag{2.10}$$

$$v \approx v^h = \sum_{j=1}^{n_{\text{DoF}}} N_j^v \hat{v}_j. \tag{2.11}$$

Under the Bubnov-Galerkin method, the sets of basis functions can be considered to be the same, such that $N_k^T = N_k^v =: N_k$, allowing the subspace superscripts to be dropped [38]. Substituting these approximations back into Equation 2.7, the problem can now be rewritten into:

$$\begin{aligned} \sum_{j=1}^{n_{\text{DoF}}} \hat{v}_j \sum_{i=1}^{n_{\text{DoF}}} \dot{\hat{T}}_i \int_\Omega N_j \, \rho c \left(\mathbf{x}, \sum_{k=1}^{n_{\text{DoF}}} N_k \hat{T}_k\right) N_i \, d\Omega \\ + \sum_{j=1}^{n_{\text{DoF}}} \hat{v}_j \sum_{i=1}^{n_{\text{DoF}}} \hat{T}_i \int_\Omega \nabla N_j \, k \left(\mathbf{x}, \sum_{k=1}^{n_{\text{DoF}}} N_k \hat{T}_k\right) \nabla N_i \, d\Omega \\ = \sum_{j=1}^{n_{\text{DoF}}} \hat{v}_j \int_\Omega f\left(\mathbf{x}\right) N_j \, d\Omega, \end{aligned} \tag{2.12}$$

where $\dot{T} = \frac{\partial T}{\partial t}$. Since Equation 2.12 is valid for arbitrary values $\hat{v}_j$, considering in particular $\hat{v} = [1, 0, \ldots, 0], [0, 1, \ldots, 0], \ldots, [0, 0, \ldots, 1]$ (i.e. taking $v^h = N_1, N_2, \ldots, N_{n_{\text{DoF}}}$), coefficient $\hat{v}_j$ can be taken out of the terms and the equation at hand can be reformulated as the governing equation:

$$\mathbf{M}(T)\dot{\hat{T}} + \mathbf{K}(T)\hat{T} = \mathbf{F}, \tag{2.13}$$

where the component matrices are defined by:

$$\mathbf{M}(T) = \int_{\Omega} \mathbf{N}^T \; \rho c \left(\mathbf{x}, \mathbf{N}\hat{T}\right) \mathbf{N} \; d\Omega, \tag{2.14a}$$

$$\mathbf{K}(T) = \int_{\Omega} \mathbf{B}^T \; k \left(\mathbf{x}, \mathbf{N}\hat{T}\right) \mathbf{B} \; d\Omega, \tag{2.14b}$$

$$\mathbf{F} = \int_{\Omega} \mathbf{N}^T \; f\left(\mathbf{x}\right) \; d\Omega, \tag{2.14c}$$

with $\mathbf{B} := \nabla \mathbf{N}$. These values are denoted in the following as Mass Matrix, Stiffness Matrix, and Force Vector, following usual FEM naming convention. This semi-discrete form can now be employed for the solution of the posed problem, but a few elements required for its resolution are still missing. These are considered in the following subsections.

Basis functions

Having reached the matrix system which will be employed in the solution of the problem itself, it is now necessary to define the set of shape functions N_i used in the bases of approximations $\mathbf{X}^h$ and $\mathbf{V}^h$. These functions can have either a local (restricted to a region) or global (spanning the entire domain) supports depending on the requirements of the problem (see e.g. [33, 38].

In particular, suitable functions aim to provide:

- High quality of approximation
- Small round-off error accumulation with respect to polynomial degree increase
- Enforcement of continuity requirements along neighboring element boundaries with minimal effort
- Efficient computation of individual element matrices

Of the available choices, the present work elaborates on the orthogonal hierarchical basis functions introduced by [33]. In comparison to conventionally used Lagrange polynomials, integrated Legendre basis functions provide some important advantages. In particular, they provide a better overall matrix structure and condition number, which makes them particularly suitable for iterative solution methods [8]. They also satisfy an orthogonality property which brings the unique possibility of fast evaluation and easier enforcement of linear independence after refinement. Moreover, the hierarchic higher order bases are constructed by augmenting the lower order bases, meaning that activation and deactivation of higher order modes can be done in a simple manner. Figure 2.2 illustrates this effect for the first few polynomial orders.

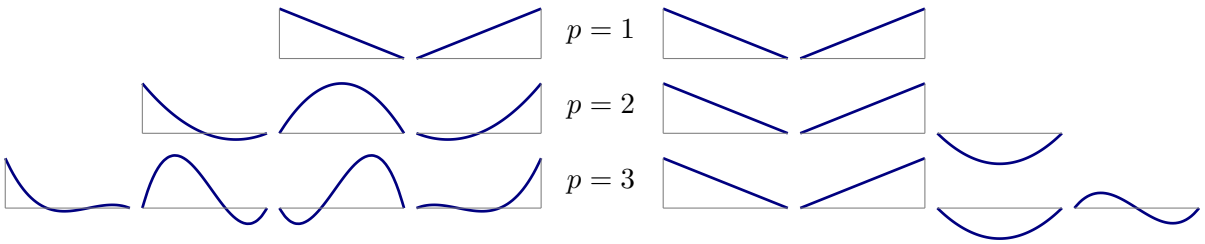


Figure 2.2: Comparison of one-dimensional shape functions (adapted from [36])

The 1D hierarchic shape functions can be described by:

$$N_1(\xi) = \frac{1-\xi}{2} \tag{2.15a}$$

$$N_2(\xi) = \frac{1+\xi}{2} \tag{2.15b}$$

$$N_n(\xi) = P_{n-1}(\xi) \quad n = 3, 4, \ldots, p+1 \tag{2.15c}$$

where P_n is defined as follows:

$$P_n(\xi) = \sqrt{\frac{2n-1}{2}} \int_{-1}^{\xi} L_{n-1}(x)\, dx = \frac{1}{\sqrt{4n-2}} \left(L_n(\xi) - L_{n-2}(\xi)\right) \quad k \geq 2, \tag{2.16}$$

and L_k represents the integrated Legendre polynomial:

$$L_k(\xi) = \frac{1}{2^k k!} \frac{d^k}{d/xi^k} \left(\xi^2 - 1\right)^k \quad k \in \mathbb{N}. \tag{2.17}$$

Higher dimensional versions of the hierarchic basis are constructed via a tensor product of the corresponding one-dimensional counterparts. For instance, in the two-dimensional case this corresponds to:

$$N_{i,j}(\xi, \eta) = N_i(\xi)\, N_j(\eta) \quad i \leq p_1 + 1,\ j \leq p_2 + 1. \tag{2.18}$$

Initial conditions

After definition of hierarchical basis functions, a starting state, i.e. initial condition, has to be considered for the solution. So far, only the PDE itself, including its corresponding boundary conditions, has been analyzed, but a starting state for the solution of transient cases has not yet been defined.

Since the application of the initial condition must be performed directly on the degrees of freedom of the discretization, it is necessary to project the target continuous function g onto this finite element space. There are two alternatives to do this, namely direct interpolation

onto solution nodes and L^2-projection. The latter of them provides an approximation in the integral sense [20], which has the advantage of also assigning the condition to non-nodal (i.e. higher order) modes.

This projection satisfies the following relation:

$$\int_\Omega (g - P_{L^2} g)\, v \; d\Omega = 0 \quad \forall v \in \mathbf{V}^h, \tag{2.19}$$

which minimizes the projection error under the L^2 norm. Under the Bubnov-Galerkin discretization described above, this is equivalent to solving the following linear system of equations:

$$\mathbf{M}\hat{g} = \mathbf{G}, \tag{2.20}$$

where

$$\mathbf{M} = \int_\Omega \mathbf{N}^T \mathbf{N} \; d\Omega \tag{2.21a}$$

$$\mathbf{G} = \int_\Omega \mathbf{N}^T \; g \; d\Omega \tag{2.21b}$$

and $\hat{g}$ are precisely the values to be assigned at the corresponding DOFs.

In the remainder of this work, the L^2-projection will be used not for the initial conditions, but also for projection of functions onto non-conforming discretizations.

Numerical integration

Lastly, the numerical evaluation of the integral terms needs to be discussed. Computation of the integrals in Equation 2.12 can be done using different quadrature rules, such as the Gaussian quadrature. Under this scheme, the integral of a function

$$I = \int_{-1}^{1} f\left(\xi\right) \; d\xi \tag{2.22}$$

is approximated by

$$I \approx \sum_{i=1}^{n} w_i f\left(\xi_i\right) \tag{2.23}$$

where w_i and ξ_i are, pre-specified weights and locations defined by the quadrature respectively. Similarly, n corresponds to the number of integration points used, which strongly affects the accuracy of the approximation. The quadrature rule considered to be exact for polynomials of order up to $2n - 1$. In addition to this, integrals in higher dimensions can

again be obtained by Cartesian product of the one-dimensional rule. When the integrands are discontinuous, a *composed integration* can be used. In this case, wherever a refinement exists, the integration points are distributed on the leaf elements instead [36].

In general, it in necessary to first remap the d-dimensional integration domain Ω onto the local $[-1,1]^d$ space, such that the aforementioned procedure can be applied. This leads to the following equation:

$$I = \int_{\Omega} f(\mathbf{x}) \; d\Omega = \int_{[-1,1]^d} f(Q(\xi)) \, |\mathbf{J}| \; d\Omega \approx \sum_{\mathbf{i} \preceq \mathbf{n}} w_{\mathbf{i}} f(Q(\xi_{\mathbf{i}})) \, |\mathbf{J}| \tag{2.24}$$

where

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \xi} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \\ & & \ddots \end{bmatrix}. \tag{2.25}$$

and $\mathbf{i} = (i_1, \ldots, i_d)$ and $\mathbf{n} = (n_1, \ldots, n_d)$ specify the individual integration points considered.

2.3 Material nonlinearities

As previously noted, due to temperature dependency of the material parameters, the system of equations 2.13 is nonlinear and requires special treatment. The most popular methods to resolve a nonlinearity in material coefficients are general root finding methods, the Newton-Raphson method being one of the most commonly used [19].

Rewriting Equation 2.13 in residual form yields:

$$\mathbf{R}(T) = \mathbf{F} - \mathbf{K}(T)T - \mathbf{M}(T)\dot{T} \stackrel{!}{=} 0 \tag{2.26}$$

where the hat denoting that the variable corresponds to its DOF values is dropped for a simpler notation. The residual can then be linearized with the help of its Taylor series expansion [4] as follows:

$$\begin{aligned} \mathbf{R}(T^{k-1} + \delta T) &= \mathbf{R}(T^{k-1}) + D\mathbf{R}\left(T^{k-1}\right)[\delta T] + H.O.T. \\ &\approx \mathbf{R}(T^{k-1}) + D\mathbf{R}\left(T^{k-1}\right)[\delta T] \stackrel{!}{=} 0. \end{aligned} \tag{2.27}$$

where $D\mathbf{R}\left(T^{k-1}\right)[\delta T]$ corresponds to the directional derivative of the residual function evaluated at the previous iteration value T^{k-1}, in the direction of increment δT. $H.O.T.$ are the higher order terms which are neglected in this approximation.

Setting the residual value after the increment to 0, which is satisfied by the solution to Equation 2.26, the system can be rewritten as follows:

$$- D\mathbf{R}\left(T^{k-1}\right)[\delta T] = \mathbf{R}(T^{k-1}), \tag{2.28}$$

and can be updated using the following equation:

$$T^k = T^{k-1} + \delta T. \tag{2.29}$$

This process is then repeated until convergence is achieved, using index k as iteration counter.

In the heat transfer problem considered, the residual function for an individual DOF value j (i.e. taking $v^h = N_j$) can be expressed as follows:

$$\begin{aligned} R_j\left(\hat{T}^{k-1}\right) = \int_\Omega f\left(\mathbf{x}\right) N_j \, d\Omega - \sum_{i=1}^{n_{\mathrm{DoF}}} \dot{\hat{T}}_i^{k-1} \int_\Omega N_j \, \rho c \left(\mathbf{x}, \sum_{l=1}^{n_{\mathrm{DoF}}} N_l \hat{T}_l^{k-1}\right) N_i \, d\Omega \\ - \sum_{i=1}^{n_{\mathrm{DoF}}} \hat{T}_i^{k-1} \int_\Omega B_j \, k \left(\mathbf{x}, \sum_{l=1}^{n_{\mathrm{DoF}}} N_l \hat{T}_l^{k-1}\right) B_i \, d\Omega, \end{aligned} \tag{2.30}$$

such that the derivative with respect to the i-th DOF will then be:

$$\begin{aligned} \frac{\partial R_j\left(\hat{T}^{k-1}\right)}{\partial \hat{T}_i} = & - \int_\Omega N_j \, \rho c \left(\mathbf{x}, \sum_{l=1}^{n_{\mathrm{DoF}}} N_l \hat{T}_l^{k-1}\right) N_i \, d\Omega \\ & - \left(\sum_{l=1}^{n_{\mathrm{DoF}}} N_l \dot{\hat{T}}_l^{k-1}\right) \int_\Omega N_j \, \left(\rho' c + \rho c'\right) \left(\mathbf{x}, \sum_{l=1}^{n_{\mathrm{DoF}}} N_l \hat{T}_l^{k-1}\right) N_i \, d\Omega \\ & - \int_\Omega B_j \, k \left(\mathbf{x}, \sum_{l=1}^{n_{\mathrm{DoF}}} N_l \hat{T}_l^{k-1}\right) B_i \, d\Omega \\ & - \left(\sum_{l=1}^{n_{\mathrm{DoF}}} B_l \hat{T}_l^{k-1}\right) \int_\Omega B_j \, k' \left(\mathbf{x}, \sum_{l=1}^{n_{\mathrm{DoF}}} N_l \hat{T}_l^{k-1}\right) N_i \, d\Omega, \end{aligned} \tag{2.31}$$

which can also be collected into a matrix form (similar to Equation 2.13) by:

$$- D\mathbf{R}\left(T^{k-1}\right)[\delta T] = \left(\mathbf{K}(T^{k-1}) + \mathbf{K}'(T^{k-1})\right) \delta T + \left(\mathbf{M}(T^{k-1}) + \mathbf{M}'(T^{k-1})\right) \delta \dot{T}, \tag{2.32}$$

and the matrices in Equation 2.32 can be defined by:

$$\mathbf{K}(T^{k-1}) = \int_{\Omega} \mathbf{B}^T \; k\left(\mathbf{x}, \mathbf{N}\hat{T}^{k-1}\right) \mathbf{B} \; d\Omega \tag{2.33a}$$

$$\mathbf{K}'(T^{k-1}) = \int_{\Omega} \mathbf{B}^T \; k'\left(\mathbf{x}, \mathbf{N}\hat{T}^{k-1}\right) \left(\mathbf{B}\hat{T}^{k-1}\right) \mathbf{N} \; d\Omega \tag{2.33b}$$

$$\mathbf{M}(T^{k-1}) = \int_{\Omega} \mathbf{N}^T \; \rho c\left(\mathbf{x}, \mathbf{N}\hat{T}^{k-1}\right) \mathbf{N} \; d\Omega \tag{2.33c}$$

$$\mathbf{M}'(T^{k-1}) = \int_{\Omega} \mathbf{N}^T \; \left[\rho' c + \rho c'\right]\left(\mathbf{x}, \mathbf{N}\hat{T}^{k-1}\right) \left(\mathbf{N}\dot{\hat{T}}^{k-1}\right) \mathbf{N} \; d\Omega \tag{2.33d}$$

2.4 Time discretization

Finally, since the heat transfer problem does not always correspond to a stationary situation where the term $\dot{T}$ from the Equation 2.13 can be ignored, it is necessary to describe possible methods to discretize it. Two alternatives are considered in the scope of the current work: one-step algorithms, which explicitly substitute the time derivative, and the space-time approach, in which the time variable is handled in the same manner as the spatial variables.

2.4.1 Time stepping

As the name implies, time stepping methods advance the transient solution step-by-step and, therefore, essentially compute stationary solutions at each time state. Among them, one-step algorithms are the most common alternatives, given their simplicity. Algorithms of the θ-family, also referred to as generalized trapezoidal methods, are typically used to discretize this class of time-dependent problems [12]. The scheme expresses the expected time derivative as a combination of the current and previous stationary solutions. Thus, the following system can be formulated:

$$\begin{cases} \mathbf{M}\mathbf{v}_{n+1} + \mathbf{K}\mathbf{u}_{n+1} = \mathbf{F}_{n+1} & (2.34a) \\ \mathbf{u}_{n+1} = \mathbf{u}_n + \delta t \; \mathbf{v}_{n+\theta} & (2.34b) \\ \mathbf{v}_{n+\alpha} = (1-\alpha)\,\mathbf{v}_n + \alpha \mathbf{v}_{n+1} & (2.34c) \end{cases}$$

where $\mathbf{u}_n \approx T(t_n)$ and $\mathbf{v}_n \approx \dot{T}(t_n)$ are the computed approximations to the temperature field and its derivative respectively, and δt is the considered time step. Coefficient θ can then take values between 0 and 1 with some typical values summarized in Table 2.1. In the current work, the implicit Backward Euler approach is used. This avoids potential stability problems which may appear from an explicit method such as the Forward Euler approach.

Rearranging Equation 2.34a for the increment of the solution in time, $\Delta T = T_{n+1} - T_n$, the

Table 2.1: Members of the θ-family of one-step methods

θ	Method
0	Forward Euler
$\frac{1}{2}$	Crank-Nicolson
1	Backward-Euler

following system of equations with time step update can be formulated:

$$\begin{cases} \left(\frac{1}{\Delta t}\mathbf{M} + \mathbf{K}\right) \Delta T = \mathbf{F}_{n+1} - \mathbf{K}T_n & \text{(2.35a)} \\ T_{n+1} = T_n + \Delta T & \text{(2.35b)} \end{cases}$$

2.4.2 Space-Time approach

As an alternative to time-stepping methods, the space-time approach is studied in this work. Time is now treated as an additional one-dimensional variable, discretized using shape functions. The finite element problem formulation should now include the newly discretized variable via a tensor product with the spatial variables [5] as illustrated in Figure 2.3. The other aspects covered in Section 2.2 can however be handled in the same way as before.

To begin with the space-time formulation, the space-time domain and its boundary are now defined as follows:

$$Q = \Omega \times (t_0, t_1) \tag{2.36a}$$
$$\delta Q = \underbrace{\Gamma \times (t_0, t_1)}_{:=P} \cup \Omega \times \{t_0\} \cup \Omega \times \{t_1\} \tag{2.36b}$$

where P is referred to as the lateral boundary of the problem, and the boundary $\Omega \times \{t_0\}$ takes on the role of the initial condition.

In this scenario, the directional derivatives of the unknown temperature field can be split into space $\nabla_{\mathbf{x}}$ and time ∇_t derivatives using the tensor space construction:

$$\nabla \left(T\left(\mathbf{x}, t\right)\right) = \left(\nabla_{\mathbf{x}} T, \nabla_t T\right) = \left(\frac{\partial T}{\partial x_1}, \cdots, \frac{\partial T}{\partial x_d}, \frac{\partial T}{\partial t}\right) \tag{2.37}$$

where d is the number of space dimensions. The problem itself, corresponding to the space-time version of Equation 2.7 can now be defined as follows:

$$\int_Q \rho c\left(\mathbf{x}, T\right) \nabla_t T \; v \; dQ + \int_Q \nabla_{\mathbf{x}} v \; k\left(\mathbf{x}, T\right) \nabla_{\mathbf{x}} T \; dQ = \int_Q f\left(\mathbf{x}\right) v \; dQ + \int_{\delta Q} k\left(\mathbf{x}, T\right) \nabla_{\mathbf{x}} T \; v \; d\left(\delta Q\right), \tag{2.38}$$

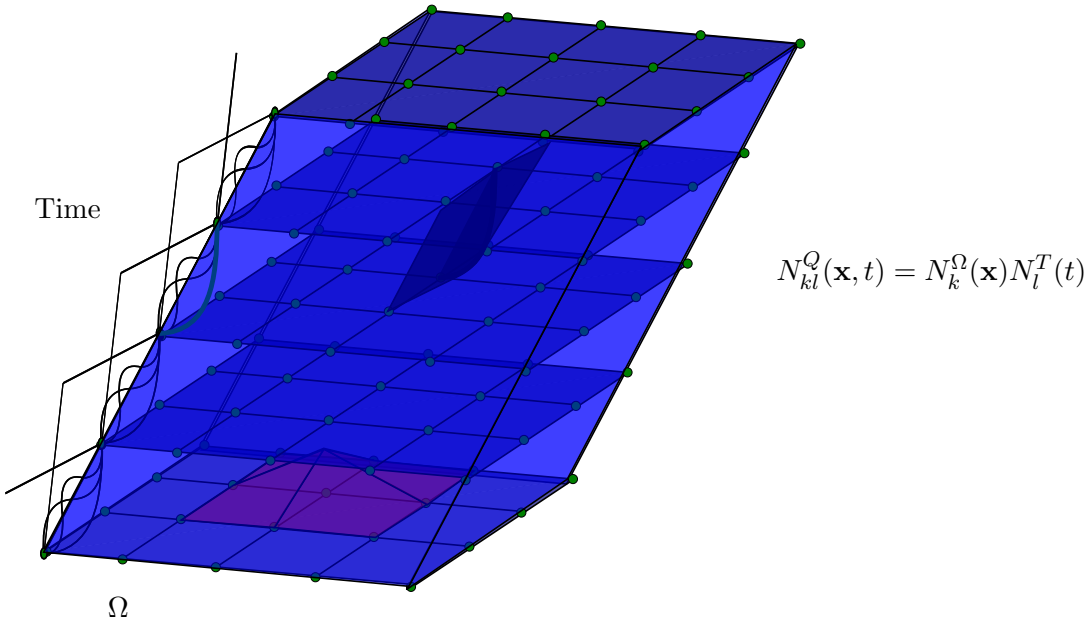


Figure 2.3: Domain definition in the space-time approach

noting that the integrals are now computed over the entire space-time domain. In addition to this, the shape functions of the finite element space are now also dependent on time, such that instead of $N_i(\mathbf{x})$ we now have $N_i(\mathbf{x},t)$, and the discretized versions of the trial solutions and test functions are:

$$T(\mathbf{x},t) \approx T^h(\mathbf{x},t) = \sum_{i=1}^{n_{\text{DoF}}} N_i^T(\mathbf{x},t)\,\hat{T}_i, \tag{2.39}$$

$$v(\mathbf{x},t) \approx v^h(\mathbf{x},t) = \sum_{j=1}^{n_{\text{DoF}}} N_j^v(\mathbf{x},t)\,\hat{v}_j. \tag{2.40}$$

These approximations ultimately lead to a system of equations in the following form:

$$(\mathbf{M}(T) + \mathbf{K}(T))\,\hat{T} = \mathbf{F}, \tag{2.41}$$

with

$$\mathbf{M}(T) = \int_Q \mathbf{N}^T \, \rho c \left(\mathbf{x}, \mathbf{N}\hat{T}\right) \mathbf{B}_t \, dQ, \tag{2.42a}$$

$$\mathbf{K}(T) = \int_Q \mathbf{B}_\mathbf{x}^T \, k \left(\mathbf{x}, \mathbf{N}\hat{T}\right) \mathbf{B}_\mathbf{x} \, dQ, \tag{2.42b}$$

$$\mathbf{F} = \int_Q \mathbf{N}^T \, f\left(\mathbf{x}\right) \, dQ. \tag{2.42c}$$

Petrov-Galerkin approach in space-time

Up to this point, the Bubnov-Galerkin approach was considered for the choice of test functions used. In the space-time formulation, this would be equivalent to having:

$$T \in \mathbf{X} \subseteq H^1(\Omega) \times H^1\left(\Theta\right) \tag{2.43a}$$

$$v \in \mathbf{V} \subseteq H^1(\Omega) \times H^1\left(\Theta\right), \tag{2.43b}$$

where $\Theta = (t_0, t_1)$ represents the considered time interval. However, now that the time direction is one of the dimensions of the finite element discretization, a closer look at Equation 2.7, and its mass term in particular, suggests the usage of an alternative discretization scheme:

$$\int_\Omega \rho c\left(\mathbf{x}, T\right) \underbrace{\frac{\partial T}{\partial t}}_{} \underbrace{v}_{\in H^1(\Theta)} \, d\Omega \quad \Rightarrow \frac{\partial T}{\partial t} \in L^2(\Theta), v \in H^1\left(\Theta\right) \subset L^2(\Theta). \tag{2.44}$$

Whilst the term coming from the primary variable can correspond to any function in $L^2(\Theta)$, the corresponding test space function is limited to $H^1(\Theta)$, with which the energy decreasing property of the scheme cannot be verified and, due to this, the scheme cannot be guaranteed to always be stable either [30]. More formally:

$$\exists T \in H^1(\Theta) \text{ s.t. } \frac{\partial T}{\partial t} \in L^2(\Theta) \backslash H^1\left(\Theta\right) \; \therefore \frac{\partial T}{\partial t} \neq v \; \forall v \in H^1(\Theta). \tag{2.45}$$

One way to recover the energy decreasing property and assure the stability of the method is to instead apply a Petrov-Galerkin concept, changing the space of test functions to one that also considers the previously missing functions. This formulation then leads to spaces:

$$T \in \mathbf{X} \subseteq H^1(\Omega) \times H^1\left(\Theta\right) \tag{2.46a}$$

$$v \in \mathbf{V} \subseteq H^1(\Omega) \times L^2\left(\Theta\right), \tag{2.46b}$$

and the mass term in Equation 2.44 now satisfies:

$$\int_{\Omega} \rho c\left(\mathbf{x}, T\right) \underbrace{\frac{\partial T}{\partial t}}_{} \underbrace{v}_{\in L^2(\Theta)} \, d\Omega \quad \Rightarrow \frac{\partial T}{\partial t}, v \in L^2(\Theta), \tag{2.47}$$

so that the required conditions for stability of the method are again met.

Space-time slab partitioning

Adding an additional dimension to the problem domain increases significantly the complexity of its solution, sometimes leading to an impossibly large system which can no longer be resolved with the same hardware setup. In this cases, it is useful to partition the augmented domain into a sequence of subdomains $Q_n = \Omega \times (t_n, t_{n+1})$, denoted *space-time slabs* which can then be solved independently.

Since the solution does not necessary have to be continuous across slabs, superscripts $+$ and $-$ are introduced to distinguish solutions $T^h(t_n^-)$ corresponding to $Q_{n-1} = \Omega \times (t_{n-1}, t_n)$ and $T^h(t_n^+)$ corresponding to $Q_n = \Omega \times (t_n, t_{n+1})$. Moreover, the functions in test space $\mathbf{V}^h$ are also no longer required to be continuous over the entire time domain, but only piecewise-continuous on each slab. For the values t_n which separate the space-time slabs, a jump operator analogous to the one from a discontinuous Galerkin approach is introduced:

$$[[v^h\left(t_n\right)]] = v^h\left(t_n^+\right) - v^h\left(t_n^-\right). \tag{2.48}$$

This procedure also requires defining the initial conditions $\Omega \times \{t_n\}$ for each slab, what is typically done by weak enforcement of $T^h(t_n^-) \approx T^h(t_n^+)$. This corresponds to taking the values from slab Q_{n-1} and setting them as initial conditions for the Q_n slab [14].

Therefore, solving the problem under this formulation involves independent solutions to Equation 2.41 for each time slab. This is similar to a time-stepping approach, with the possibility of having a higher order approximation for the time variable, as well as resolution of multiple elements in the time direction within the same time slab.

Chapter 3

Refinement schemes for the finite element method

Since the additive manufacture problem includes local effects in both space and time, it is necessary to be able to represent the effects occurring at the different scales precisely. Specifically, the moment and the region where the laser stroke is being applied should be captured with a higher precision than the rest of the domain to solve the high temperature gradients. Only then is it possible to accurately represent both the laser's effect in itself, and its influence over the rest of the domain. Since the refinement must be applied in both space and time, and each of them has it's own nuances - at least in traditional time-stepping methods, they are analyzed separately in the following chapter. Special interest is given to superposition refinement, which is part of the method analyzed in this thesis.

3.1 Mesh Refinement strategies

We first look at refinement of the spatial mesh. Mesh refinement is typically performed under one of two main strategies: refinement by replacement and replacement by superposition, with the first type being historically more popular. In the case of multi-scale phenomena, however, classic strategies such as h- and p-refinement, can be prohibitively expensive after a certain number of elements or a certain degree.

3.1.1 Refinement by replacement

As indicated previously, the classical h- and p-refinement methods belong in the refinement by replacement category. In this case, a region of the original mesh is modified or substituted to obtain a finer discretization, either by usage of smaller elements in the region of interest (h-refinement) or of higher order elements (p-refinement). Multiple varieties of each method exists, and both can even be combined into what are known as hp-refinement techniques. This third group of techniques combine the respective advantages of the previous methods, but can be harder to implement. They ultimately provide a locally higher number of degrees of freedom at the location of interest, and the possibility to have a more accurate approximation of the solution functions respectively [38].

h–refinement

h-refinement is most commonly achieved in one of the three ways depicted in Figure 3.1. In this case, refinement tends to be focused around features such as singularities which cannot be resolved in the original mesh. *Element subdivision*, as the name suggests, consists of partitioning elements from the original mesh, where the error is the largest, into smaller elements. However, since the boundaries of the original elements remain unchanged in this approach, newly generated refined nodes will not match any surrounding nodes from the original mesh. These mismatching nodes are known as *hanging nodes* and must be treated specially to conserve solution compatibility (i.e. continuity across elements). Different alternatives to do so include constraining in post-refinement steps [7], enhancing the coarse elements with the newly added modes [10], and extending the refinement region such that the change occurs smoothly over a larger area [29].

The second variant of *h*-refinement corresponds to *remeshing* (Figure 3.1c). While similar to the transition-element strategy of dealing with hanging nodes, in this case a new mesh is generated in order to reduce the estimated errors. This is less implementationally involved, but requires higher computational power due to the generation of the new mesh and the corresponding transfer of existing results.

Finally, the *r*-refinement (Figure 3.1d) consists in a version of *h*-refinement in which the total number of DOFs is kept constant, but the existing nodes are relocated to provide a higher resolution of the area to refine. Naturally, if the original mesh failed to capture the local phenomena, this refinement does not provide better results [38].

p–refinement

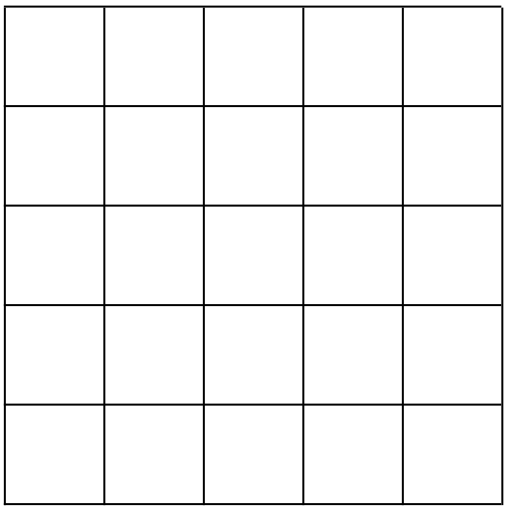
In the case of *p*-refinement, as previously indicated, the mesh itself stays unmodified, but its elements are enhanced by adding higher order modes. This can be done either globally, raising the polynomial degree everywhere, or locally, such that only the region of interest is refined. In the case of hierarchic basis functions, less computational effort is required since the preexisting shape functions are kept in the new bases (see Section 2.2).

hp–refinement

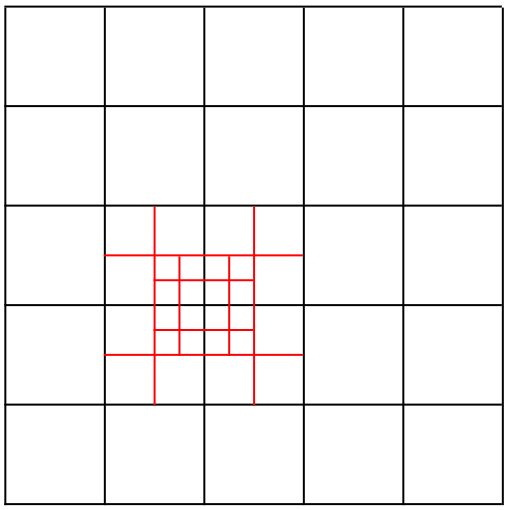
Finally, the two previous approaches can also be used together. This is especially true in the presence of singularities or other effects affecting solution smoothness. In this cases, *h*-refinement is performed near the non-smooth region, while *p*-refinement is performed further away. This leads to fine linear elements close to the special features, while the order and element sizes gradually increase further away. The described approach is able to maintain exponential convergence even when the individual *h*- and *p*-refinements don't [8].

3.1.2 Refinement by superposition

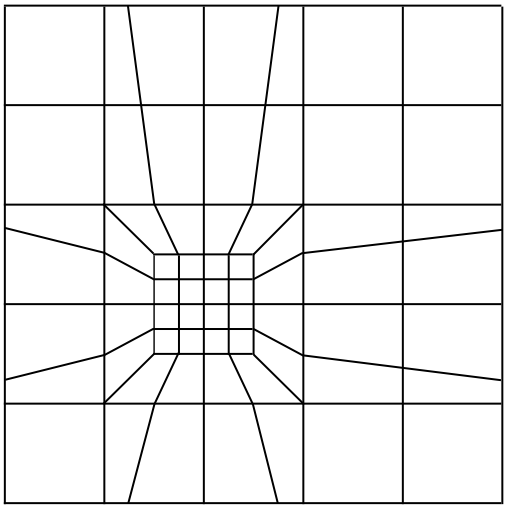
The other possible refinement strategy consists in the split of the solution into (at least) two components: the overarching global solution and the highly resolved local one. The solution



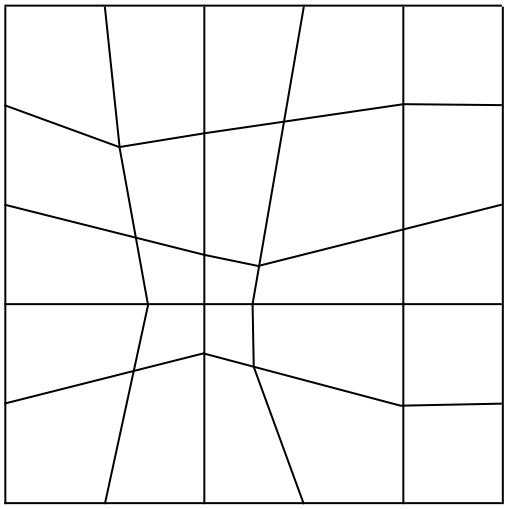
a) Original mesh



b) Element subdivision



c) Remeshing



d) r-refinement

Figure 3.1: *h*-refinement techniques (adapted from [38])

is then computed on the original coarse base mesh and one or more finer overlay levels. Methods in this category include the multigrid method [15], the *s*-method for superposition of independent meshes [9], the extended finite element method [24], among others. To ensure solution compatibility, the overlaid levels are typically constrained to satisfy homogeneous Dirichlet boundary conditions at the transition regions, therefore also removing the need to deal with the potential hanging nodes.

In the following, two of these superposition methods remain in focus: the multiscale $hp-d$ approach introduced by Rank [28, 27] for linear stationary problems and applied by Korshunova to the nonlinear transient heat equation [18], and the multi-level *hp* approach by Zander [35]. Moreover, the partitioned solution of these methods is also briefly outlined.

The multiscale $hp-d$ method

In the multiscale $hp-d$ method, a coarse base mesh Ω_b spanning the complete computational domain Ω is used to capture the global-scale behavior. In order to improve the solution quality, a linear element overlay mesh is defined over the region of interest Ω_o in order to resolve local-scale effects. One important restriction to the overlay mesh is that the support of an overlay element cannot intersect a base element's boundary. That is, individual elements in the overlay should be completely contained within a single base element. This condition simplifies the communication between the two solutions, especially with respect to coupling terms, such that the total solution of the system is then:

$$T_{\text{total}} = T_b + T_o. \tag{3.1}$$

The validity of the method relies on the fulfillment of two conditions: compatibility and linear independence. *Compatibility*, as noted previously, is satisfied by restriction of the overlay boundary $\Gamma_o \setminus (\Gamma_b \cap \Gamma_o)$ via homogeneous Dirichlet boundary conditions, immediately ensuring C^0-continuity across element boundaries. The common boundary $(\Gamma_b \cap \Gamma_o)$ is, however restricted to the global boundary conditions, since the overlay provides additional approximation capabilities. On the other hand, *linear independence* is a result of separating the linear and higher order modes into overlay and base levels respectively: as long as no modes are repeated for a single location, the orthogonality of the hierarchic shape functions used naturally ensures linear independence. In this case, it is only necessary to remove the linear modes corresponding to base mesh nodes. Figure 3.2 illustrates these two conditions on a sample problem.

In principle, the method was conceived considering a single overlay level with linear modes only. However, it is possible to extend the method naturally to incorporate more overlays or introduce higher order modes on the overlay as well (see e.g. [31]).

The multi-level *hp*–method

The concept of the multi-level *hp*-method, illustrated in Figure 3.3, is similar to that of the multiscale $hp-d$. However, the main difference between the two corresponds to the usage of hierarchically overlaid higher order meshes.

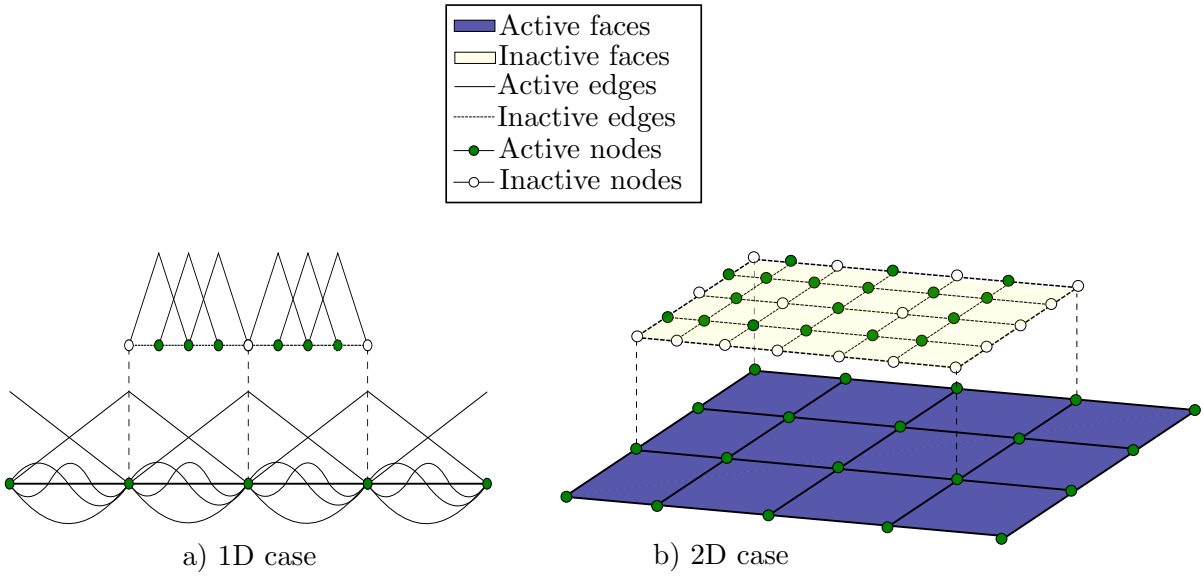


Figure 3.2: Multiscale $hp-d$ mesh construction (adapted from [35])

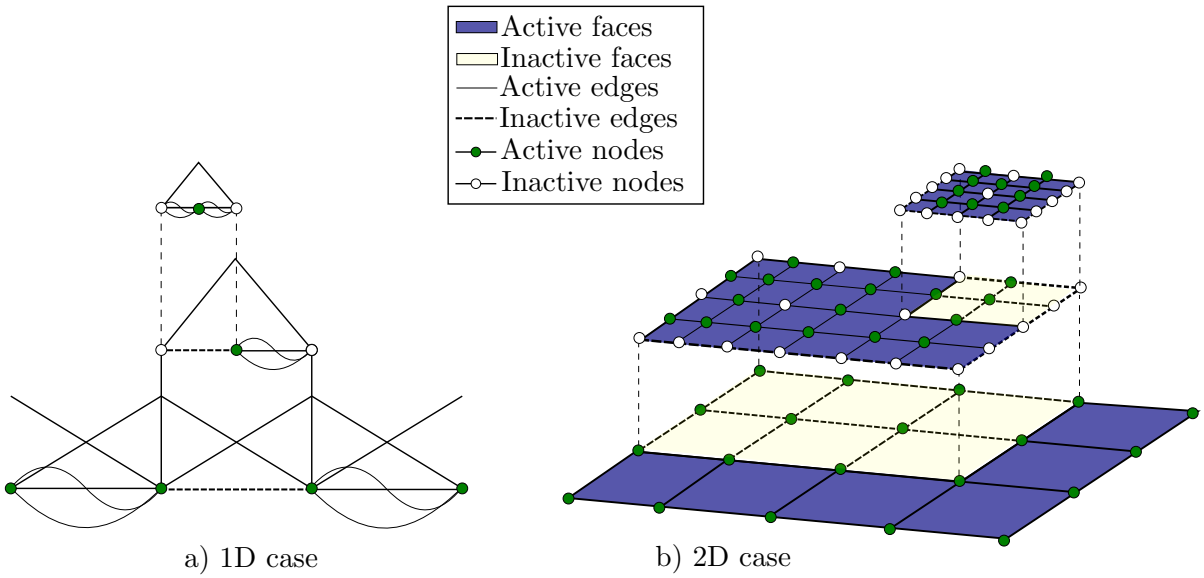


Figure 3.3: Multi-level hp mesh construction (adapted from [35])

As before, compatibility of the different levels is satisfied by application of homogeneous Dirichlet boundary conditions. Nonetheless, linear independence is guaranteed this time by introduction of the high-order shape functions only on leaf elements instead. Therefore, lower levels contain only linear modes over the overlaid region - decreasing its individual approximation capabilities, but providing an overall better solution when summed with the overlay solutions.

Partitioned solution of the scales

Both of these approaches (multiscale $hp-d$ and multi-level hp) can be resolved not only monolithically, as they are usually formulated, but also in a partitioned manner. In this case, the solutions T_b and T_o are computed from separate problems as part of a coupled equation system of the form:

$$\begin{bmatrix} \mathbf{A}_{bb} & \mathbf{A}_{bo} \\ \mathbf{A}_{ob} & \mathbf{A}_{oo} \end{bmatrix} \begin{bmatrix} T_b \\ T_o \end{bmatrix} = \begin{bmatrix} \mathbf{b}_b \\ \mathbf{b}_o \end{bmatrix} \Rightarrow \begin{cases} \mathbf{A}_{bb} T_b = \mathbf{b}_b - \mathbf{A}_{bo} T_o \\ \mathbf{A}_{oo} T_o = \mathbf{b}_o - \mathbf{A}_{ob} T_b \end{cases}, \tag{3.2}$$

where the coupling terms are included into the right hand side vector. The system of equations can then be solved by different schemes such as the block-partitioned Jacobi, Gauss-Seidel, and Successive over-relaxation methods among others [26], while each of the individual linear systems can be solved by any suitable linear solver.

In the following, the Gauss-Seidel approach is employed due to the reduced number of required iterations in comparison to the Jacobi method. Moreover, for the method to converge, the following condition must be satisfied [26]:

$$\rho\left(\left[\mathbf{D}(x) - \mathbf{L}(x)\right]^{-1}\left[\mathbf{U}(x)\right]\right) < 1 \tag{3.3}$$

where U, D, and L correspond to the upper triangular, diagonal and lower triangular components of the system matrix ($A = D - L - U$), and rho represents the spectral radius. This will be used as a measure of the method's stability in the remainder of the document.

3.2 Refinement in Time

Now that the local spatial effect has been addressed, it is necessary to also consider ways to deal with refinement in time, since the laser application is also a sudden and rapid event in the scope of the overall simulation time. In this case, time-stepping approaches allow two ways to improve solution results: higher order time approximations and sub-stepping techniques. Moreover, since the problem is intended to be solved in a partitioned manner, it is possible to restrict the finer sub-stepping to the overlaid region only, further diminishing the total required computational power. Only the latter strategy is considered here, since the former would also change approximation of the time derivative and hence the computed equations. The space-time approach, on the other hand, handles refinement in time naturally since the time direction is treated simply as another dimension. In this sense, all the previous ideas from mesh refinement can be adopted directly for the time dimension as well.

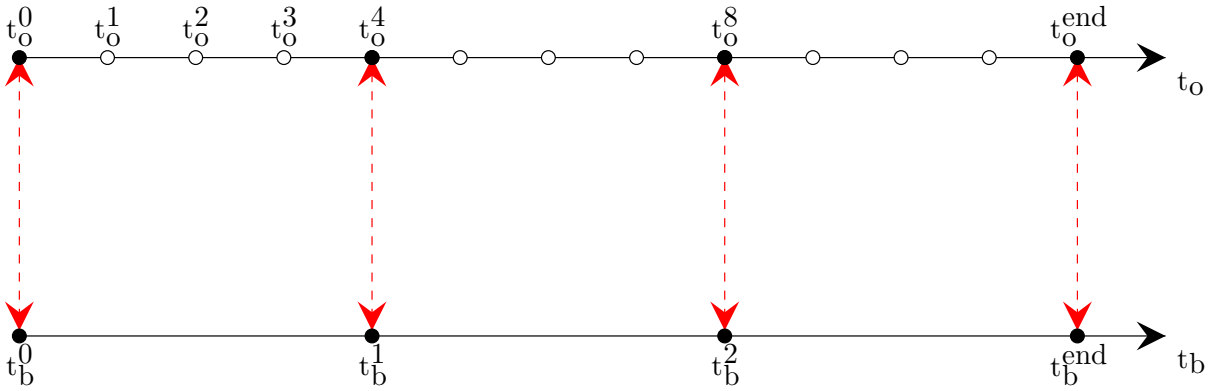


Figure 3.4: Time sub-stepping within partitioned approaches (from [18])

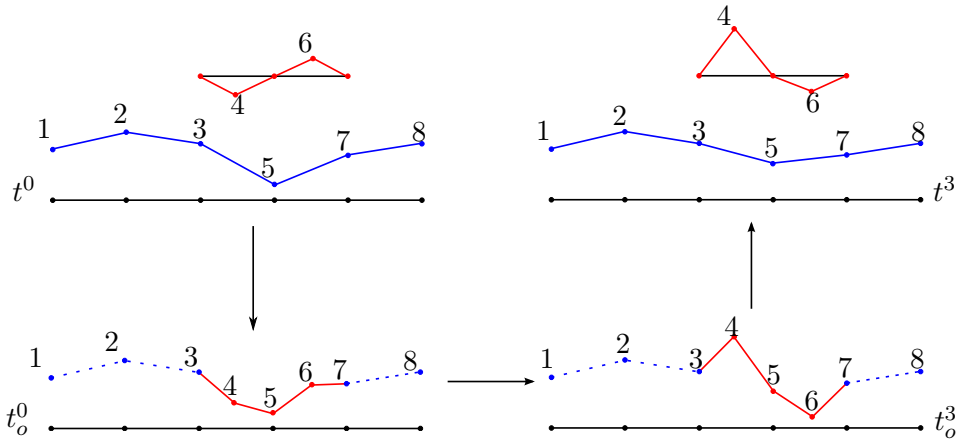


Figure 3.5: Overlay sub-stepping concept (from [18])

3.2.1 Time sub-stepping

Using an overlay as described in Section 3.1.2 allows the solution to be refined locally in time through the application of sub-stepping. In this case, additional finer time steps are used in the overlay problem only, as portrayed in Figure 3.4. Thus, the coupling between base and overlay problems occurs only each time the overlay solution reaches the same time as one of a base solution, namely at steps $t_o^4, t_o^8, \ldots$.

For the solution to be accurate, however, the overlay sub-steps must be able to affect the total solution across the overlaid region, i.e. including the base mesh modes [18]. This solution must then be projected into the individual variables T_b and T_o for coupling and further resolution of the problem. Figure 3.5 depicts this process.

3.2.2 Refinement in Space-Time

For the Space-Time finite element method, refinement in time is no different than regular mesh refinement since time is just treated as any other dimension. In particular, this means that order elevation is possible without adversely affecting the solution quality. In contrast

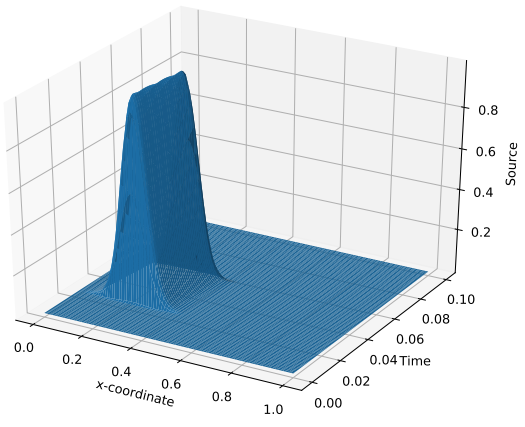


Figure 3.6: Space-time representation of emulated one-dimensional laser stroke application

to this, changing the time-stepping scheme would imply modifying the system to solve and, therefore, demanding a high amount of computational resources.

Secondly, refinement can be performed locally in time, meaning only around a specific time interval of interest. For instance, Figure 3.6 shows the space-time representation of a one-dimensional problem which emulates a laser stroke traveling only within a brief window of time with respect to the total simulated time interval. In this case, only the time surrounding the application of the laser (and slightly after, until temperature gradients have decreased) needs to be refined. In contrast to this, changing the overlay time step in a time-stepping scheme would require gradual changes, since abruptly altering the time step would strongly affect the estimation of the time derivative, potentially perturbing the solution around the region of interest, thereby introducing another source of error.

Chapter 4

Implementation of the partitioned Multi-level *hp*–Finite Element Method for heat transfer problems

Having laid out the individual elements involved in the solution of problems in heat transfer analysis, the Multi-level *hp*–FEM is applied in this chapter. To highlight the key implementational aspects involved, a series of different cases is presented incrementally: linear stationary, linear transient, nonlinear stationary, and nonlinear transient. For each case, the full derivation starting from the governing equations and a summary of the applied algorithms are presented. The numerical results are then verified against the in-house high-order FEM code AdhoC++, which includes a monolithic implementation of the multi-level *hp*–FEM.

4.1 The linear stationary heat equation

The first case to be analyzed is the linear stationary case, which is typical of end-state (rather than process evolution) computations on systems with physical properties which are approximately constant with respect to temperature. In this situation, the nonlinearities and inertial terms from Equation 2.8 are neglected, leading to a single linear system to be solved. The first of these assumptions results in the simplification of the nonlinear functional $\mathscr{A}(T, k(\mathbf{x}, T), v)$ to a bilinear form which can be instead represented by $\mathscr{B}(T, v)$. It should be noted that the sign of the bilinear form is flipped with respect to the original to simplify further expressions throughout the formulation. That is

$$\mathscr{B}(T, v) := -\mathscr{A}(T, k(\mathbf{x}, \not{T}), v)\,. \tag{4.1}$$

At the same time, the second simplification allows the time-dependent terms (i.e. the internal product $\langle \rho c(\mathbf{x})\dot{T}, v \rangle$) to be neglected.

4.1.1 Partitioned problem formulation

Once these changes are considered, the governing equation can be expressed as follows:

$$\mathscr{B}(T, v) = \mathscr{F}(v) \quad \forall v \in \mathbf{V}. \tag{4.2}$$

Furthermore, the multi-level principle - here with only two levels, denoted as *base* and *overlay*, can be applied to yield:

$$\mathscr{B}(T_b + T_o, v_b + v_o) = \mathscr{F}(v_b + v_o) \quad \forall v_b \in \mathbf{V}_b, \forall v_o \in \mathbf{V}_o. \tag{4.3}$$

Now, since the base and overlay finite element spaces were defined to be linearly independent ($\mathbf{V} = \mathbf{V}_b \oplus \mathbf{V}_o$), the previous system can be split into a system of two independent equations

$$\begin{cases} \mathscr{B}(T_b + T_o, v_b) = \mathscr{F}(v_b) & \forall v_b \in \mathbf{V}_b \quad (4.4a) \\ \mathscr{B}(T_b + T_o, v_o) = \mathscr{F}(v_o) & \forall v_o \in \mathbf{V}_o, \quad (4.4b) \end{cases}$$

which can be further decomposed by applying the bilinear property of the form $\mathscr{B}(\cdot, \cdot)$:

$$\begin{cases} \mathscr{B}(T_b, v_b) + \mathscr{B}(T_o, v_b) = \mathscr{F}(v_b) & \forall v_b \in \mathbf{V}_b \quad (4.5a) \\ \mathscr{B}(T_b, v_o) + \mathscr{B}(T_o, v_o) = \mathscr{F}(v_o) & \forall v_o \in \mathbf{V}_o. \quad (4.5b) \end{cases}$$

After discretizing Equations 4.5a and 4.5b the matrix form of the system can be written as

$$\begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bo} \\ \mathbf{K}_{ob} & \mathbf{K}_{oo} \end{bmatrix} \begin{bmatrix} T_b \\ T_o \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \mathbf{F}_o \end{bmatrix} \tag{4.6}$$

where the terms used are represented by:

$$\mathbf{K_{bb}} = \int_{\Omega_b} \mathbf{B}_b^T k(\mathbf{x}) \mathbf{B}_b \, d\Omega \tag{4.7a}$$

$$\mathbf{K_{bo}} = \int_{\Omega_o} \mathbf{B}_b^T k(\mathbf{x}) \mathbf{B}_o \, d\Omega \tag{4.7b}$$

$$\mathbf{K_{ob}} = \int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \mathbf{B}_b \, d\Omega \tag{4.7c}$$

$$\mathbf{K_{oo}} = \int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \mathbf{B}_o \, d\Omega. \tag{4.7d}$$

This system can then be solved monolithically, as in AdhoC++, or in a partitioned way depending on the treatment given to the two variables T_b and T_o. Under the block Gauss-Seidel scheme, the i-th iteration of the solution becomes the triangular system

$$\begin{bmatrix} \mathbf{K}_{bb} & 0 \\ \mathbf{K}_{ob} & \mathbf{K}_{oo} \end{bmatrix} \begin{bmatrix} T_b^i \\ T_o^i \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b - \mathbf{K}_{bo} T_o^{i-1} \\ \mathbf{F}_o \end{bmatrix} \tag{4.8}$$

where the updated base solution T_b^i is used immediately to solve for T_o^i. Thus, the individual systems solved at each iteration are

$$\begin{cases} \mathbf{K}_{bb} T_b = \mathbf{R}_b(T_o^{i-1}) & \text{(4.9a)} \\ \mathbf{K}_{oo} T_o = \mathbf{R}_o(T_b^i). & \text{(4.9b)} \end{cases}$$

Notably, given the linear nature of the problem considered, the matrices only need to be computed once and hence no indices are used. Indeed, only the actual solution values T_b and T_o are updated after every iteration. A summary of the individual terms used for Equations 4.9a and 4.9b can be found in Table 4.1.

Table 4.1: Multi-level hp finite element matrices for a linear stationary heat equation resolved by the block Gauss-Seidel scheme

Term	Symbol	Definition
Base Stiffness Matrix	$\mathbf{K_{bb}}$	$\int_{\Omega_b} \mathbf{B}_b^T k(\mathbf{x}) \mathbf{B}_b \, d\Omega$
Overlay Stiffness Matrix	$\mathbf{K_{oo}}$	$\int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \mathbf{B}_o \, d\Omega$
Base Right-hand-side Vector	$\mathbf{R_b}(T_o^{i-1})$	$\mathbf{F_b} - \mathbf{K_{bo}} T_o^{i-1}$
Overlay Right-hand-side Vector	$\mathbf{R_o}(T_b^i)$	$\mathbf{F_o} - \mathbf{K_{ob}} T_b^i$
Base Coupling Stiffness Matrix	$\mathbf{K_{bo}}$	$\int_{\Omega_o} \mathbf{B}_b^T k(\mathbf{x}) \mathbf{B}_o \, d\Omega$
Overlay Coupling Stiffness Matrix	$\mathbf{K_{ob}}$	$\int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \mathbf{B}_b \, d\Omega$
Base Source Vector	$\mathbf{F_b}$	$\int_{\Omega_b} \mathbf{N}_b^T f(\mathbf{x}) \, d\Omega$
Overlay Source Vector	$\mathbf{F_o}$	$\int_{\Omega_o} \mathbf{N}_o^T f(\mathbf{x}) \, d\Omega$

4.1.2 Method Implementation

Having outlined the equation system to solve and the terms involved in its calculation, Algorithm 1 describes the process implemented for its solution. In this case, since the coupling terms are considered directly as additional source terms on the right hand side of the equations, rather than first determining a rectangular coupling matrix and then multiplying it with the corresponding solution vector, the results of this operation are directly computed. This leads to the following expressions for the complete coupling terms:

$$\begin{cases} \mathbf{K}_{bo} T_o^{i-1} = \displaystyle\int_{\Omega_o} \mathbf{B}_b^T k(\mathbf{x}) \mathbf{B}_o \, d\Omega \, T_o^{i-1} = \int_{\Omega_o} \mathbf{B}_b^T k(\mathbf{x}) \varepsilon_o^{i-1} \, d\Omega & \text{(4.10a)} \\ \mathbf{K}_{ob} T_b^i = \displaystyle\int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \mathbf{B}_b \, d\Omega \, T_b^i = \int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \varepsilon_b^i \, d\Omega & \text{(4.10b)} \end{cases}$$

where $\varepsilon = \mathbf{B}T$ represents the solution strain, which for heat problems corresponds to the

temperature gradient.

Algorithm 1: Multi-level hp algorithm for linear stationary problems

define overlay location
create global base mesh and overlay mesh in specified region
remove degrees of freedom to ensure linear independence
compute stiffness matrices $\mathbf{K_{bb}}$ and $\mathbf{K_{oo}}$ and source terms $\mathbf{F_b}$ and $\mathbf{F_o}$
set Gauss-Seidel iteration counter $i \leftarrow 0$
while *not converged* **and** *below maximum iteration number* **do**
  Compute *base problem*
    compute coupling term $\mathbf{F}_{bo}^{i-1} = \int_{\Omega_o} \mathbf{B}_b^T k \varepsilon_o^{i-1} \, d\Omega$ (Equation 4.10a)
    apply base system constraints (boundary conditions)
    solve $\mathbf{K}_{bb} T_b^i = \mathbf{F}_b - \mathbf{F}_{bo}^{i-1}$ (Equation 4.9a)
  Compute *overlay problem*
    compute coupling term $\mathbf{F}_{ob}^{i} = \int_{\Omega_o} \mathbf{B}_o^T k \varepsilon_b^{i} \, d\Omega$ (Equation 4.10a)
    apply overlay system constraints (homogeneous Dirichlet)
    solve $\mathbf{K}_{oo} T_o^i = \mathbf{F}_o - \mathbf{F}_{ob}^{i}$ (Equation 4.9b)
  compute strain energy $\mathscr{U}(T^i) = \frac{1}{2}\left(\mathscr{B}(T_b^i, T_b^i) + 2\mathscr{B}(T_o^i, T_b^i) + \mathscr{B}(T_o^i, T_o^i)\right)$
  check convergence in strain energy
  $i \leftarrow i + 1$
compute total solution $T_t = T_b + T_o$

For the computation of the numerical results themselves, the shape function and integration rules described in Chapter 2 were used.

In addition to this, the strain energy of the system was used as a measure of convergence, since this represents the square of the temperature in the energy norm, and this value can be computed precisely since the problem is linear and stationary. However, it is necessary to note that the total strain energy is not just the sum of the individual base and overlay energies, but also contemplates the coupling. Indeed, the strain energy can be calculated by:

$$\begin{aligned}\mathscr{U}(T) = \frac{1}{2}\mathscr{B}(T,T) = \frac{1}{2}\left(\mathscr{B}(T_b, T_b + T_o) + \mathscr{B}(T_o, T_b + T_o)\right) \\ = \frac{1}{2}\left(\mathscr{B}(T_b, T_b) + 2\mathscr{B}(T_o, T_b) + \mathscr{B}(T_o, T_o)\right). \quad (4.11)\end{aligned}$$

4.2 The linear transient heat equation

As second study case, the linear transient heat equation is considered. Although nonlinear effects are still not considered, leading to the same simplification to a bilinear form as in Equation 4.1, the inertial terms coming from the inner product $\langle \rho c(\mathbf{x}) \dot{T}, v \rangle$ are now present. Therefore, the computation must now be performed with consideration of the time variable to describe how the solution evolves.

4.2.1 Partitioned problem formulation

Equation 2.8 is, under the specified simplifications, now described by:

$$\left\langle \rho c(\mathbf{x})\dot{T}, v \right\rangle + \mathscr{B}(T, v) = \mathscr{F}(v) \quad \forall v \in \mathbf{V}, \tag{4.12}$$

which again can be rewritten, under the two-level base-overlay split, as

$$\left\langle \rho c(\mathbf{x})(\dot{T}_b + \dot{T}_o), v_b + v_o \right\rangle + \mathscr{B}(T_b + T_o, v_b + v_o) = \mathscr{F}(v_b + v_o) \quad \forall v_b \in \mathbf{V}_b, \forall v_o \in \mathbf{V}_o. \tag{4.13}$$

Again considering linear independence and the bilinear property of the form $\mathscr{B}(\cdot,\cdot)$, as well as that of the internal product $\langle \cdot, \cdot \rangle$, the system can be further separated into:

$$\begin{cases} \left\langle \rho c(\mathbf{x})\dot{T}_b, v_b \right\rangle + \left\langle \rho c(\mathbf{x})\dot{T}_o, v_b \right\rangle + \mathscr{B}(T_b, v_b) + \mathscr{B}(T_o, v_b) = \mathscr{F}(v_b) \quad \forall v_b \in \mathbf{V}_b & \text{(4.14a)} \\ \left\langle \rho c(\mathbf{x})\dot{T}_b, v_o \right\rangle + \left\langle \rho c(\mathbf{x})\dot{T}_o, v_o \right\rangle + \mathscr{B}(T_b, v_o) + \mathscr{B}(T_o, v_o) = \mathscr{F}(v_o) \quad \forall v_o \in \mathbf{V}_o. & \text{(4.14b)} \end{cases}$$

With the addition of the mass matrix terms, the previous system can be represented in discretized matrix form by following the ideas from Section 2.2 as follows:

$$\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bo} \\ \mathbf{M}_{ob} & \mathbf{M}_{oo} \end{bmatrix} \begin{bmatrix} \dot{T}_b \\ \dot{T}_o \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bo} \\ \mathbf{K}_{ob} & \mathbf{K}_{oo} \end{bmatrix} \begin{bmatrix} T_b \\ T_o \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \mathbf{F}_o \end{bmatrix} \tag{4.15}$$

where the stiffness matrices retain their values from Equations 4.7a to 4.7d and the new Mass terms are defined by:

$$\mathbf{M_{bb}} = \int_{\Omega_b} \mathbf{N}_b^T \rho c(\mathbf{x}) \mathbf{N}_b \, d\Omega \tag{4.16a}$$

$$\mathbf{M_{bo}} = \int_{\Omega_o} \mathbf{N}_b^T \rho c(\mathbf{x}) \mathbf{N}_o \, d\Omega \tag{4.16b}$$

$$\mathbf{M_{ob}} = \int_{\Omega_o} \mathbf{N}_o^T \rho c(\mathbf{x}) \mathbf{N}_b \, d\Omega \tag{4.16c}$$

$$\mathbf{M_{oo}} = \int_{\Omega_o} \mathbf{N}_o^T \rho c(\mathbf{x}) \mathbf{N}_o \, d\Omega. \tag{4.16d}$$

For the solution of this system, the fully implicit Backward-Euler time discretization scheme is applied such that, for the n-th timestep, the system is not directly solved for the total temperature field values $T_{b,n}$ and $T_{o,n}$, but instead for its increments in time ΔT_b and ΔT_o.

These increments then satisfy:

$$\begin{cases} T_{b,n} = T_{b,n-1} + \Delta T_b & \text{(4.17a)} \\ T_{o,n} = T_{o,n-1} + \Delta T_o & \text{(4.17b)} \end{cases}$$

and the resulting system is then

$$\frac{1}{\Delta t}\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bo} \\ \mathbf{M}_{ob} & \mathbf{M}_{oo} \end{bmatrix}\begin{bmatrix} \Delta T_{b,n} \\ \Delta T_{o,n} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bo} \\ \mathbf{K}_{ob} & \mathbf{K}_{oo} \end{bmatrix}\begin{bmatrix} T_{b,n-1} + \Delta T_{b,n} \\ T_{o,n-1} + \Delta T_{o,n} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \mathbf{F}_o \end{bmatrix} \tag{4.18}$$

or, reordering all known terms to the right hand side:

$$\left(\frac{1}{\Delta t}\begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bo} \\ \mathbf{M}_{ob} & \mathbf{M}_{oo} \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bo} \\ \mathbf{K}_{ob} & \mathbf{K}_{oo} \end{bmatrix}\right)\begin{bmatrix} \Delta T_b \\ \Delta T_o \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \mathbf{F}_o \end{bmatrix} - \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bo} \\ \mathbf{K}_{ob} & \mathbf{K}_{oo} \end{bmatrix}\begin{bmatrix} T_{b,n-1} \\ T_{o,n-1} \end{bmatrix}. \tag{4.19}$$

In conjunction with the block Gauss-Seidel scheme, this leads to the following equation:

$$\begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}_{bb} + \mathbf{K}_{bb} & 0 \\ \frac{1}{\Delta t}\mathbf{M}_{ob} + \mathbf{K}_{ob} & \frac{1}{\Delta t}\mathbf{M}_{oo} + \mathbf{K}_{oo} \end{bmatrix}\begin{bmatrix} \Delta T_b^i \\ \Delta T_o^i \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b - \mathbf{K}_{bb}T_{b,n-1} - \mathbf{K}_{bo}T_{o,n-1} - \left(\frac{1}{\Delta t}\mathbf{M}_{bo} + \mathbf{K}_{bo}\right)\Delta T_o^{i-1} \\ \mathbf{F}_o - \mathbf{K}_{ob}T_{b,n-1} - \mathbf{K}_{oo}T_{o,n-1} \end{bmatrix} \tag{4.20}$$

for the i-th iteration of the n-th timestep. Alternatively, collecting the coupling terms together yields:

$$\begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}_{bb} + \mathbf{K}_{bb} & 0 \\ \frac{1}{\Delta t}\mathbf{M}_{ob} + \mathbf{K}_{ob} & \frac{1}{\Delta t}\mathbf{M}_{oo} + \mathbf{K}_{oo} \end{bmatrix}\begin{bmatrix} \Delta T_b^i \\ \Delta T_o^i \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b - \mathbf{K}_{bb}T_{b,n-1} - \frac{1}{\Delta t}\mathbf{M}_{bo}\Delta T_o^{i-1} - \mathbf{K}_{bo}\left(T_{o,n-1} + \Delta T_o^{i-1}\right) \\ \mathbf{F}_o - \mathbf{K}_{oo}T_{o,n-1} \qquad\qquad - \mathbf{K}_{ob}\ T_{b,n-1} \end{bmatrix}. \tag{4.21}$$

Having obtained the previous system, the individual linear systems solved at each iteration can be expressed as follows:

$$\begin{cases} \mathbf{J}_{bb}\Delta T_b = \mathbf{R}_{b,n}(T_b^{i-1}, T_o^{i-1}) & \text{(4.22a)} \\ \mathbf{J}_{oo}\Delta T_o = \mathbf{R}_{o,n}(T_b^{i}, T_o^{i-1}), & \text{(4.22b)} \end{cases}$$

where each of the terms used is defined in Table 4.2.

Table 4.2: Multi-level hp finite element matrices for a linear transient heat equation resolved by block Gauss-Seidel iterations under a Backward Euler scheme

Term	Symbol	Definition
Base Total Matrix	$\mathbf{J_{bb}}$	$\frac{1}{\Delta t}\mathbf{M_{bb}} + \mathbf{K_{bb}}$
Overlay Total Matrix	$\mathbf{J_{oo}}$	$\frac{1}{\Delta t}\mathbf{M_{oo}} + \mathbf{K_{oo}}$
Base Right-hand-side Vector	$\mathbf{R_{b,n}}(T_b^{i-1}, T_o^{i-1})$	$\mathbf{F_b} - \mathbf{K_{bb}}T_{b,n-1} - \frac{1}{\Delta t}\mathbf{M_{bo}}\Delta T_o^{i-1} - \mathbf{K_{bo}}\left(T_{o,n-1} + \Delta T_o^{i-1}\right)$
Overlay Right-hand-side Vector	$\mathbf{R_{o,n}}(T_b^{i}, T_o^{i-1})$	$\mathbf{F_o} - \mathbf{K_{oo}}T_{o,n-1} - \frac{1}{\Delta t}\mathbf{M_{ob}}\Delta T_b^{i} - \mathbf{K_{ob}}\left(T_{b,n-1} + \Delta T_b^{i}\right)$
Base/Overlay Stiffness Matrix†	$\mathbf{K}_{\alpha\beta}$	$\int_\Omega \mathbf{B}_\alpha^T k(\mathbf{x}) \mathbf{B}_\beta \, d\Omega$
Base/Overlay Mass Matrix†	$\mathbf{M}_{\alpha\beta}$	$\int_\Omega \mathbf{N}_\alpha^T \rho c(\mathbf{x}) \mathbf{N}_\beta \, d\Omega$
Base Source Vector	$\mathbf{F_b}$	$\int_{\Omega_b} \mathbf{N}_b^T f(\mathbf{x}) \, d\Omega$
Overlay Source Vector	$\mathbf{F_o}$	$\int_{\Omega_o} \mathbf{N}_o^T f(\mathbf{x}) \, d\Omega$

† Here the subindices α and β correspond to either **b** for base or **o** for overlay. The resulting term is the corresponding system or coupling matrix.

4.2.2 Method Implementation

As in the previous case, the process corresponding to the implemented code is described in Algorithm 2. For this implementation, the coupling terms can also be directly computed into their stiffness and mass components, as Equation 4.21 indicates. This leads to the following expressions for the coupling terms:

$$
\begin{cases}
\mathbf{K}_{bo}\left(T_{o,n-1} + \Delta T_o^{i-1}\right) = \displaystyle\int_{\Omega_o} \mathbf{B}_b^T k(\mathbf{x}) \mathbf{B}_o \, d\Omega \; \left(T_{o,n-1} + \Delta T_o^{i-1}\right) \\
\qquad\qquad = \displaystyle\int_{\Omega_o} \mathbf{B}_b^T k(\mathbf{x}) \varepsilon_{o,n}^{i-1} \, d\Omega & \text{(4.23a)} \\
\mathbf{K}_{ob}\left(T_{b,n-1} + \Delta T_b^{i}\right) = \displaystyle\int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \mathbf{B}_b \, d\Omega \; \left(T_{b,n-1} + \Delta T_b^{i}\right) \\
\qquad\qquad = \displaystyle\int_{\Omega_o} \mathbf{B}_o^T k(\mathbf{x}) \varepsilon_{b,n}^{i} \, d\Omega & \text{(4.23b)} \\
\mathbf{M}_{bo}\Delta T_o^{i-1} = \displaystyle\int_{\Omega_o} \mathbf{N}_b^T \rho c(\mathbf{x}) \mathbf{N}_o \, d\Omega \, \Delta T_o^{i-1} = \int_{\Omega_o} \mathbf{N}_b^T \rho c(\mathbf{x}) \delta_o^{i-1} \, d\Omega & \text{(4.23c)} \\
\mathbf{M}_{ob}\Delta T_b^{i} = \displaystyle\int_{\Omega_o} \mathbf{N}_o^T \rho c(\mathbf{x}) \mathbf{N}_b \, d\Omega \, \Delta T_b^{i} = \int_{\Omega_o} \mathbf{N}_o^T \rho c(\mathbf{x}) \delta_b^{i} \, d\Omega & \text{(4.23d)}
\end{cases}
$$

where ε now represents the current solution strain, which includes both the last timestep solution and the current increment values (i.e. $\varepsilon = \mathbf{B}\left(T_{n-1} + \Delta T\right)$), and $\delta = \mathbf{N}\Delta T$ corresponds

to the evaluated solution increment.

Algorithm 2: Multi-level hp algorithm for linear transient problems

define overlay location
create global base mesh and overlay mesh in specified region
remove degrees of freedom to ensure linear independence
set initial conditions $T_{b,0}$ and $T_{o,0}$
compute system matrices $\mathbf{K_{bb}}$, $\mathbf{K_{oo}}$, $\mathbf{M_{bb}}$, and $\mathbf{M_{oo}}$
set total base system matrix $\mathbf{J_{bb}} \leftarrow \frac{1}{\Delta t}\mathbf{M_{bb}} + \mathbf{K_{bb}}$
set total overlay system matrix $\mathbf{J_{oo}} \leftarrow \frac{1}{\Delta t}\mathbf{M_{oo}} + \mathbf{K_{oo}}$
$currentTime \leftarrow t_0$
for $n \leftarrow 1$ **to** *numberOfTimeSteps* **do**
  compute source terms $\mathbf{F_b}$ and $\mathbf{F_o}$
  $\Delta T_b \leftarrow 0, \Delta T_o \leftarrow 0$
  $currentTime \leftarrow currentTime + \Delta t$
  set Gauss-Seidel iteration counter $i \leftarrow 0$
  while *not converged* **and** *below maximum iteration number* **do**
    Compute *base problem*
      compute coupling term $\mathbf{F}_{bo}^{i-1} = \frac{1}{\Delta t}\mathbf{M}_{bo}\Delta T_o^{i-1} + \mathbf{K}_{bo}\left(T_{o,n-1} + \Delta T_o^{i-1}\right)$ (Equations 4.23c and 4.23a)
      apply base system constraints (boundary conditions)
      solve $\mathbf{J}_{bb}\Delta T_b^i = (\mathbf{F}_b - \mathbf{K_{bb}}\mathbf{T_{b,n-1}}) - \mathbf{F}_{bo}^{i-1}$ (Equation 4.22a)
    Compute *overlay problem*
      compute coupling term $\mathbf{F}_{ob}^{i} = \frac{1}{\Delta t}\mathbf{M}_{ob}\Delta T_b^{i} + \mathbf{K}_{ob}\left(T_{b,n-1} + \Delta T_b^{i}\right)$ (Equations 4.23d and 4.23b)
      apply overlay system constraints (homogeneous Dirichlet)
      solve $\mathbf{J}_{oo}\Delta T_o^i = (\mathbf{F}_o - \mathbf{K_{oo}}\mathbf{T_{o,n-1}}) - \mathbf{F}_{ob}^{i}$ (Equation 4.22b)
    compute L^2 norm of the complete solution
    check convergence in L^2 norm
    $i \leftarrow i + 1$
  $T_{b,n} \leftarrow T_{b,n-1} + \Delta T_b^i$
  $T_{o,n} \leftarrow T_{o,n-1} + \Delta T_o^i$
compute total solution

As mentioned in Chapters 2 and 3, the initial condition is obtained by L^2-projection, since it is provided as a continuous function. Moreover, the strain energy is no longer a valid convergence measure for the solution, since the energy norm value will be changing throughout the transient simulation process. An alternative convergence measure in this case corresponds to the L^2-norm of the solution (or, as before, its square):

$$\|T^\ell\|_{L^2}^2 = \|T_b^\ell + T_b^\ell\|_{L^2}^2 = \int_\Omega \left(T_b^\ell(\mathbf{x}) + T_o^\ell(\mathbf{x})\right)^2 \, d\Omega \tag{4.24}$$

4.3 The nonlinear stationary heat equation

Understanding now how to handle the problem itself, as well as transient effects, one last component must now be included: nonlinearities. The physical properties of the material(s) are normally temperature dependent and, therefore, the system to solve will no longer be linear in temperature. For simplicity, this effect is first isolated again from transient ones, leading to the nonlinear stationary case. As such, the internal product $\langle \rho c(\mathbf{x})\dot{T}, v\rangle$) can again be neglected, but the nonlinear form can no longer be simplified - meaning that the bilinear property does not apply anymore.

4.3.1 Partitioned problem formulation

The problem defined by Equation 2.8, after excluding the terms which are set to zero, can now be written as follows:

$$-\mathscr{A}\left(T, k(\mathbf{x}, T), v\right) = \mathscr{F}\left(v\right) \quad \forall v \in \mathbf{V}, \tag{4.25}$$

which can then be partitioned into different scales given that the linear independence between the test spaces defined at each level, namely $\mathbf{V} = \mathbf{V}_b \oplus \mathbf{V}_o$ when considering a base level and, in this case, a single overlay, still holds. This results in the equation system:

$$\begin{cases} -\mathscr{A}\left(T, k(\mathbf{x}, T), v_b\right) = \mathscr{F}\left(v_b\right) & \forall v_b \in \mathbf{V}_b \quad (4.26a) \\ -\mathscr{A}\left(T, k(\mathbf{x}, T), v_o\right) = \mathscr{F}\left(v_o\right) & \forall v_o \in \mathbf{V}_o \quad (4.26b) \end{cases}$$

or, in terms of the individual temperature fields, defined on the base and overlay spaces:

$$\begin{cases} -\mathscr{A}\left(T_b + T_o, k(\mathbf{x}, T_b + T_o), v_b\right) = \mathscr{F}\left(v_b\right) & \forall v_b \in \mathbf{V}_b \quad (4.27a) \\ -\mathscr{A}\left(T_b + T_o, k(\mathbf{x}, T_b + T_o), v_o\right) = \mathscr{F}\left(v_o\right) & \forall v_o \in \mathbf{V}_o. \quad (4.27b) \end{cases}$$

While the temperature terms are now coupled, the system after discretization can still be represented in matrix form by

$$\begin{bmatrix} \mathbf{K}_{bb}\left(T_b + T_o\right) & \mathbf{K}_{bo}\left(T_b + T_o\right) \\ \mathbf{K}_{ob}\left(T_b + T_o\right) & \mathbf{K}_{oo}\left(T_b + T_o\right) \end{bmatrix} \begin{bmatrix} T_b \\ T_o \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \mathbf{F}_o \end{bmatrix}, \tag{4.28}$$

where each of the involved matrices are now dependent on the full solution field (i.e. the base and overlay temperature fields). In this case, the expressions corresponding to each of these

terms are:

$$\mathbf{K_{bb}}\left(T_b + T_o\right) = \int_{\Omega_b} \mathbf{B}_b^T k\left(\mathbf{x}, T_b + T_o\right) \mathbf{B}_b \, d\Omega \tag{4.29a}$$

$$\mathbf{K_{bo}}\left(T_b + T_o\right) = \int_{\Omega_o} \mathbf{B}_b^T k\left(\mathbf{x}, T_b + T_o\right) \mathbf{B}_o \, d\Omega \tag{4.29b}$$

$$\mathbf{K_{ob}}\left(T_b + T_o\right) = \int_{\Omega_o} \mathbf{B}_o^T k\left(\mathbf{x}, T_b + T_o\right) \mathbf{B}_b \, d\Omega \tag{4.29c}$$

$$\mathbf{K_{oo}}\left(T_b + T_o\right) = \int_{\Omega_o} \mathbf{B}_o^T k\left(\mathbf{x}, T_b + T_o\right) \mathbf{B}_o \, d\Omega. \tag{4.29d}$$

At this point, the system can be solved under a range of nonlinear solvers, such as those described by [26]. In particular, some notable alternatives are the Newton-Gauss-Seidel and the m-step block Gauss-Seidel-Newton schemes, which both stem from the linear block Gauss-Seidel solver and the Newton method, but differ in the order that the linear block solver and nonlinearities are resolved in. Both of these, among others, were analyzed in [18] and, given the lower computational and implementational costs, only the latter is applied in this scenario. More precisely, the *one*-step block Gauss-Seidel-Newton scheme is described, given how the internal iterations required to resolve the Newton problem would imply recomputing the involved terms and, depending on the resolved problem, do not necessarily improve the method's convergence [18].

To proceed with the solution, the system in Equation 4.28 can be expressed in residual form as follows:

$$\begin{cases} R_b\left(T_b, T_o\right) = \mathscr{F}\left(v_b\right) + \mathscr{A}\left(T_b + T_o, k(\mathbf{x}, T_b + T_o), v_b\right) \overset{!}{=} 0 \quad \forall v_b \in \mathbf{V}_b & (4.30a) \\ R_o\left(T_b, T_o\right) = \mathscr{F}\left(v_o\right) + \mathscr{A}\left(T_b + T_o, k(\mathbf{x}, T_b + T_o), v_o\right) \overset{!}{=} 0 \quad \forall v_o \in \mathbf{V}_o, & (4.30b) \end{cases}$$

and the nonlinear Gauss-Seidel iterative process can then be described directly on the partial differential equations by:

$$\begin{cases} R_b\left(T_b^i, T_o^{i-1}\right) = \mathscr{F}\left(v_b\right) + \mathscr{A}\left(T_b^i + T_o^{i-1}, k(\mathbf{x}, T_b^i + T_o^{i-1}), v_b\right) \overset{!}{=} 0 \quad \forall v_b \in \mathbf{V}_b & (4.31a) \\ R_o\left(T_b^i, T_o^i\right) = \mathscr{F}\left(v_o\right) + \mathscr{A}\left(T_b^i + T_o^i, k(\mathbf{x}, T_b^i + T_o^i), v_o\right) \overset{!}{=} 0 \quad \forall v_o \in \mathbf{V}_o, & (4.31b) \end{cases}$$

where i represents the Gauss-Seidel iteration number. In this way, Equation 4.31a can be used to solve for T_b and Equation 4.31b for T_o by keeping the other variable fixed. However, since this is still a nonlinear system, it is necessary to first resolve the nonlinearity before proceeding with the system solution.

Following the m-step Gauss-Seidel-Newton scheme, the nonlinearities are treated by the New-

ton method. Once higher order terms are omitted, this provides the following linearizations:

$$\begin{cases} R_b\left(T_b^{i,k}, T_o^{i-1}\right) \approx R_b\left(T_b^{i,k-1}, T_o^{i-1}\right) + DR_b\left(T_b^{i,k-1}, T_o^{i-1}\right)[\delta T_b] \stackrel{!}{=} 0 & \text{(4.32a)} \\ R_o\left(T_b^{i}, T_o^{i,k}\right) \approx R_o\left(T_b^{i}, T_o^{i,k-1}\right) + DR_o\left(T_b^{i}, T_o^{i,k-1}\right)[\delta T_o] \stackrel{!}{=} 0, & \text{(4.32b)} \end{cases}$$

where $D(\cdot)$ is the directional derivative operator, δT_b and δT_o the solution increments, and k the internal Newton iteration counter. Under this notation, the full temperature field is updated for each Gauss-Seidel iteration as

$$T_b^i = T_b^{i,k-1} + \delta T_b \quad \text{(4.33a)}$$
$$T_o^i = T_o^{i,k-1} + \delta T_o. \quad \text{(4.33b)}$$

However, since a single Newton approximation is applied (instead of a full iteration loop) in the one-step variant, the indices from the system can be merged to obtain the following system of equations:

$$\begin{cases} R_b\left(T_b^{l}, T_o^{l-1}\right) \approx R_b\left(T_b^{l-1}, T_o^{l-1}\right) + DR_b\left(T_b^{l-1}, T_o^{l-1}\right)[\delta T_b] \stackrel{!}{=} 0 & \text{(4.34a)} \\ R_o\left(T_b^{l}, T_o^{l}\right) \approx R_o\left(T_b^{l}, T_o^{l-1}\right) + DR_o\left(T_b^{l}, T_o^{l-1}\right)[\delta T_o] \stackrel{!}{=} 0 & \text{(4.34b)} \end{cases}$$

for the l-th merged one-step Gauss-Seidel-Newton iteration.

Finally, using the fact that the residual is required to be zero at the solution to the system, Equations 4.34a and 4.34b can be reorganized as follows:

$$\begin{cases} -DR_b\left(T_b^{l-1}, T_o^{l-1}\right)[\delta T_b] = R_b\left(T_b^{l-1}, T_o^{l-1}\right) & \text{(4.35a)} \\ -DR_o\left(T_b^{l}, T_o^{l-1}\right)[\delta T_o] = R_o\left(T_b^{l}, T_o^{l-1}\right) & \text{(4.35b)} \end{cases}$$

where the residuals can now be expanded into their integral forms:

$$R_b\left(T_b^{l-1}, T_o^{l-1}\right) = \int_\Omega f(\mathbf{x}) v_b \, d\Omega - \int_\Omega \nabla v_b \, k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right) \nabla\left(T_b^{l-1} + T_o^{l-1}\right) \, d\Omega \quad \text{(4.36)}$$

and

$$R_o\left(T_b^{l}, T_o^{l-1}\right) = \int_\Omega f(\mathbf{x}) v_o \, d\Omega - \int_\Omega \nabla v_o \, k\left(\mathbf{x}, T_b^{l} + T_o^{l-1}\right) \nabla\left(T_b^{l} + T_o^{l-1}\right) \, d\Omega \quad \text{(4.37)}$$

after applying integration by parts and omitting boundary terms. Applying the product rule, the corresponding directional derivative in the individual equations are calculated to be:

$$
\begin{aligned}
DR_b\left(T_b^{l-1}, T_o^{l-1}\right)[\delta T_b] = & \int_\Omega \cancelto{0}{D\left(f(\mathbf{x})v_b\right)[\delta T_b]}\, d\Omega \\
& - \int_\Omega \cancelto{0}{D\left(\nabla v_b\right)[\delta T_b]}\, k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right) \nabla\left(T_b^{l-1} + T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla v_b\, D\left(k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right)\right)[\delta T_b] \nabla\left(T_b^{l-1} + T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla v_b\, k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right) D\left(\nabla\left(T_b^{l-1} + T_o^{l-1}\right)\right)[\delta T_b]\, d\Omega
\end{aligned}
\tag{4.38}
$$

for the base and, analogously

$$
\begin{aligned}
DR_o\left(T_b^{l}, T_o^{l-1}\right)[\delta T_o] = & \int_\Omega \cancelto{0}{D\left(f(\mathbf{x})v_o\right)[\delta T_o]}\, d\Omega \\
& - \int_\Omega \cancelto{0}{D\left(\nabla v_o\right)[\delta T_o]}\, k\left(\mathbf{x}, T_b^{l} + T_o^{l-1}\right) \nabla\left(T_b^{l} + T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla v_o\, D\left(k\left(\mathbf{x}, T_b^{l} + T_o^{l-1}\right)\right)[\delta T_o] \nabla\left(T_b^{l} + T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla v_o\, k\left(\mathbf{x}, T_b^{l} + T_o^{l-1}\right) D\left(\nabla\left(T_b^{l} + T_o^{l-1}\right)\right)[\delta T_o]\, d\Omega
\end{aligned}
\tag{4.39}
$$

for the overlay.

As for the derivatives of the individual terms, their Taylor expansions lead to:

$$D\left(k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right)\right)[\delta T_b] = \frac{d}{d\varepsilon}\left[k\left(\mathbf{x}, T_b^{l-1} + \varepsilon\delta T_b + T_o^{l-1}\right)\right]_{\varepsilon=0} \approx \frac{dk}{dT}\delta T_b \tag{4.40a}$$

$$D\left(\nabla\left(T_b^{l-1} + T_o^{l-1}\right)\right)[\delta T_b] = \frac{d}{d\varepsilon}\left[\nabla\left(T_b^{l-1} + \varepsilon\delta T_b + T_o^{l-1}\right)\right]_{\varepsilon=0} \approx \nabla\left(\delta T_b\right) \tag{4.40b}$$

$$D\left(k\left(\mathbf{x}, T_b^{l} + T_o^{l-1}\right)\right)[\delta T_o] = \frac{d}{d\varepsilon}\left[k\left(\mathbf{x}, T_b^{l} + T_o^{l-1} + \varepsilon\delta T_o\right)\right]_{\varepsilon=0} \approx \frac{dk}{dT}\delta T_o \tag{4.40c}$$

$$D\left(\nabla\left(T_b^{l} + T_o^{l-1}\right)\right)[\delta T_o] = \frac{d}{d\varepsilon}\left[\nabla\left(T_b^{l} + T_o^{l-1} + \varepsilon\delta T_o\right)\right]_{\varepsilon=0} \approx \nabla\left(\delta T_o\right). \tag{4.40d}$$

Putting these results back into Equations 4.35a and 4.35b, this leads to the system formed by:

$$
\begin{aligned}
\int_\Omega \nabla v_b\, \frac{dk}{dT}\delta T_b \nabla\left(T_b^{l-1} + T_o^{l-1}\right)\, d\Omega + \int_\Omega \nabla v_b\, k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right) \nabla\left(\delta T_b\right)\, d\Omega = \\
\int_\Omega f(\mathbf{x}) v_b\, d\Omega - \int_\Omega \nabla v_b\, k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right) \nabla\left(T_b^{l-1} + T_o^{l-1}\right)\, d\Omega
\end{aligned}
\tag{4.41}
$$

and

$$\int_\Omega \nabla v_o \, \frac{dk}{dT} \delta T_o \nabla \left(T_b^l + T_o^{l-1} \right) \, d\Omega + \int_\Omega \nabla v_o \, k \left(\mathbf{x}, T_b^l + T_o^{l-1} \right) \nabla \left(\delta T_o \right) \, d\Omega = \int_\Omega f(\mathbf{x}) v_o \, d\Omega - \int_\Omega \nabla v_o \, k \left(\mathbf{x}, T_b^l + T_o^{l-1} \right) \nabla \left(T_b^l + T_o^{l-1} \right) \, d\Omega. \quad (4.42)$$

After discretization, the system composed of Equations 4.41 and 4.42 can be ultimately expressed, in terms of the component matrices, as follows:

$$\begin{cases} \mathbf{J}_{bb} \left(T_b^{l-1}, T_o^{l-1} \right) \delta T_b = \mathbf{R}_b \left(T_b^{l-1}, T_o^{l-1} \right) & (4.43a) \\ \mathbf{J}_{oo} \left(T_b^{l}, T_o^{l-1} \right) \delta T_o = \mathbf{R}_o \left(T_b^{l}, T_o^{l-1} \right). & (4.43b) \end{cases}$$

where the definition of each of the individual terms can be found in Table 4.3. As an important difference with respect to the previous cases, since the problem is now nonlinear, each of the component matrices must be recomputed at each iteration.

Table 4.3: Multi-level hp finite element matrices for a nonlinear stationary heat equation resolved by the one-step Gauss-Seidel-Newton scheme

Term	Symbol	Definition
Base Jacobian Matrix	$\mathbf{J_{bb}} \left(T_b^{l-1}, T_o^{l-1} \right)$	$\left(\mathbf{K_{bb}} \left(T_b^{l-1}, T_o^{l-1} \right) + \mathbf{K'_{bb}} \left(T_b^{l-1}, T_o^{l-1} \right) \right)$
Overlay Jacobian Matrix	$\mathbf{J_{oo}} \left(T_b^{l}, T_o^{l-1} \right)$	$\left(\mathbf{K_{oo}} \left(T_b^{l}, T_o^{l-1} \right) + \mathbf{K'_{oo}} \left(T_b^{l}, T_o^{l-1} \right) \right)$
Base Residual Vector	$\mathbf{R_b} \left(T_b^{l-1}, T_o^{l-1} \right)$	$\mathbf{F_b} - \mathbf{K_{bb}} \left(T_b^{l-1}, T_o^{l-1} \right) T_b^{l-1} - \mathbf{K_{bo}} \left(T_b^{l-1}, T_o^{l-1} \right) T_o^{l-1}$
Overlay Residual Vector	$\mathbf{R_o} \left(T_b^{l}, T_o^{l-1} \right)$	$\mathbf{F_o} - \mathbf{K_{oo}} \left(T_b^{l}, T_o^{l-1} \right) T_o^{l-1} - \mathbf{K_{ob}} \left(T_b^{l}, T_o^{l-1} \right) T_b^{l}$
Base Stiffness Matrix	$\mathbf{K_{bb}} \left(T_b^{l-1}, T_o^{l-1} \right)$	$\int_{\Omega_b} \mathbf{B}_b^T \, k \left(\mathbf{x}, \mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1} \right) \mathbf{B}_b \, d\Omega$
Derived Base Stiffness Matrix	$\mathbf{K'_{bb}} \left(T_b^{l-1}, T_o^{l-1} \right)$	$\int_{\Omega_b} \mathbf{B}_b^T \left. \frac{dk}{dT} \right\vert_{\mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1}} \left(\mathbf{B}_b T_b^{l-1} + \mathbf{B}_o T_o^{l-1} \right) \mathbf{N}_b \, d\Omega$
Base Coupling Stiffness Matrix	$\mathbf{K_{bo}} \left(T_b^{l-1}, T_o^{l-1} \right)$	$\int_{\Omega_o} \mathbf{B}_b^T \, k \left(\mathbf{x}, \mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1} \right) \mathbf{B}_o \, d\Omega$
Overlay Stiffness Matrix	$\mathbf{K_{oo}} \left(T_b^{l}, T_o^{l-1} \right)$	$\int_{\Omega_o} \mathbf{B}_o^T \, k \left(\mathbf{x}, \mathbf{N}_b T_b^{l} + \mathbf{N}_o T_o^{l-1} \right) \mathbf{B}_o \, d\Omega$
Derived Overlay Stiffness Matrix	$\mathbf{K'_{oo}} \left(T_b^{l}, T_o^{l-1} \right)$	$\int_{\Omega_o} \mathbf{B}_o^T \left. \frac{dk}{dT} \right\vert_{\mathbf{N}_b T_b^{l} + \mathbf{N}_o T_o^{l-1}} \left(\mathbf{B}_b T_b^{l} + \mathbf{B}_o T_o^{l-1} \right) \mathbf{N}_o \, d\Omega$

Continued on next page

Table 4.3 – *Continued from previous page*

Term	Symbol	Definition
Overlay Coupling Stiffness Matrix	$\mathbf{K_{ob}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_o^T \; k\left(\mathbf{x}, \mathbf{N}_b T_b^l + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_b \; d\Omega$
Base Source Vector	$\mathbf{F_b}$	$\int_{\Omega_b} \mathbf{N}_b^T f(\mathbf{x}) \; d\Omega$
Overlay Source Vector	$\mathbf{F_o}$	$\int_{\Omega_o} \mathbf{N}_o^T f(\mathbf{x}) \; d\Omega$

4.3.2 Method Implementation

Algorithm 3 describes the implemented one-step nonlinear Gauss-Seidel-Newton scheme. In contrast to the previous cases, only the source terms can be precomputed outside of the iteration loop, which in this case is omitted for clarity (i.e. they are computed together with the system matrices).

Algorithm 3: Multi-level hp algorithm for nonlinear stationary problems

define overlay location
create global base mesh and overlay mesh in specified region
remove degrees of freedom to ensure linear independence
set Gauss-Seidel-Newton iteration counter $l \leftarrow 0$
while *not converged* **and** *below maximum iteration number* **do**
  Compute *base solution increment*
    compute stiffness matrices $\mathbf{K_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$ and $\mathbf{K'_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$, and source term $\mathbf{F_b}$
    compute coupling term $\mathbf{K_{bo}}\left(T_b^{l-1}, T_o^{l-1}\right) T_o^{l-1}$
    apply base system constraints (boundary conditions)
    solve Equation 4.43a
    $T_b^l \leftarrow T_b^{l-1} + \delta T_b$
  Compute *overlay solution increment*
    compute stiffness matrices $\mathbf{K_{oo}}\left(T_b^l, T_o^{l-1}\right)$ and $\mathbf{K'_{oo}}\left(T_b^l, T_o^{l-1}\right)$, and source term $\mathbf{F_o}$
    compute coupling term $\mathbf{K_{ob}}\left(T_b^l, T_o^{l-1}\right) T_b^l$
    apply overlay system constraints (homogeneous Dirichlet)
    solve Equation 4.43b
    $T_o^l \leftarrow T_o^{l-1} + \delta T_o$
  compute L^2 norm of the complete solution
  check convergence in L^2 norm
  $l \leftarrow l + 1$
postprocess total solution

4.4 Nonlinear transient heat equation

Introducing transient effects again, now the full form of Equation 2.8 is considered. The approaches to decouple the system variables and solve transient and nonlinear effects were introduced in earlier cases and will thus be applied here without further introduction. As before, the treatment of transient effects will follow the Backward Euler scheme, while the nonlinearities will be resolved via one-step Gauss-Seidel-Newton.

4.4.1 Partitioned problem formulation

As indicated previously, the problem analyzed corresponds to the full nonlinear transient heat equation, namely:

$$\left\langle \rho c(\mathbf{x}, T)\dot{T}, v \right\rangle - \mathscr{A}\left(T, k(\mathbf{x}, T), v\right) = \mathscr{F}\left(v\right) \quad \forall v \in \mathbf{V}, \tag{4.44}$$

which under the Backward Euler discretization can be expressed as

$$\left\langle \rho c(\mathbf{x}, T_{n-1} + \Delta T_n)\frac{\Delta T_n}{\Delta t}, v \right\rangle - \mathscr{A}\left(T_{n-1} + \Delta T_n, k(\mathbf{x}, T_{n-1} + \Delta T_n), v\right) = \mathscr{F}\left(v\right) \tag{4.45}$$

where n indicates the computed time step as before, and the update rule for the temperature is

$$T_n = T_{n-1} + \Delta T_n. \tag{4.46}$$

It is then possible to separate previous time step solution T_{n-1} from increment solution ΔT_n using the linearity of the gradient operator, which leads to:

$$\begin{aligned}\left\langle \rho c(\mathbf{x}, T_{n-1} + \Delta T_n)\frac{\Delta T_n}{\Delta t}, v \right\rangle - \mathscr{A}\left(T_{n-1}, k(\mathbf{x}, T_{n-1} + \Delta T_n), v\right) \\ - \mathscr{A}\left(\Delta T_n, k(\mathbf{x}, T_{n-1} + \Delta T_n), v\right) = \mathscr{F}\left(v\right).\end{aligned} \tag{4.47}$$

This expression can then be rewritten in residual form, in preparation for resolution of the nonlinearities, defining:

$$\begin{aligned}R_n\left(T_n\right) = \mathscr{F}\left(v\right) - \left\langle \rho c(\mathbf{x}, T_{n-1} + \Delta T_n)\frac{\Delta T_n}{\Delta t}, v \right\rangle + \mathscr{A}\left(T_{n-1}, k(\mathbf{x}, T_{n-1} + \Delta T_n), v\right) \\ + \mathscr{A}\left(\Delta T_n, k(\mathbf{x}, T_{n-1} + \Delta T_n), v\right)\end{aligned} \tag{4.48}$$

This can then be split into base and overlay problems under the multilevel concept to yield

the equation system:

$$\begin{aligned} R_{b,n}\left(T_{b,n}, T_{o,n}\right) =& \mathscr{F}\left(v_b\right) - \left\langle \rho c\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n} + T_{o,n-1} + \Delta T_{o,n}\right) \frac{\Delta T_{b,n} + \Delta T_{o,n}}{\Delta t}, v_b \right\rangle \\ &+ \mathscr{A}\left(T_{b,n-1} + T_{o,n-1}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n} + T_{o,n-1} + \Delta T_{o,n}\right), v_b\right) \\ &+ \mathscr{A}\left(\Delta T_{b,n} + \Delta T_{o,n}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n} + T_{o,n-1} + \Delta T_{o,n}\right), v_b\right) \stackrel{!}{=} 0 \end{aligned} \tag{4.49}$$

$$\begin{aligned} R_{o,n}\left(T_{b,n}, T_{o,n}\right) =& \mathscr{F}\left(v_o\right) - \left\langle \rho c\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n} + T_{o,n-1} + \Delta T_{o,n}\right) \frac{\Delta T_{b,n} + \Delta T_{o,n}}{\Delta t}, v_o \right\rangle \\ &+ \mathscr{A}\left(T_{b,n-1} + T_{o,n-1}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n} + T_{o,n-1} + \Delta T_{o,n}\right), v_o\right) \\ &+ \mathscr{A}\left(\Delta T_{b,n} + \Delta T_{o,n}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n} + T_{o,n-1} + \Delta T_{o,n}\right), v_o\right) \stackrel{!}{=} 0. \end{aligned} \tag{4.50}$$

Application of the nonlinear one-step Gauss-Seidel-Newton scheme requires now to apply the Gauss-Seidel approach to the residual equations, resulting in:

$$\begin{aligned} R_{b,n}\left(T_{b,n}^{i}, T_{o,n}^{i-1}\right) =& \mathscr{F}\left(v_b\right) - \left\langle \rho c\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{i} + T_{o,n-1} + \Delta T_{o,n}^{i-1}\right) \frac{\Delta T_{b,n}^{i} + \Delta T_{o,n}^{i-1}}{\Delta t}, v_b \right\rangle \\ &+ \mathscr{A}\left(T_{b,n-1} + T_{o,n-1}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{i} + T_{o,n-1} + \Delta T_{o,n}^{i-1}\right), v_b\right) \\ &+ \mathscr{A}\left(\Delta T_{b,n}^{i} + \Delta T_{o,n}^{i-1}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{i} + T_{o,n-1} + \Delta T_{o,n}^{i-1}\right), v_b\right) \stackrel{!}{=} 0 \end{aligned} \tag{4.51}$$

$$\begin{aligned} R_{o,n}\left(T_{b,n}^{i}, T_{o,n}^{i}\right) =& \mathscr{F}\left(v_o\right) - \left\langle \rho c\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{i} + T_{o,n-1} + \Delta T_{o,n}^{i}\right) \frac{\Delta T_{b,n}^{i} + \Delta T_{o,n}^{i}}{\Delta t}, v_o \right\rangle \\ &+ \mathscr{A}\left(T_{b,n-1} + T_{o,n-1}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{i} + T_{o,n-1} + \Delta T_{o,n}^{i}\right), v_o\right) \\ &+ \mathscr{A}\left(\Delta T_{b,n}^{i} + \Delta T_{o,n}^{i}, k\left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{i} + T_{o,n-1} + \Delta T_{o,n}^{i}\right), v_o\right) \stackrel{!}{=} 0. \end{aligned} \tag{4.52}$$

where i indicates the Gauss-Seidel iterations, so that the Newton method leads to:

$$\begin{cases} R_b\left(T_b^l, T_o^{l-1}\right) \approx R_b\left(T_b^{l-1}, T_o^{l-1}\right) + DR_b\left(T_b^{l-1}, T_o^{l-1}\right)\left[\delta T_b\right] \stackrel{!}{=} 0 & \text{(4.53a)} \\ R_o\left(T_b^l, T_o^l\right) \approx R_o\left(T_b^l, T_o^{l-1}\right) + DR_o\left(T_b^l, T_o^{l-1}\right)\left[\delta T_o\right] \stackrel{!}{=} 0 & \text{(4.53b)} \end{cases}$$

when using l as the merged Gauss-Seidel-Newton iteration index. That is, the update for the

step increment is now:

$$\Delta T_{b,n}^{l} = \Delta T_{b,n}^{l-1} + \delta T_b \tag{4.54a}$$
$$\Delta T_{o,n}^{l} = \Delta T_{o,n}^{l-1} + \delta T_o, \tag{4.54b}$$

or for the full temperature field:

$$T_{b,n}^{l} = T_{b,n}^{l-1} + \delta T_b = T_{b,n-1} + \Delta T_{b,n}^{l-1} + \delta T_b \tag{4.55a}$$
$$T_{o,n}^{l} = T_{o,n}^{l-1} + \delta T_o = T_{o,n-1} + \Delta T_{o,n}^{l-1} + \delta T_o. \tag{4.55b}$$

Using the fact that the residual at a solution is required to be zero, Equations 4.53a and 4.53b can then be expressed as follows:

$$\begin{cases} -DR_{b,n}\left(T_{b,n}^{l-1}, T_{o,n}^{l-1}\right)[\delta T_b] = R_{b,n}\left(T_{b,n}^{l-1}, T_{o,n}^{l-1}\right) & (4.56a) \\ -DR_{o,n}\left(T_{b,n}^{l}, T_{o,n}^{l-1}\right)[\delta T_o] = R_{o,n}\left(T_{b,n}^{l}, T_{o,n}^{l-1}\right). & (4.56b) \end{cases}$$

Here, the individual residuals can be expressed in the integral form, after integration by parts and omission of boundary terms, to obtain:

$$\begin{aligned} R_{b,n}\left(T_{b,n}^{l-1}, T_{o,n}^{l-1}\right) = & \int_\Omega f(\mathbf{x}, t_n) v_b \, d\Omega \\ & - \int_\Omega v_b \, \rho c \left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{l-1} + T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) \frac{\Delta T_{b,n}^{l-1} + \Delta T_{o,n}^{l-1}}{\Delta t} \, d\Omega \\ & - \int_\Omega \nabla v_b \, k \left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{l-1} + T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) \nabla \left(T_{b,n-1} + T_{o,n-1}\right) \, d\Omega \\ & - \int_\Omega \nabla v_b \, k \left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{l-1} + T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) \nabla \left(\Delta T_{b,n}^{l-1} + \Delta T_{o,n}^{l-1}\right) \, d\Omega \end{aligned} \tag{4.57}$$

$$\begin{aligned} R_{o,n}\left(T_{b,n}^{l}, T_{o,n}^{l-1}\right) = & \int_\Omega f(\mathbf{x}, t_n) v_o \, d\Omega \\ & - \int_\Omega v_o \, \rho c \left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{l} + T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) \frac{\Delta T_{b,n}^{l} + \Delta T_{o,n}^{l-1}}{\Delta t} \, d\Omega \\ & - \int_\Omega \nabla v_o \, k \left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{l} + T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) \nabla \left(T_{b,n-1} + T_{o,n-1}\right) \, d\Omega \\ & - \int_\Omega \nabla v_o \, k \left(\mathbf{x}, T_{b,n-1} + \Delta T_{b,n}^{l} + T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) \nabla \left(\Delta T_{b,n}^{l} + \Delta T_{o,n}^{l-1}\right) \, d\Omega. \end{aligned} \tag{4.58}$$

Applying the product rule of the directional derivative, the Gateaux derivatives can then calculated to be:

$$
\begin{aligned}
DR_{b,n}\left(T_{b,n}^{l-1},T_{o,n}^{l-1}\right)[\delta T_b] = {} & \cancelto{0}{\int_\Omega D\left(f(\mathbf{x},t_n)v_b\right)[\delta T_b]\,d\Omega} \\
& - \cancelto{0}{\int_\Omega D\left(v_b\right)[\delta T_b]\,\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\frac{\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}}{\Delta t}\,d\Omega} \\
& - \int_\Omega v_b\, D\left(\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_b]\,\frac{\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}}{\Delta t}\,d\Omega \\
& - \int_\Omega v_b\,\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right) D\left(\frac{\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}}{\Delta t}\right)[\delta T_b]\;d\Omega \\
& - \cancelto{0}{\int_\Omega D\left(\nabla v_b\right)[\delta T_b]\,k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(T_{b,n-1}+T_{o,n-1}\right)\,d\Omega} \\
& - \int_\Omega \nabla v_b\, D\left(k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_b]\,\nabla\left(T_{b,n-1}+T_{o,n-1}\right)\,d\Omega \\
& - \int_\Omega \nabla v_b\, k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right) D\left(\nabla\left(\cancel{T_{b,n-1}+T_{o,n-1}}\right)\right)[\delta T_b]\,d\Omega \\
& - \cancelto{0}{\int_\Omega D\left(\nabla v_b\right)[\delta T_b]\,k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}\right)\,d\Omega} \\
& - \int_\Omega \nabla v_b\, D\left(k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_b]\,\nabla\left(\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}\right)\,d\Omega \\
& - \int_\Omega \nabla v_b\, k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right) D\left(\nabla\left(\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_b]\;d\Omega
\end{aligned}
\tag{4.59}
$$

where the individual term derivatives are

$$
\begin{aligned}
& D\left(\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_b] \\
& \qquad = \frac{d}{d\varepsilon}\left[\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+\varepsilon\delta T_b+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right]_{\varepsilon=0} \\
& \qquad \approx \left(\frac{d\rho}{dT}c+\rho\frac{dc}{dT}\right)\Bigg|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}}\delta T_b
\end{aligned}
\tag{4.60}
$$

$$
\begin{aligned}
& D\left(k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_b] \\
& \qquad = \frac{d}{d\varepsilon}\left[k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+\varepsilon\delta T_b+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right]_{\varepsilon=0} \\
& \qquad \approx \frac{dk}{dT}\Bigg|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}}\delta T_b
\end{aligned}
\tag{4.61}
$$

$$D\left(\frac{\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}}{\Delta t}\right)[\delta T_b]=\frac{d}{d\varepsilon}\left[\frac{\Delta T_{b,n}^{l-1}+\varepsilon\delta T_b+\Delta T_{o,n}^{l-1}}{\Delta t}\right]_{\varepsilon=0}\approx\frac{\delta T_b}{\Delta t} \tag{4.62}$$

$$D\left(\nabla\left(\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_b]=\frac{d}{d\varepsilon}\left[\nabla\left(\Delta T_{b,n}^{l-1}+\varepsilon\delta T_b+\Delta T_{o,n}^{l-1}\right)\right]_{\varepsilon=0}\approx\nabla\left(\delta T_b\right), \tag{4.63}$$

and analogously for the overlay:

$$\begin{aligned}
DR_{o,n}\left(T_{b,n}^{l},T_{o,n}^{l-1}\right)[\delta T_o]=&\int_\Omega \cancelto{0}{D\left(f(\mathbf{x},t_n)v_o\right)[\delta T_o]}\,d\Omega\\
&-\int_\Omega \cancelto{0}{D\left(v_o\right)[\delta T_o]}\,\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\frac{\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}}{\Delta t}\,d\Omega\\
&-\int_\Omega v_o\,D\left(\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_o]\,\frac{\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}}{\Delta t}\,d\Omega\\
&-\int_\Omega v_o\,\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)D\left(\frac{\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}}{\Delta t}\right)[\delta T_o]\,\,d\Omega\\
&-\int_\Omega \cancelto{0}{D\left(\nabla v_o\right)[\delta T_o]}\,k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(T_{b,n-1}+T_{o,n-1}\right)\,d\Omega\\
&-\int_\Omega \nabla v_o\,D\left(k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_o]\,\nabla\left(T_{b,n-1}+T_{o,n-1}\right)\,d\Omega\\
&-\int_\Omega \nabla v_o\,k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)D\left(\nabla\left(T_{b,n-1}+T_{o,n-1}\right)\right)[\delta T_o]\,d\Omega\\
&-\int_\Omega \cancelto{0}{D\left(\nabla v_o\right)[\delta T_o]}\,k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}\right)\,d\Omega\\
&-\int_\Omega \nabla v_o\,D\left(k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_o]\,\nabla\left(\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}\right)\,d\Omega\\
&-\int_\Omega \nabla v_o\,k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)D\left(\nabla\left(\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_o]\,\,d\Omega.
\end{aligned} \tag{4.64}$$

where

$$\begin{aligned}
&D\left(\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_o]\\
&\qquad=\frac{d}{d\varepsilon}\left[\rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}+\varepsilon\delta T_o\right)\right]_{\varepsilon=0}\\
&\qquad\approx\left(\frac{d\rho}{dT}c+\rho\frac{dc}{dT}\right)\Big|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}}\delta T_o
\end{aligned} \tag{4.65}$$

$$
\begin{aligned}
D\left(k\left(\mathbf{x}, T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_o] \\
&= \frac{d}{d\varepsilon}\left[k\left(\mathbf{x}, T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}+\varepsilon\delta T_o\right)\right]_{\varepsilon=0} \\
&\approx \left.\frac{dk}{dT}\right|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}} \delta T_o
\end{aligned}
\tag{4.66}
$$

$$
D\left(\frac{\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}}{\Delta t}\right)[\delta T_b] = \frac{d}{d\varepsilon}\left[\frac{\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}+\varepsilon\delta T_o}{\Delta t}\right]_{\varepsilon=0} \approx \frac{\delta T_o}{\Delta t} \tag{4.67}
$$

$$
D\left(\nabla\left(\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}\right)\right)[\delta T_o] = \frac{d}{d\varepsilon}\left[\nabla\left(\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}+\varepsilon\delta T_o\right)\right]_{\varepsilon=0} \approx \nabla\left(\delta T_o\right). \tag{4.68}
$$

These results can then be inserted back into Equations 4.56a and 4.56b to obtain

$$
\begin{aligned}
&\int_\Omega v_b \left(\frac{d\rho}{dT}c+\rho\frac{dc}{dT}\right)\Big|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}} \delta T_b \frac{\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}}{\Delta t}\, d\Omega \\
&\quad+\int_\Omega v_b\, \rho c\left(\mathbf{x}, T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\frac{\delta T_b}{\Delta t}\, d\Omega \\
&\quad+\int_\Omega \nabla v_b \left.\frac{dk}{dT}\right|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}} \delta T_b\, \nabla\left(T_{b,n-1}+T_{o,n-1}\right)\, d\Omega \\
&\quad+\int_\Omega \nabla v_b \left.\frac{dk}{dT}\right|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}} \delta T_b\, \nabla\left(\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}\right)\, d\Omega \\
&\quad+\int_\Omega \nabla v_b\, k\left(\mathbf{x}, T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(\delta T_b\right)\, d\Omega \\
&\qquad=\int_\Omega f(\mathbf{x},t_n) v_b\, d\Omega \\
&\qquad\quad-\int_\Omega v_b\, \rho c\left(\mathbf{x}, T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\frac{\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}}{\Delta t}\, d\Omega \\
&\qquad\quad-\int_\Omega \nabla v_b\, k\left(\mathbf{x}, T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(T_{b,n-1}+T_{o,n-1}\right)\, d\Omega \\
&\qquad\quad-\int_\Omega \nabla v_b\, k\left(\mathbf{x}, T_{b,n-1}+\Delta T_{b,n}^{l-1}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(\Delta T_{b,n}^{l-1}+\Delta T_{o,n}^{l-1}\right)\, d\Omega
\end{aligned}
\tag{4.69}
$$

$$
\begin{aligned}
&\int_\Omega v_o \left(\frac{d\rho}{dT}c+\rho\frac{dc}{dT}\right)\Big|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}} \delta T_o \frac{\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}}{\Delta t}\, d\Omega \\
&\quad+\int_\Omega v_o\, \rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\frac{\delta T_o}{\Delta t}\, d\Omega \\
&\quad+\int_\Omega \nabla v_o \left.\frac{dk}{dT}\right|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}} \delta T_o\, \nabla\left(T_{b,n-1}+T_{o,n-1}\right)\, d\Omega \\
&\quad+\int_\Omega \nabla v_o \left.\frac{dk}{dT}\right|_{\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}} \delta T_o\, \nabla\left(\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}\right)\, d\Omega \\
&\quad+\int_\Omega \nabla v_o\, k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(\delta T_o\right)\, d\Omega \\
&\qquad=\int_\Omega f(\mathbf{x},t_n) v_o\, d\Omega \\
&\qquad\quad-\int_\Omega v_o\, \rho c\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\frac{\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}}{\Delta t}\, d\Omega \\
&\qquad\quad-\int_\Omega \nabla v_o\, k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(T_{b,n-1}+T_{o,n-1}\right)\, d\Omega \\
&\qquad\quad-\int_\Omega \nabla v_o\, k\left(\mathbf{x},T_{b,n-1}+\Delta T_{b,n}^{l}+T_{o,n-1}+\Delta T_{o,n}^{l-1}\right)\nabla\left(\Delta T_{b,n}^{l}+\Delta T_{o,n}^{l-1}\right)\, d\Omega.
\end{aligned}
\tag{4.70}
$$

This system, although involving multiple components, can ultimately be rewritten, after discretization, as follows:

$$
\begin{cases}
\mathbf{J}_{bb,n}\left(T_b^{l-1},T_o^{l-1}\right)\delta T_b=\mathbf{R}_{b,n}\left(T_b^{l-1},T_o^{l-1}\right) & \text{(4.71a)}\\
\mathbf{J}_{oo,n}\left(T_b^{l},T_o^{l-1}\right)\delta T_o=\mathbf{R}_{o,n}\left(T_b^{l},T_o^{l-1}\right). & \text{(4.71b)}
\end{cases}
$$

where each of terms used are defined in Table 4.4. In this case, all terms are consistently being updated - after every iteration or every time step, throughout the computation.

Table 4.4: Multi-level hp finite element matrices for a nonlinear transient heat equation resolved by the one-step Gauss-Seidel-Newton scheme with Backward Euler time discretization

Term	Symbol	Definition
Base Jacobian Matrix	$\mathbf{J_{bb,n}}\left(T_b^{l-1},T_o^{l-1}\right)$	$\frac{1}{\Delta t}\left(\mathbf{M_{bb,n}}\left(T_b^{l-1},T_o^{l-1}\right)+\mathbf{M'_{bb,n}}\left(T_b^{l-1},T_o^{l-1}\right)\right)$ $+\left(\mathbf{K_{bb,n}}\left(T_b^{l-1},T_o^{l-1}\right)+\mathbf{K'_{bb,n}}\left(T_b^{l-1},T_o^{l-1}\right)\right)$
Overlay Jacobian Matrix	$\mathbf{J_{oo,n}}\left(T_b^{l},T_o^{l-1}\right)$	$\frac{1}{\Delta t}\left(\mathbf{M_{oo,n}}\left(T_b^{l},T_o^{l-1}\right)+\mathbf{M'_{oo,n}}\left(T_b^{l},T_o^{l-1}\right)\right)$ $+\left(\mathbf{K_{oo,n}}\left(T_b^{l},T_o^{l-1}\right)+\mathbf{K'_{oo,n}}\left(T_b^{l},T_o^{l-1}\right)\right)$

Continued on next page

Table 4.4 – *Continued from previous page*

Term	**Symbol**	**Definition**
Base Residual Vector	$\mathbf{R_b}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\mathbf{F_{b,n}} - \mathbf{K_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)\left(T_{b,n-1} + \Delta T_{b,n}^{l-1}\right) - \mathbf{K_{bo,n}}\left(T_b^{l-1}, T_o^{l-1}\right)\left(T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) - \frac{1}{\Delta t}\left(\mathbf{M_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)\Delta T_{b,n}^{l-1} + \mathbf{M_{bo,n}}\left(T_b^{l-1}, T_o^{l-1}\right)\Delta T_{o,n}^{l-1}\right)$
Overlay Residual Vector	$\mathbf{R_o}\left(T_b^{l}, T_o^{l-1}\right)$	$\mathbf{F_{o,n}} - \mathbf{K_{oo,n}}\left(T_b^{l}, T_o^{l-1}\right)\left(T_{o,n-1} + \Delta T_{o,n}^{l-1}\right) - \mathbf{K_{ob,n}}\left(T_b^{l}, T_o^{l-1}\right)\left(T_{b,n-1} + \Delta T_{b,n}^{l}\right) - \frac{1}{\Delta t}\left(\mathbf{M_{oo,n}}\left(T_b^{l}, T_o^{l-1}\right)\Delta T_{o,n}^{l-1} + \mathbf{M_{ob,n}}\left(T_b^{l}, T_o^{l-1}\right)\Delta T_{b,n}^{l}\right)$
Base Stiffness Matrix	$\mathbf{K_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{B}_b^T\, k\left(\mathbf{x}, \mathbf{N}_b T_{b,n}^{l-1} + \mathbf{N}_o T_{o,n}^{l-1}\right)\mathbf{B}_b\, d\Omega$
Derived Base Stiffness Matrix	$\mathbf{K'_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{B}_b^T \left.\frac{dk}{dT}\right\vert_{\mathbf{N}_b T_{b,n}^{l-1} + \mathbf{N}_o T_{o,n}^{l-1}} \left(\mathbf{B}_b T_{b,n}^{l-1} + \mathbf{B}_o T_{o,n}^{l-1}\right)\mathbf{N}_b\, d\Omega$
Base Coupling Stiffness Matrix	$\mathbf{K_{bo,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_b^T\, k\left(\mathbf{x}, \mathbf{N}_b T_{b,n}^{l-1} + \mathbf{N}_o T_{o,n}^{l-1}\right)\mathbf{B}_o\, d\Omega$
Overlay Stiffness Matrix	$\mathbf{K_{oo,n}}\left(T_b^{l}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_o^T\, k\left(\mathbf{x}, \mathbf{N}_b T_{b,n}^{l} + \mathbf{N}_o T_{o,n}^{l-1}\right)\mathbf{B}_o\, d\Omega$
Derived Overlay Stiffness Matrix	$\mathbf{K'_{oo,n}}\left(T_b^{l}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_o^T \left.\frac{dk}{dT}\right\vert_{\mathbf{N}_b T_{b,n}^{l} + \mathbf{N}_o T_{o,n}^{l-1}} \left(\mathbf{B}_b T_{b,n}^{l} + \mathbf{B}_o T_{o,n}^{l-1}\right)\mathbf{N}_o\, d\Omega$
Overlay Coupling Stiffness Matrix	$\mathbf{K_{ob,n}}\left(T_b^{l}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_o^T\, k\left(\mathbf{x}, \mathbf{N}_b T_{b,n}^{l} + \mathbf{N}_o T_{o,n}^{l-1}\right)\mathbf{B}_b\, d\Omega$
Base Mass Matrix	$\mathbf{M_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{N}_b^T\, \rho c\left(\mathbf{x}, \mathbf{N}_b T_{b,n}^{l-1} + \mathbf{N}_o T_{o,n}^{l-1}\right)\mathbf{N}_b\, d\Omega$
Derived Base Mass Matrix	$\mathbf{M'_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{N}_b^T \left.\left(\frac{d\rho}{dT}c + \rho\frac{dc}{dT}\right)\right\vert_{\mathbf{N}_b T_{b,n}^{l-1} + \mathbf{N}_o T_{o,n}^{l-1}} \left(\mathbf{N}_b \Delta T_{b,n}^{l-1} + \mathbf{N}_o \Delta T_{o,n}^{l-1}\right)\mathbf{N}_b\, d\Omega$
Base Coupling Mass Matrix	$\mathbf{M_{bo,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_b^T\, \rho c\left(\mathbf{x}, \mathbf{N}_b T_{b,n}^{l-1} + \mathbf{N}_o T_{o,n}^{l-1}\right)\mathbf{N}_o\, d\Omega$
Overlay Mass Matrix	$\mathbf{M_{oo,n}}\left(T_b^{l}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_o^T\, \rho c\left(\mathbf{x}, \mathbf{N}_b T_{b,n}^{l} + \mathbf{N}_o T_{o,n}^{l-1}\right)\mathbf{N}_o\, d\Omega$
Derived Overlay Mass Matrix	$\mathbf{M'_{oo,n}}\left(T_b^{l}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_o^T \left.\left(\frac{d\rho}{dT}c + \rho\frac{dc}{dT}\right)\right\vert_{\mathbf{N}_b T_{b,n}^{l} + \mathbf{N}_o T_{o,n}^{l-1}} \left(\mathbf{N}_b \Delta T_{b,n}^{l} + \mathbf{N}_o \Delta T_{o,n}^{l-1}\right)\mathbf{N}_o\, d\Omega$

Continued on next page

Table 4.4 – *Continued from previous page*

Term	Symbol	Definition
Overlay Coupling Mass Matrix	$\mathbf{M_{ob,n}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_o^T \, \rho c \left(\mathbf{x}, \mathbf{N}_b T_{b,n}^l + \mathbf{N}_o T_{o,n}^{l-1}\right) \mathbf{N}_b \, d\Omega$
Base Source Vector	$\mathbf{F_{b,n}}$	$\int_{\Omega_b} \mathbf{N}_b^T f(\mathbf{x}, t_n) \, d\Omega$
Overlay Source Vector	$\mathbf{F_{o,n}}$	$\int_{\Omega_o} \mathbf{N}_o^T f(\mathbf{x}, t_n) \, d\Omega$

4.4.2 Method Implementation

As in the previous cases, Algorithm 4 describes the implemented scheme, namely, the one-step nonlinear Gauss-Seidel-Newton method.

Algorithm 4: Multi-level hp algorithm for nonlinear transient problems

define overlay location
create global base mesh and overlay mesh in specified region
remove degrees of freedom to ensure linear independence
set initial conditions $T_{b,0}$ and $T_{o,0}$
$currentTime \leftarrow t_0$
for $n \leftarrow 1$ **to** *numberOfTimeSteps* **do**
  compute source terms $\mathbf{F_{b,n}}$ and $\mathbf{F_{o,n}}$
  $\Delta T_b \leftarrow 0, \Delta T_o \leftarrow 0$
  $currentTime \leftarrow currentTime + \Delta t$
  set Gauss-Seidel-Newton iteration counter $l \leftarrow 0$
  while *not converged* **and** *below maximum iteration number* **do**
    Compute *base solution increment*
      compute stiffness matrices $\mathbf{K_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$ and $\mathbf{K'_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$, and mass matrices $\mathbf{M_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$ and $\mathbf{M'_{bb,n}}\left(T_b^{l-1}, T_o^{l-1}\right)$
      compute coupling terms $\mathbf{K_{bo,n}}\left(T_b^{l-1}, T_o^{l-1}\right)\left(T_{o,n-1} + \Delta T_{o,n}^{l-1}\right)$ and $\mathbf{M_{bo,n}}\left(T_b^{l-1}, T_o^{l-1}\right)\left(\Delta T_{o,n}^{l-1}\right)$
      apply base system constraints (boundary conditions)
      solve Equation 4.71a
      $\Delta T_{b,n}^l \leftarrow T_b^{l-1} + \delta T_b$
    Compute *overlay solution increment*
      compute stiffness matrices $\mathbf{K_{oo,n}}\left(T_b^l, T_o^{l-1}\right)$ and $\mathbf{K'_{oo,n}}\left(T_b^l, T_o^{l-1}\right)$, and mass matrices $\mathbf{M_{oo,n}}\left(T_b^l, T_o^{l-1}\right)$ and $\mathbf{M'_{oo,n}}\left(T_b^l, T_o^{l-1}\right)$
      compute coupling terms $\mathbf{K_{ob}}\left(T_b^l, T_o^{l-1}\right)\left(T_{b,n-1} + \Delta T_{b,n}^l\right)$ and $\mathbf{M_{ob}}\left(T_b^l, T_o^{l-1}\right)\left(\Delta T_{b,n}^l\right)$
      apply overlay system constraints (homogeneous Dirichlet)
      solve Equation 4.71b
      $\Delta T_{o,n}^l \leftarrow T_o^{l-1} + \delta T_o$
    compute L^2 norm of the complete solution
    check convergence in L^2 norm
    $l \leftarrow l + 1$
  $T_{b,n} \leftarrow T_{b,n-1} + \Delta T_b^l$
  $T_{o,n} \leftarrow T_{o,n-1} + \Delta T_o^l$
postprocess total solution

4.5 Linear Space-Time method

After describing how the transient heat equation can be solved by approximating the time derivative with conventional time-stepping techniques such as Backward Euler, now the Space-Time FEM is analyzed. Again, as a new component is introduced, nonlinear effects are first omitted to highlight the differences between the two methods. This approach leads to a formulation which is structurally similar to that from Section 4.1, but with time-dependent terms also included.

4.5.1 Partitioned problem formulation

When nonlinearities are ignored, the equation to solve is the same as in Section 4.2. In this case, however, time-dependence is not resolved by discretizing the derivative into separate timesteps via Equations 4.17a and 4.17b, what yielded solutions

$$T_n = T\left(\mathbf{x}, t_n\right) = T\left(\mathbf{x}, t_0 + n\Delta t\right), \tag{4.72}$$

but is instead treated as described in Section 2.4.2. That is, the time variable is given a finite element discretization which can then be combined with the former spatial discretization via a Cartesian product. The auxiliary definitions $\nabla_{\mathbf{x}}$ and ∇_t defined in Section 2.4.2 are again used, such that:

$$\nabla\left(T\left(\mathbf{x}, t\right)\right) = \left(\nabla_{\mathbf{x}} T, \nabla_t T\right) = \left(\frac{\partial T}{\partial x_1}, \cdots, \frac{\partial T}{\partial x_d}, \frac{\partial T}{\partial t}\right) \tag{4.73}$$

where d is the number of space dimensions. Under this notation, the equation to solve can now be written in the weak form:

$$\langle \rho c(\mathbf{x}) \nabla_t T, v \rangle + \mathscr{B}\left(T, v\right) = \mathscr{F}\left(v\right) \quad \forall v \in \mathbf{V}, \tag{4.74}$$

with

$$\mathscr{B}\left(T, v\right) = \int_{\Omega} \nabla_{\mathbf{x}} v \; k\left(\mathbf{x}\right) \nabla_{\mathbf{x}} T \; d\Omega \tag{4.75}$$

after applying integration by parts and omitting boundary terms. As before, linear independence and linearity of the parameters and forms let the problem be split into base and overlay components:

$$\begin{cases} \langle \rho c(\mathbf{x}) \nabla_t T_b, v_b \rangle + \langle \rho c(\mathbf{x}) \nabla_t T_o, v_b \rangle + \mathscr{B}\left(T_b, v_b\right) + \mathscr{B}\left(T_o, v_b\right) = \mathscr{F}\left(v_b\right) & \forall v_b \in \mathbf{V}_b \quad (4.76a) \\ \langle \rho c(\mathbf{x}) \nabla_t T_b, v_o \rangle + \langle \rho c(\mathbf{x}) \nabla_t T_o, v_o \rangle + \mathscr{B}\left(T_b, v_o\right) + \mathscr{B}\left(T_o, v_o\right) = \mathscr{F}\left(v_o\right) & \forall v_o \in \mathbf{V}_o. \quad (4.76b) \end{cases}$$

The matrix expression for the system after discretization is then

$$\begin{bmatrix} \mathbf{M}_{bb} + \mathbf{K}_{bb} & \mathbf{M}_{bo} + \mathbf{K}_{bo} \\ \mathbf{M}_{ob} + \mathbf{K}_{ob} & \mathbf{M}_{oo} + \mathbf{K}_{oo} \end{bmatrix} \begin{bmatrix} T_b \\ T_o \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \mathbf{F}_o \end{bmatrix} \tag{4.77}$$

where the individual Mass and Stiffness matrices can be described by

$$\mathbf{M}_{\alpha\beta} = \int_\Omega \mathbf{N}_\alpha^T \rho c(\mathbf{x}) \mathbf{B}_{t,\beta} \, d\Omega \tag{4.78a}$$

$$\mathbf{B}_{\alpha\beta} = \int_\Omega \mathbf{B}_{\mathbf{x},\alpha}^T \kappa(\mathbf{x}) \mathbf{B}_{\mathbf{x},\beta} \, d\Omega \tag{4.78b}$$

for α, β indicating b (base) or o (overlay) components, and with $\mathbf{B}_\mathbf{x}$ and $\mathbf{B}_t$ being the discretized derivatives of the shape functions N. The problem can then be solved iteratively via the Gauss-Seidel procedure such that

$$\begin{bmatrix} \mathbf{M}_{bb} + \mathbf{K}_{bb} & 0 \\ \mathbf{M}_{ob} + \mathbf{K}_{ob} & \mathbf{M}_{oo} + \mathbf{K}_{oo} \end{bmatrix} \begin{bmatrix} T_b^i \\ T_o^i \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b - (\mathbf{M}_{bo} + \mathbf{K}_{bo}) \, T_o^{i-1} \\ \mathbf{F}_o \end{bmatrix} \tag{4.79}$$

so that the final system to be solved for is

$$\begin{cases} \mathbf{J}_{bb} T_b = \mathbf{R}_b(T_o^{i-1}) & \text{(4.80a)} \\ \mathbf{J}_{oo} T_o = \mathbf{R}_o(T_b^i). & \text{(4.80b)} \end{cases}$$

All of the terms involved are described in Table 4.5.

Table 4.5: Multi-level hp finite element matrices for a linear transient heat equation resolved by the block Gauss-Seidel scheme with a space-time discretization

Term	**Symbol**	**Definition**
Base System Matrix	$\mathbf{J_{bb}}$	$\mathbf{M_{oo}} + \mathbf{K_{oo}}$
Overlay System Matrix	$\mathbf{J_{oo}}$	$\mathbf{M_{oo}} + \mathbf{K_{oo}}$
Base Right-hand-side Vector	$\mathbf{R_b}(T_o^{i-1})$	$\mathbf{F_b} - (\mathbf{M_{bo}} + \mathbf{K_{bo}}) \, T_o^{i-1}$
Overlay Right-hand-side Vector	$\mathbf{R_{o,n}}(T_b^i)$	$\mathbf{F_o} - (\mathbf{M_{ob}} + \mathbf{K_{ob}}) \, T_b^i$
Base/Overlay Stiffness Matrix†	$\mathbf{K}_{\alpha\beta}$	$\int_\Omega \mathbf{B}_{\mathbf{x},\alpha}^T k(\mathbf{x}) \mathbf{B}_{\mathbf{x},\beta} \, d\Omega$
Base/Overlay Mass Matrix†	$\mathbf{M}_{\alpha\beta}$	$\int_\Omega \mathbf{N}_\alpha^T \rho c(\mathbf{x}) \mathbf{B}_{t,\beta} \, d\Omega$
Base Source Vector	$\mathbf{F_b}$	$\int_{\Omega_b} \mathbf{N}_b^T f(\mathbf{x}) \, d\Omega$
Overlay Source Vector	$\mathbf{F_o}$	$\int_{\Omega_o} \mathbf{N}_o^T f(\mathbf{x}) \, d\Omega$

† Here the subindices α and β correspond to either **b** for base or **o** for overlay. The resulting term is the corresponding system or coupling matrix.

4.5.2 Method Implementation

Algorithm 5 describes the solution process implemented for the problem. In this case, given the differences in treatment of the matrices $\mathbf{M}$ and $\mathbf{K}$, the coupling terms warrant to be reviewed again. In this case, multiplying by the corresponding Temperature fields leads to:

$$\begin{cases} (\mathbf{M_{bo}} + \mathbf{K_{bo}})\, T_o^{i-1} = \displaystyle\int_{\Omega_o} \left(\mathbf{N}_b^T \rho c(\mathbf{x}) \mathbf{B}_{t,o} + \mathbf{B}_{\mathbf{x},b}^T k(\mathbf{x}) \mathbf{B}_{\mathbf{x},o}\right)\, d\Omega\, T_o^{i-1} \\ \qquad\qquad = \displaystyle\int_{\Omega_o} \left(\mathbf{N}_b^T \rho c(\mathbf{x}) \delta_o^{i-1} + \mathbf{B}_{\mathbf{x},b}^T k(\mathbf{x}) \varepsilon_o^{i-1}\right)\, d\Omega \qquad (4.81a) \\ (\mathbf{M_{ob}} + \mathbf{K_{ob}})\, T_b^{i} = \displaystyle\int_{\Omega_o} \left(\mathbf{N}_o^T \rho c(\mathbf{x}) \mathbf{B}_{t,b} + \mathbf{B}_{\mathbf{x},o}^T k(\mathbf{x}) \mathbf{B}_{\mathbf{x},b}\right)\, d\Omega\, T_b^{i} \\ \qquad\qquad = \displaystyle\int_{\Omega_o} \left(\mathbf{N}_o^T \rho c(\mathbf{x}) \delta_b^{i} + \mathbf{B}_{\mathbf{x},o}^T k(\mathbf{x}) \varepsilon_b^{i}\right)\, d\Omega \qquad (4.81b) \end{cases}$$

where ε and δ represent, as before, the solution strain and increment respectively. That is, they represent the space and time derivatives respectively.

Algorithm 5: Multi-level hp algorithm for linear Space-Time problems

define overlay location in $\Omega \times [0, T]$
create global base mesh and overlay mesh in specified region
remove degrees of freedom to ensure linear independence
compute system matrices $\mathbf{J_{bb}} = \mathbf{M_{bb}} + \mathbf{K_{bb}}$ and $\mathbf{J_{oo}} = \mathbf{M_{oo}} + \mathbf{K_{oo}}$, and source terms $\mathbf{F_b}$ and $\mathbf{F_o}$
set Gauss-Seidel iteration counter $i \leftarrow 0$
while *not converged* **and** *below maximum iteration number* **do**
    Compute *base problem*
        compute coupling term $\mathbf{F}_{bo}^{i-1} = (\mathbf{M_{bo}} + \mathbf{K_{bo}})\, T_o^{i-1}$
        apply base system constraints (boundary and initial conditions)
        solve Equation 4.80a
    Compute *overlay problem*
        compute coupling term $\mathbf{F}_{ob}^{i} = (\mathbf{M_{ob}} + \mathbf{K_{ob}})\, T_b^{i}$
        apply overlay system constraints (homogeneous Dirichlet)
        solve Equation 4.80b
    compute L^2 norm of the complete solution
    check convergence in L^2 norm
    $i \leftarrow i + 1$
postprocess total solution

It should be noted that both the Bubnov-Galerkin and the Petrov-Galerkin approaches were implemented for the space-time method. While the basis functions are not the same for the trial and test function spaces in the second approach, the differences are only in the time-discretization and, therefore, no changes were introduced in the notation. When evaluating the shape functions, however, care must be taken to distinguish whether the term corresponds to the primary variable or the test function.

4.6 Nonlinear Space-Time method

Finally, for the last studied case, the space-time method is now combined with a problem including nonlinear terms. This time, the resulting system will be structurally similar to the one in Section 4.3 given that both mass and stiffness terms can be considered as a single joint term.

4.6.1 Partitioned problem formulation

As in Section 4.4, the problem analyzed corresponds to the full version of Equation 2.8, which can be expressed by:

$$\langle \rho c(\mathbf{x}, T)\nabla_t T, v\rangle - \mathscr{A}\left(T, k(\mathbf{x}, T), v\right) = \mathscr{F}\left(v\right) \quad \forall v \in \mathbf{V} \tag{4.82}$$

when using the definitions from Equation 4.73. This equation can then be written in residual form as follows:

$$R_b\left(T\right) = \mathscr{F}\left(v\right) - \langle \rho c(\mathbf{x}, T)\nabla_t T, v\rangle + \mathscr{A}\left(T, k(\mathbf{x}, T), v\right) \overset{!}{=} 0 \quad \forall v \in \mathbf{V} \tag{4.83}$$

for the full problem, or as

$$\begin{cases} R_b\left(T_b, T_o\right) = \mathscr{F}\left(v_b\right) - \langle \rho c(\mathbf{x}, T_b + T_o)\nabla_t \left(T_b + T_o\right), v_b\rangle \\ \qquad\qquad + \mathscr{A}\left(T_b + T_o, k(\mathbf{x}, T_b + T_o), v_b\right) \overset{!}{=} 0 \quad \forall v_b \in \mathbf{V}_b & (4.84a) \\ R_o\left(T_b, T_o\right) = \mathscr{F}\left(v_o\right) - \langle \rho c(\mathbf{x}, T_b + T_o)\nabla_t \left(T_b + T_o\right), v_o\rangle \\ \qquad\qquad + \mathscr{A}\left(T_b + T_o, k(\mathbf{x}, T_b + T_o), v_o\right) \overset{!}{=} 0 \quad \forall v_o \in \mathbf{V}_o, & (4.84b) \end{cases}$$

when splitting into base and overlay problems (based on the linearity of the finite element spaces $\mathbf{V} = \mathbf{V}_b \oplus \mathbf{V}_o$. The residual equation can then be subjected to the nonlinear Gauss-Seidel-Newton iterative process so that in the i-th outer iteration reads:

$$\begin{cases} R_b\left(T_b^i, T_o^{i-1}\right) = \mathscr{F}\left(v_b\right) - \langle \rho c\left(\mathbf{x}, T_b^i + T_o^{i-1}\right)\nabla_t \left(T_b^i + T_o^{i-1}\right), v_b\rangle \\ \qquad\qquad + \mathscr{A}\left(T_b^i + T_o^{i-1}, k(\mathbf{x}, T_b^i + T_o^{i-1}), v_b\right) \overset{!}{=} 0 & (4.85a) \\ R_o\left(T_b^i, T_o^i\right) = \mathscr{F}\left(v_o\right) - \langle \rho c\left(\mathbf{x}, T_b^i + T_o^i\right)\nabla_t \left(T_b^i + T_o^i\right), v_o\rangle \\ \qquad\qquad + \mathscr{A}\left(T_b^i + T_o^i, k(\mathbf{x}, T_b^i + T_o^i), v_o\right) \overset{!}{=} 0. & (4.85b) \end{cases}$$

Linearizing under the Newton method with a single iteration and merging the Gauss-Seidel and Newton indices as l then produces:

$$\begin{cases} R_b\left(T_b^l, T_o^{l-1}\right) \approx R_b\left(T_b^{l-1}, T_o^{l-1}\right) + DR_b\left(T_b^{l-1}, T_o^{l-1}\right)\left[\delta T_b\right] \overset{!}{=} 0 & (4.86a) \\ R_o\left(T_b^l, T_o^l\right) \approx R_o\left(T_b^l, T_o^{l-1}\right) + DR_o\left(T_b^l, T_o^{l-1}\right)\left[\delta T_o\right] \overset{!}{=} 0, & (4.86b) \end{cases}$$

with update rule:

$$T_b^l = T_b^{l-1} + \delta T_b \tag{4.87a}$$
$$T_o^l = T_o^{l-1} + \delta T_o. \tag{4.87b}$$

Enforcing the zero value of the residual at a solution value, the system from Equations 4.86a and 4.86b is reorganized into:

$$\begin{cases} -DR_b\left(T_b^{l-1}, T_o^{l-1}\right)[\delta T_b] = R_b\left(T_b^{l-1}, T_o^{l-1}\right) & (4.88a) \\ -DR_o\left(T_b^{l}, T_o^{l-1}\right)[\delta T_o] = R_o\left(T_b^{l}, T_o^{l-1}\right) & (4.88b) \end{cases}$$

where the residuals in the system are described, after applying integration by parts and neglecting boundary terms, by:

$$\begin{aligned} R_b\left(T_b^{l-1}, T_o^{l-1}\right) = \int_\Omega f(\mathbf{x}) v_b \, d\Omega - \int_\Omega v_b \, \rho c\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right) \nabla_t \left(T_b^{l-1} + T_o^{l-1}\right) \, d\Omega \\ - \int_\Omega \nabla_{\mathbf{x}} v_b \, k\left(\mathbf{x}, T_b^{l-1} + T_o^{l-1}\right) \nabla_{\mathbf{x}} \left(T_b^{l-1} + T_o^{l-1}\right) \, d\Omega \end{aligned} \tag{4.89}$$

for the base equation and

$$\begin{aligned} R_o\left(T_b^{l}, T_o^{l-1}\right) = \int_\Omega f(\mathbf{x}) v_o \, d\Omega - \int_\Omega v_o \, \rho c\left(\mathbf{x}, T_b^{l} + T_o^{l-1}\right) \nabla_t \left(T_b^{l} + T_o^{l-1}\right) \, d\Omega \\ - \int_\Omega \nabla_{\mathbf{x}} v_o \, k\left(\mathbf{x}, T_b^{l} + T_o^{l-1}\right) \nabla_{\mathbf{x}} \left(T_b^{l} + T_o^{l-1}\right) \, d\Omega \end{aligned} \tag{4.90}$$

for the overlay. Just as before, the directional derivatives can be evaluated via the product rule such that:

$$
\begin{aligned}
DR_b\left(T_b^{l-1}, T_o^{l-1}\right)[\delta T_b] = & \int_\Omega \cancelto{0}{D\left(f(\mathbf{x}) v_b\right)[\delta T_b]}\, d\Omega \\
& - \int_\Omega \cancelto{0}{D\left(v_b\right)[\delta T_b]}\, \rho c\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right) \nabla_t\left(T_b^{l-1}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega v_b\, D\left(\rho c\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b] \nabla_t\left(T_b^{l-1}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega v_b\, \rho c\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right) D\left(\nabla_t\left(T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b]\, d\Omega \\
& - \int_\Omega \cancelto{0}{D\left(\nabla_{\mathbf{x}} v_b\right)[\delta T_b]}\, k\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right) \nabla_{\mathbf{x}}\left(T_b^{l-1}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla_{\mathbf{x}} v_b\, D\left(k\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b] \nabla_{\mathbf{x}}\left(T_b^{l-1}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla_{\mathbf{x}} v_b\, k\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right) D\left(\nabla_{\mathbf{x}}\left(T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b]\, d\Omega
\end{aligned}
\tag{4.91}
$$

and, analogously

$$
\begin{aligned}
DR_o\left(T_b^{l}, T_o^{l-1}\right)[\delta T_o] = & \int_\Omega \cancelto{0}{D\left(f(\mathbf{x}) v_o\right)[\delta T_o]}\, d\Omega \\
& - \int_\Omega \cancelto{0}{D\left(v_o\right)[\delta T_o]}\, \rho c\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right) \nabla_t\left(T_b^{l}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega v_o\, D\left(\rho c\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o] \nabla_t\left(T_b^{l}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega v_o\, \rho c\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right) D\left(\nabla_t\left(T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o]\, d\Omega \\
& - \int_\Omega \cancelto{0}{D\left(\nabla_{\mathbf{x}} v_o\right)[\delta T_o]}\, k\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right) \nabla_{\mathbf{x}}\left(T_b^{l}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla_{\mathbf{x}} v_o\, D\left(k\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o] \nabla_{\mathbf{x}}\left(T_b^{l}+T_o^{l-1}\right)\, d\Omega \\
& - \int_\Omega \nabla_{\mathbf{x}} v_o\, k\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right) D\left(\nabla_{\mathbf{x}}\left(T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o]\, d\Omega.
\end{aligned}
\tag{4.92}
$$

The individual derivatives are then evaluated to be:

$$D\left(\rho c\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b] = \frac{d}{d\varepsilon}\left[\rho c\left(\mathbf{x}, T_b^{l-1}+\varepsilon\delta T_b+T_o^{l-1}\right)\right]_{\varepsilon=0} \approx \left(\frac{d\rho}{dT}c+\rho\frac{dc}{dT}\right)\Big|_{\mathbf{x},T_b^{l-1}+T_o^{l-1}}\delta T_b \tag{4.93a}$$

$$D\left(k\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b] = \frac{d}{d\varepsilon}\left[k\left(\mathbf{x}, T_b^{l-1}+\varepsilon\delta T_b+T_o^{l-1}\right)\right]_{\varepsilon=0} \approx \frac{dk}{dT}\Big|_{\mathbf{x},T_b^{l-1}+T_o^{l-1}}\delta T_b \tag{4.93b}$$

$$D\left(\nabla_t\left(T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b] = \frac{d}{d\varepsilon}\left[\nabla_t\left(T_b^{l-1}+\varepsilon\delta T_b+T_o^{l-1}\right)\right]_{\varepsilon=0} \approx \nabla_t\left(\delta T_b\right) \tag{4.93c}$$

$$D\left(\nabla_{\mathbf{x}}\left(T_b^{l-1}+T_o^{l-1}\right)\right)[\delta T_b] = \frac{d}{d\varepsilon}\left[\nabla_{\mathbf{x}}\left(T_b^{l-1}+\varepsilon\delta T_b+T_o^{l-1}\right)\right]_{\varepsilon=0} \approx \nabla_{\mathbf{x}}\left(\delta T_b\right) \tag{4.93d}$$

$$D\left(\rho c\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o] = \frac{d}{d\varepsilon}\left[\rho c\left(\mathbf{x}, T_b^{l}+T_o^{l-1}+\varepsilon\delta T_o\right)\right]_{\varepsilon=0} \approx \left(\frac{d\rho}{dT}c+\rho\frac{dc}{dT}\right)\Big|_{\mathbf{x},T_b^{l}+T_o^{l-1}}\delta T_o \tag{4.93e}$$

$$D\left(k\left(\mathbf{x}, T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o] = \frac{d}{d\varepsilon}\left[k\left(\mathbf{x}, T_b^{l}+T_o^{l-1}+\varepsilon\delta T_o\right)\right]_{\varepsilon=0} \approx \frac{dk}{dT}\Big|_{\mathbf{x},T_b^{l}+T_o^{l-1}}\delta T_o \tag{4.93f}$$

$$D\left(\nabla_t\left(T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o] = \frac{d}{d\varepsilon}\left[\nabla_t\left(T_b^{l}+T_o^{l-1}+\varepsilon\delta T_o\right)\right]_{\varepsilon=0} \approx \nabla_t\left(\delta T_o\right) \tag{4.93g}$$

$$D\left(\nabla_{\mathbf{x}}\left(T_b^{l}+T_o^{l-1}\right)\right)[\delta T_o] = \frac{d}{d\varepsilon}\left[\nabla_{\mathbf{x}}\left(T_b^{l}+T_o^{l-1}+\varepsilon\delta T_o\right)\right]_{\varepsilon=0} \approx \nabla_{\mathbf{x}}\left(\delta T_o\right), \tag{4.93h}$$

and can be substituted back into their corresponding expressions such that Equations 4.88a and 4.88b finally yield:

$$\begin{aligned}
&\int_\Omega v_b\ \left(\frac{d\rho}{dT}c+\rho\frac{dc}{dT}\right)\Big|_{\mathbf{x},T_b^{l-1}+T_o^{l-1}}\delta T_b\ \nabla_t\left(T_b^{l-1}+T_o^{l-1}\right)\ d\Omega \\
&\quad+\int_\Omega v_b\ \rho c\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\nabla_t\left(\delta T_b\right)\ d\Omega \\
&\quad+\int_\Omega \nabla_{\mathbf{x}} v_b\ \frac{dk}{dT}\Big|_{\mathbf{x},T_b^{l-1}+T_o^{l-1}}\delta T_b\ \nabla_{\mathbf{x}}\left(T_b^{l-1}+T_o^{l-1}\right)\ d\Omega \\
&\quad+\int_\Omega \nabla_{\mathbf{x}} v_b\ k\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\nabla_{\mathbf{x}}\left(\delta T_b\right)\ d\Omega \\
&\qquad=\int_\Omega f(\mathbf{x}) v_b\ d\Omega-\int_\Omega v_b\ \rho c\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\nabla_t\left(T_b^{l-1}+T_o^{l-1}\right)\ d\Omega \\
&\qquad\qquad-\int_\Omega \nabla_{\mathbf{x}} v_b\ k\left(\mathbf{x}, T_b^{l-1}+T_o^{l-1}\right)\nabla_{\mathbf{x}}\left(T_b^{l-1}+T_o^{l-1}\right)\ d\Omega
\end{aligned} \tag{4.94}$$

and

$$\begin{aligned}
&\int_\Omega v_o \left(\frac{d\rho}{dT}c + \rho\frac{dc}{dT}\right)\Big|_{\mathbf{x},T_b^l+T_o^{l-1}} \delta T_o \, \nabla_t\left(T_b^l + T_o^{l-1}\right) \, d\Omega \\
&\quad + \int_\Omega v_o \, \rho c\left(\mathbf{x}, T_b^l + T_o^{l-1}\right) \nabla_t\left(\delta T_o\right) \, d\Omega \\
&\quad + \int_\Omega \nabla_{\mathbf{x}} v_o \, \frac{dk}{dT}\Big|_{\mathbf{x},T_b^l+T_o^{l-1}} \delta T_b \, \nabla_{\mathbf{x}}\left(T_b^l + T_o^{l-1}\right) \, d\Omega \\
&\quad + \int_\Omega \nabla_{\mathbf{x}} v_o \, k\left(\mathbf{x}, T_b^l + T_o^{l-1}\right) \nabla_{\mathbf{x}}\left(\delta T_o\right) \, d\Omega \\
&\qquad = \int_\Omega f(\mathbf{x}) v_o \, d\Omega - \int_\Omega v_o \, \rho c\left(\mathbf{x}, T_b^l + T_o^{l-1}\right) \nabla_t\left(T_b^l + T_o^{l-1}\right) \, d\Omega \\
&\qquad\qquad - \int_\Omega \nabla_{\mathbf{x}} v_o \, k\left(\mathbf{x}, T_b^l + T_o^{l-1}\right) \nabla_{\mathbf{x}}\left(T_b^l + T_o^{l-1}\right) \, d\Omega,
\end{aligned} \tag{4.95}$$

which can then be rewritten in a more compact form as the system:

$$\begin{cases}
\mathbf{J}_{bb}\left(T_b^{l-1}, T_o^{l-1}\right)\delta T_b = \mathbf{R}_b\left(T_b^{l-1}, T_o^{l-1}\right) & \text{(4.96a)} \\
\mathbf{J}_{oo}\left(T_b^{l}, T_o^{l-1}\right)\delta T_o = \mathbf{R}_o\left(T_b^{l}, T_o^{l-1}\right). & \text{(4.96b)}
\end{cases}$$

with its individual terms developed in Table 4.6.

Table 4.6: Multi-level hp finite element matrices for a nonlinear transient heat equation resolved by the one-step Gauss-Seidel-Newton scheme with a space-time discretization

Term	**Symbol**	**Definition**
Base Jacobian Matrix	$\mathbf{J_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\left(\mathbf{M_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right) + \mathbf{M'_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right) + \mathbf{K_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right) + \mathbf{K'_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)\right)$
Overlay Jacobian Matrix	$\mathbf{J_{oo}}\left(T_b^{l}, T_o^{l-1}\right)$	$\left(\mathbf{M_{oo}}\left(T_b^{l}, T_o^{l-1}\right) + \mathbf{M'_{oo}}\left(T_b^{l}, T_o^{l-1}\right) + \mathbf{K_{oo}}\left(T_b^{l}, T_o^{l-1}\right) + \mathbf{K'_{oo}}\left(T_b^{l}, T_o^{l-1}\right)\right)$
Base Residual Vector	$\mathbf{R_b}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\mathbf{F_b} - \mathbf{K_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right) T_b^{l-1} - \mathbf{K_{bo}}\left(T_b^{l-1}, T_o^{l-1}\right) T_o^{l-1} - \mathbf{M_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right) T_b^{l-1} - \mathbf{M_{bo}}\left(T_b^{l-1}, T_o^{l-1}\right) T_o^{l-1}$

Continued on next page

Table 4.6 – *Continued from previous page*

Term	**Symbol**	**Definition**
Overlay Residual Vector	$\mathbf{R_o}\left(T_b^l, T_o^{l-1}\right)$	$\mathbf{F_o} - \mathbf{K_{oo}}\left(T_b^l, T_o^{l-1}\right) T_o^{l-1} - \mathbf{K_{ob}}\left(T_b^l, T_o^{l-1}\right) T_b^l - \mathbf{M_{oo}}\left(T_b^l, T_o^{l-1}\right) T_o^{l-1} - \mathbf{M_{ob}}\left(T_b^l, T_o^{l-1}\right) T_b^l$
Base Stiffness Matrix	$\mathbf{K_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{B}_{\mathbf{x},b}^T \, k\left(\mathbf{x}, \mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{\mathbf{x},b} \, d\Omega$
Derived Base Stiffness Matrix	$\mathbf{K'_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{B}_{\mathbf{x},b}^T \left.\frac{dk}{dT}\right\|_{\mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1}} \left(\mathbf{B}_{\mathbf{x},b} T_b^{l-1} + \mathbf{B}_{\mathbf{x},o} T_o^{l-1}\right) \mathbf{N}_b \, d\Omega$
Base Coupling Stiffness Matrix	$\mathbf{K_{bo}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_{\mathbf{x},b}^T \, k\left(\mathbf{x}, \mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{\mathbf{x},o} \, d\Omega$
Overlay Stiffness Matrix	$\mathbf{K_{oo}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_{\mathbf{x},o}^T \, k\left(\mathbf{x}, \mathbf{N}_b T_b^l + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{\mathbf{x},o} \, d\Omega$
Derived Overlay Stiffness Matrix	$\mathbf{K'_{oo}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_{\mathbf{x},o}^T \left.\frac{dk}{dT}\right\|_{\mathbf{N}_b T_b^l + \mathbf{N}_o T_o^{l-1}} \left(\mathbf{B}_{\mathbf{x},b} T_b^l + \mathbf{B}_{\mathbf{x},o} T_o^{l-1}\right) \mathbf{N}_o \, d\Omega$
Overlay Coupling Stiffness Matrix	$\mathbf{K_{ob}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{B}_{\mathbf{x},o}^T \, k\left(\mathbf{x}, \mathbf{N}_b T_b^l + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{\mathbf{x},b} \, d\Omega$
Base Mass Matrix	$\mathbf{M_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{N}_b^T \, \rho c\left(\mathbf{x}, \mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{t,b} \, d\Omega$
Derived Base Mass Matrix	$\mathbf{M'_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_b} \mathbf{N}_b^T \left.\left(\frac{d\rho}{dT} c + \rho \frac{dc}{dT}\right)\right\|_{\mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1}} \left(\mathbf{B}_{t,b} T_b^{l-1} + \mathbf{B}_{t,o} T_o^{l-1}\right) \mathbf{N}_b \, d\Omega$
Base Coupling Mass Matrix	$\mathbf{M_{bo}}\left(T_b^{l-1}, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_b^T \, \rho c\left(\mathbf{x}, \mathbf{N}_b T_b^{l-1} + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{t,o} \, d\Omega$
Overlay Mass Matrix	$\mathbf{M_{oo}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_o^T \, \rho c\left(\mathbf{x}, \mathbf{N}_b T_b^l + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{t,o} \, d\Omega$
Derived Overlay Mass Matrix	$\mathbf{M'_{oo}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_o^T \left.\left(\frac{d\rho}{dT} c + \rho \frac{dc}{dT}\right)\right\|_{\mathbf{N}_b T_b^l + \mathbf{N}_o T_o^{l-1}} \left(\mathbf{B}_{t,b} T_b^l + \mathbf{B}_{t,o} T_o^{l-1}\right) \mathbf{N}_o \, d\Omega$
Overlay Coupling Mass Matrix	$\mathbf{M_{ob}}\left(T_b^l, T_o^{l-1}\right)$	$\int_{\Omega_o} \mathbf{N}_o^T \, \rho c\left(\mathbf{x}, \mathbf{N}_b T_b^l + \mathbf{N}_o T_o^{l-1}\right) \mathbf{B}_{t,b} \, d\Omega$
Base Source Vector	$\mathbf{F_b}$	$\int_{\Omega_b} \mathbf{N}_b^T f(\mathbf{x}, t) \, d\Omega$
Overlay Source Vector	$\mathbf{F_o}$	$\int_{\Omega_o} \mathbf{N}_o^T f(\mathbf{x}, t) \, d\Omega$

4.6.2 Method Implementation

To conclude this section, Algorithm 6 describes the implemented routine, which applies the one-step nonlinear Gauss-Seidel-Newton scheme under the nonlinear space-time discretization as described in the former. Note that since the time dependency is handled internally by the discretization, the source terms can again be precomputed outside of the iteration loop, just like in the nonlinear stationary case.

Algorithm 6: Multi-level hp algorithm for nonlinear space-time problems

define overlay location in $\Omega \times [0, T]$
create global base mesh and overlay mesh in specified region
remove degrees of freedom to ensure linear independence
set Gauss-Seidel-Newton iteration counter $l \leftarrow 0$
while *not converged* **and** *below maximum iteration number* **do**
  Compute *base solution increment*
    compute stiffness matrices $\mathbf{K_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$ and $\mathbf{K'_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$, mass matrices $\mathbf{M_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$ and $\mathbf{M'_{bb}}\left(T_b^{l-1}, T_o^{l-1}\right)$, and source term $\mathbf{F_b}$
    compute coupling terms $\mathbf{K_{bo}}\left(T_b^{l-1}, T_o^{l-1}\right) T_o^{l-1}$ and $\mathbf{M_{bo}}\left(T_b^{l-1}, T_o^{l-1}\right) T_o^{l-1}$
    apply base system constraints (boundary and initial conditions)
    solve Equation 4.96a
    $T_b^l \leftarrow T_b^{l-1} + \delta T_b$
  Compute *overlay solution increment*
    compute stiffness matrices $\mathbf{K_{oo}}\left(T_b^l, T_o^{l-1}\right)$ and $\mathbf{K'_{oo}}\left(T_b^l, T_o^{l-1}\right)$, mass matrices $\mathbf{M_{oo}}\left(T_b^l, T_o^{l-1}\right)$ and $\mathbf{M'_{oo}}\left(T_b^l, T_o^{l-1}\right)$, and source term $\mathbf{F_o}$
    compute coupling terms $\mathbf{K_{ob}}\left(T_b^l, T_o^{l-1}\right) T_b^l$ and $\mathbf{M_{ob}}\left(T_b^l, T_o^{l-1}\right) T_b^l$
    apply overlay system constraints (homogeneous Dirichlet)
    solve Equation 4.96b
    $T_o^l \leftarrow T_o^{l-1} + \delta T_o$
  compute L^2 norm of the complete solution
  check convergence in L^2 norm
  $l \leftarrow l + 1$
postprocess total solution

Chapter 5

Verification of the multi-level *hp*–Finite Element Method for heat transfer problems

In this chapter, the implemented code for each of the cases from Chapter 4 is verified. The results from the code were compared both to the analytical values of the manufactured solutions which generated the setups, as well as equivalent simulations run in AdhoC++ for a monolithical version of the multi-level *hp*-FEM. In addition to this, a comparison was made between one-dimensional examples computed with the time-stepping FEM code and with the space-time code. Lastly, a comparison of the Bubnov-Galerkin and Petrov-Galerkin methods for the linear space-time implementation was performed.

Since all of the problems considered were relatively simple and the code has not been optimized, the time required for the computation is not a robust measurement of the computational cost of simulation. Instead, the number of linear system solutions required, which is the most expensive operation in the presented pipeline, was taken as a reference. During each iteration, two linear systems are solved: one for the base Temperature T_b and one for the overlay T_o. In addition to this, setting up the initial conditions of a transient problem involved two linear system solutions; one for the total coupled system solution (with size equal to the sum of sizes of T_b and T_o), and one for the projection of the base solution onto the overlay. Ideally, the size of both systems should be similar, so the overall cost of solving the problem can be estimated at approximately twice the cost of solving one linear system of the size of either system times the number of iterations. For transient cases, this must, of course, be repeated for each time step. In general, the cost of solving a problem with the proposed method would seem to increase proportionally with the number of iterations required in its solution. In contrast to this, solving the monolithic *hp*–system is much more expensive in terms of memory usage due to the increased problem size and, for instance, in the case of a direct solver, the factorization step.

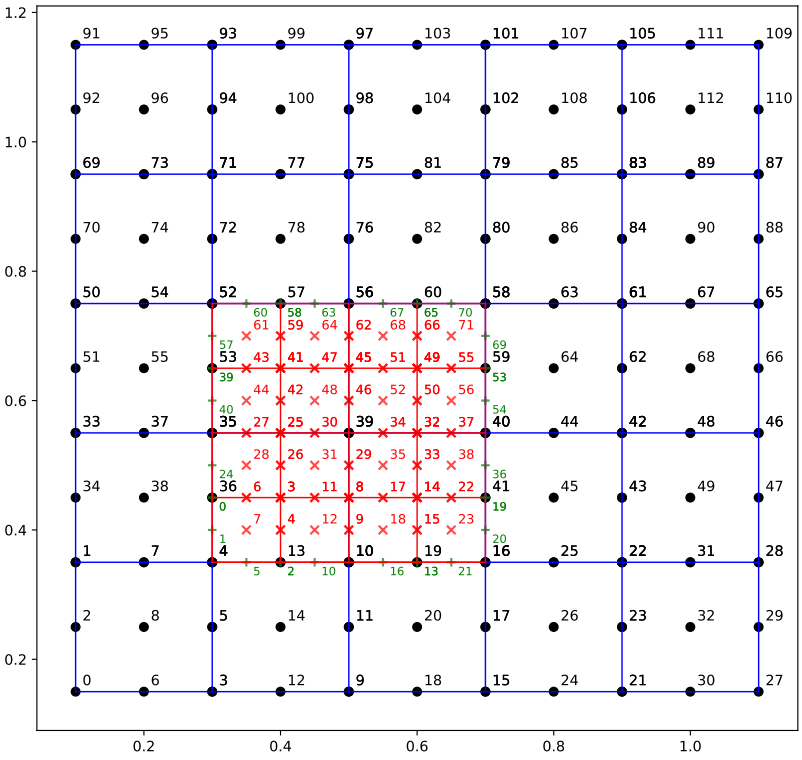


Figure 5.1: Mesh definition for the linear stationary case

5.1 Linear stationary

The aim of this first study was to verify the general functionality of the proposed partitioned solution of the multi-level *hp*–method in conjunction with a higher order multidimensional problem. To this effect, a problem with the following manufactured solution was chosen as reference:

$$T(x,y) = \mathbf{e}^{-\left(\frac{(x-0.5)^2+(y-0.5)^2}{2\times 0.05^2}\right)}, \tag{5.1}$$

where the solution represents a Gaussian peak, the typical beam intensity pattern used in the SLM additive manufacturing process. For the discretization, the $[0.10, 1.10] \times [0.15, 1.15]$ domain was subdivided into a 5×5 base mesh with quadratic elements and a 2×2 subregion was overlaid. Each base element was then bisected in both dimensions to form the overlay elements. The described meshes and their corresponding DOF numbers are displayed in Figure 5.1.

While the implemented partitioned approach allows any number of overlay elements per base element, this number was kept at two for comparison with the AdhoC++ monolithic *hp*–code. Additionally, the physical parameter used for the heat conductivity was:

$$\kappa(x,y) = (10 + x)\sin(y) + 10.0. \tag{5.2}$$

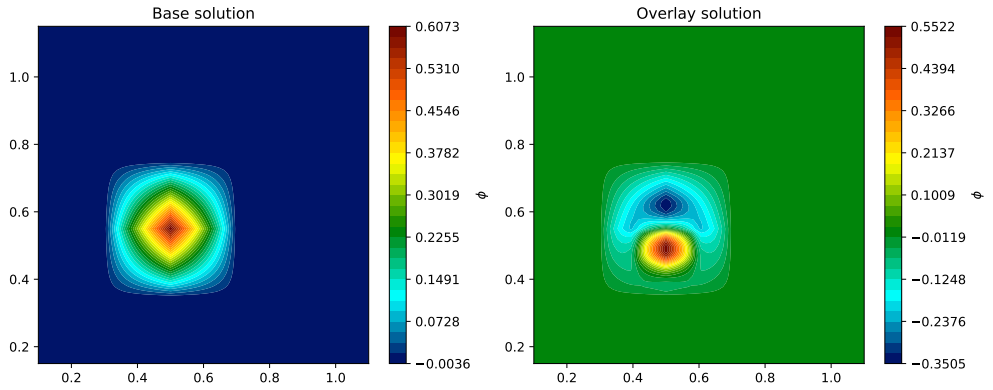


Figure 5.2: Base and overlay solutions of the linear stationary example

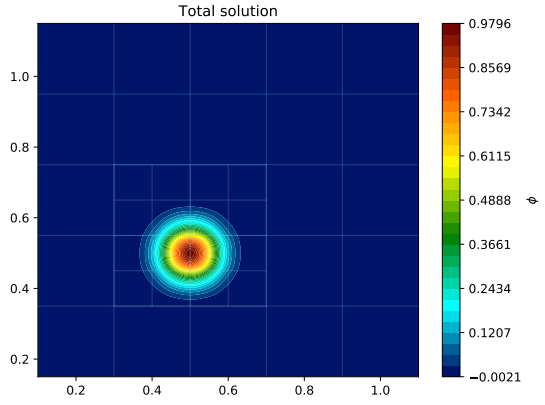


Figure 5.3: Total solution color map of the linear stationary example

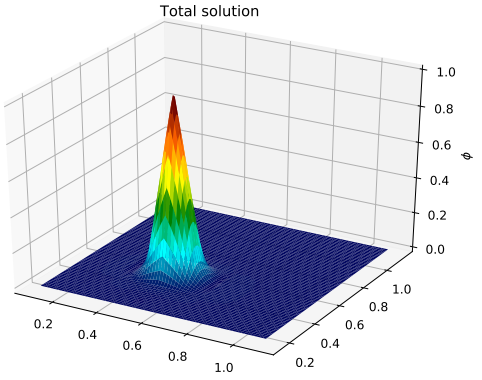


Figure 5.4: Total solution surface plot of the linear stationary example

While this type of trigonometrical heat conductivity is nonphysical, it allows us to further test the capabilities of the method.

The Gauss-Seidel solver was set to a convergence tolerance of 1×10^{-10} and the boundary conditions, homogeneous Dirichlet, were applied in the weak sense with a penalty value of 1×10^{8}.

Figure 5.2 displays the individual solutions to the base and overlay problems, which highlights the partitioned problem solution. In these figures, the lack of higher order modes on the overlaid region of the base mesh explains the decreased quality of the depicted solution. However, the overlay solution compensates for this fact and the total solution can be observed in Figures 5.3 and 5.4.

In addition to the previous studies, both h- and p-convergence tests were performed with the same initial setup, as shown in Figure 5.5. In this case, h denotes the number of subdivisions taken for each element (e.g. at $h = 3$ each original base and overlay element has been divided into three in each direction). The h-refinement curves feature an algebraic convergence rate, whereas p-refinement leads to an exponential one. Iteration numbers corresponding to these

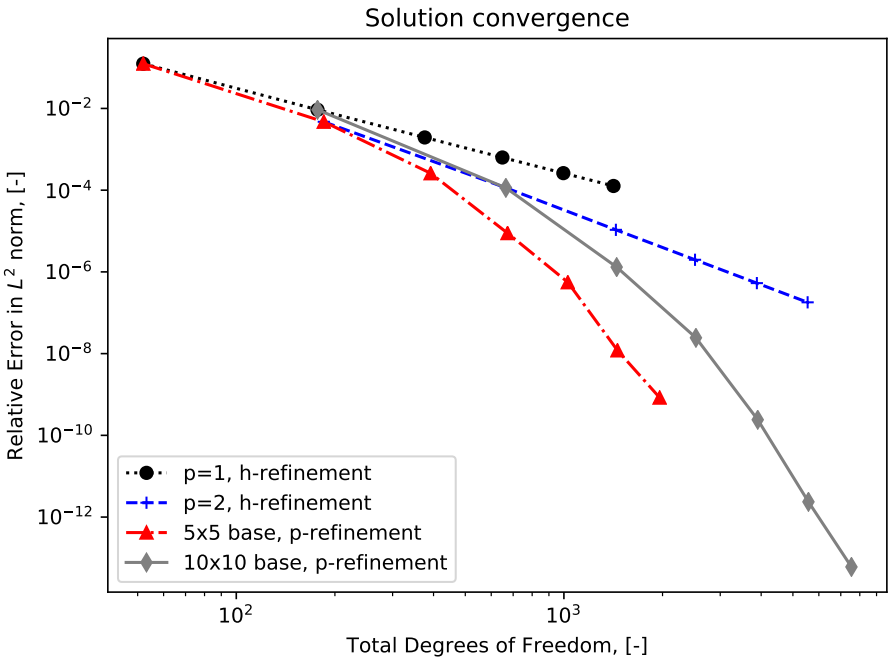


Figure 5.5: Convergence studies for the linear stationary example

studies are included in Appendix A.

The results obtained for the linear stationary example show no obstacles in the application of the partitioned multi-level *hp*–approach in conjunction with a two dimensional stationary problem with higher order shape functions.

5.2 Linear transient

The addition of transient effects serves two purposes: testing the effect of mass coupling terms and providing a result, which may later be compared with the linear space-time formulation. In this case, the manufactured solution considered was:

$$T(x,y,t) = \mathbf{e}^{-\left(\frac{(x-2t-0.5)^2+(y-0.5)^2}{2\times 0.05^2}\right)}, \tag{5.3}$$

which has the same form as the one from the previous case, but travels linearly from the previous location and towards the right over a span of 0.1 time units. The total number of 0.1 units was subdivided into 5 time steps. The base discretization used is the same ($[0.10, 1.10] \times [0.15, 1.15]$ subdivided into a 5×5 base mesh), but now a 3×2 sub-region is overlaid to include the elements at the end location of the Gaussian peak. The described

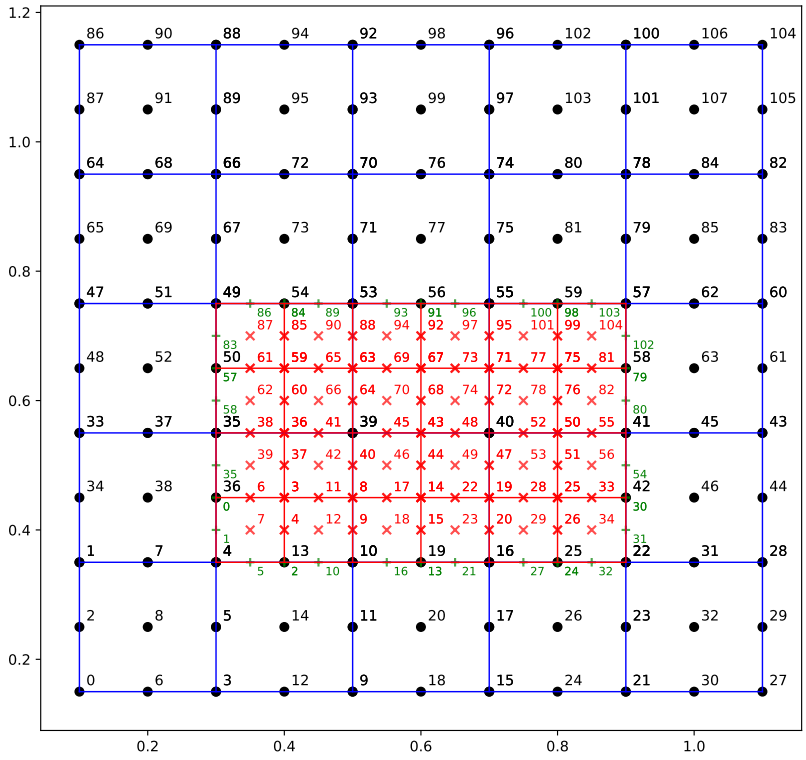


Figure 5.6: Mesh definition for the linear transient case

setup is displayed in Figure 5.6 and the physical parameters used for this example were:

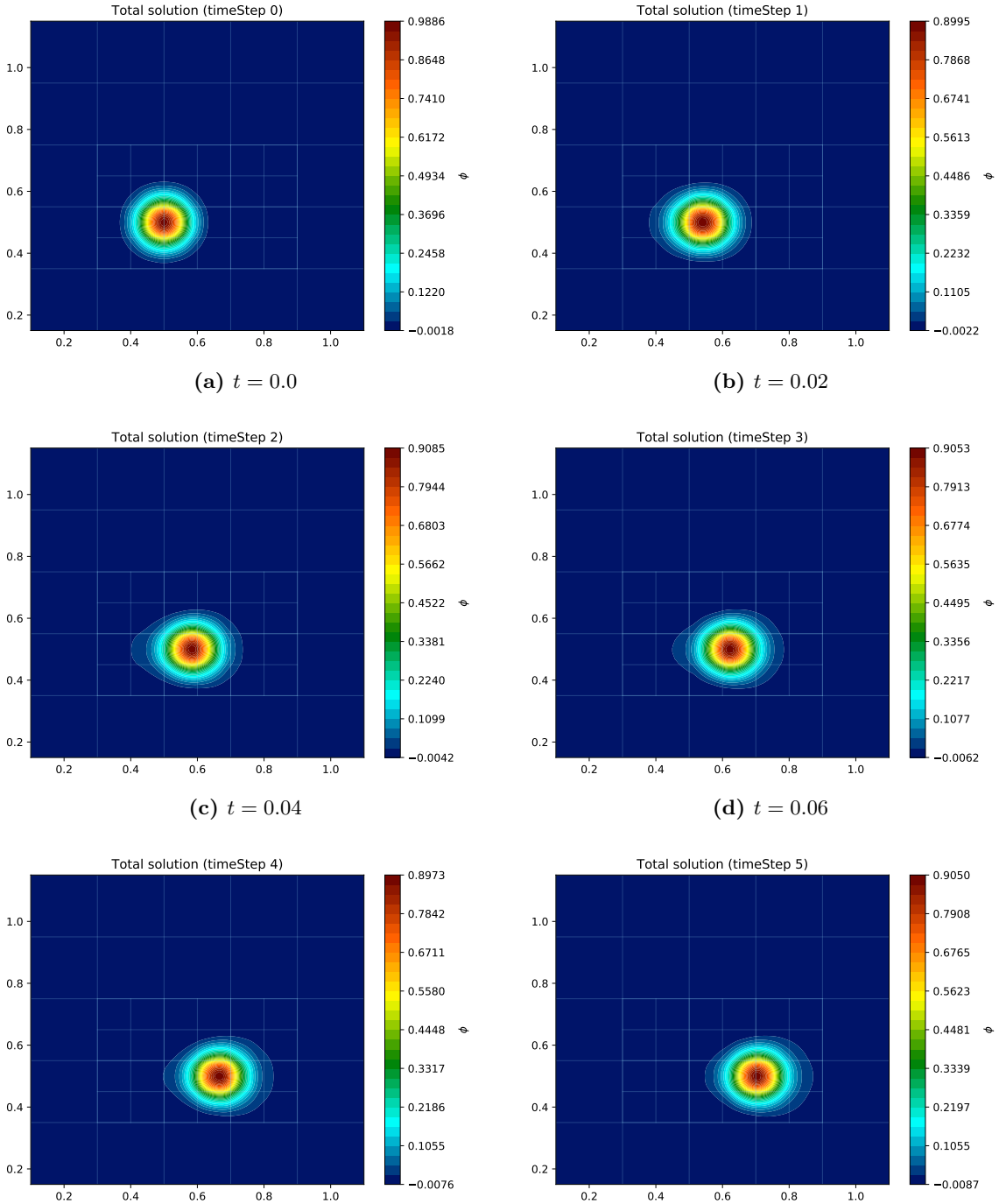
$$\begin{cases} \kappa(x,y) = (10.0 + x)\sin(y) + 10.0 & \text{(5.4a)} \\ \rho(x,y) = (10.0 + y)\sin(x) + 10.0 & \text{(5.4b)} \\ c(x,y) = 5.0. & \text{(5.4c)} \end{cases}$$

For this problem, the Gauss-Seidel solver was now set to a convergence tolerance of 1×10^{-6} while the boundary conditions remain as homogeneous Dirichlet.

Since the viability of the partitioning scheme was already verified for the linear case, only the total solution at each time step is presented (Figure 5.7).

These results, although qualitatively correct, do present a certain degree of dissipation as a result of the chosen time discretization. This is further proven by the convergence tests performed (Figure 5.8) on the same setup with a total of 20 time steps instead. In this case, neither h- nor p-refinement are able to bring the solution error below a certain threshold, where the time discretization error dominates. In order to reduce the relative L^2 norm difference of the solution further, a refinement in time is necessary, i.e. more time steps need to be resolved.

Despite this fact, the verification of the method in the context of linear transient problems is satisfactory and the approach can be used without major difficulties.



(a) $t = 0.0$ **(b)** $t = 0.02$

(c) $t = 0.04$ **(d)** $t = 0.06$

(e) $t = 0.08$ **(f)** $t = 0.1$

Figure 5.7: Total solution color map plots of the linear transient example

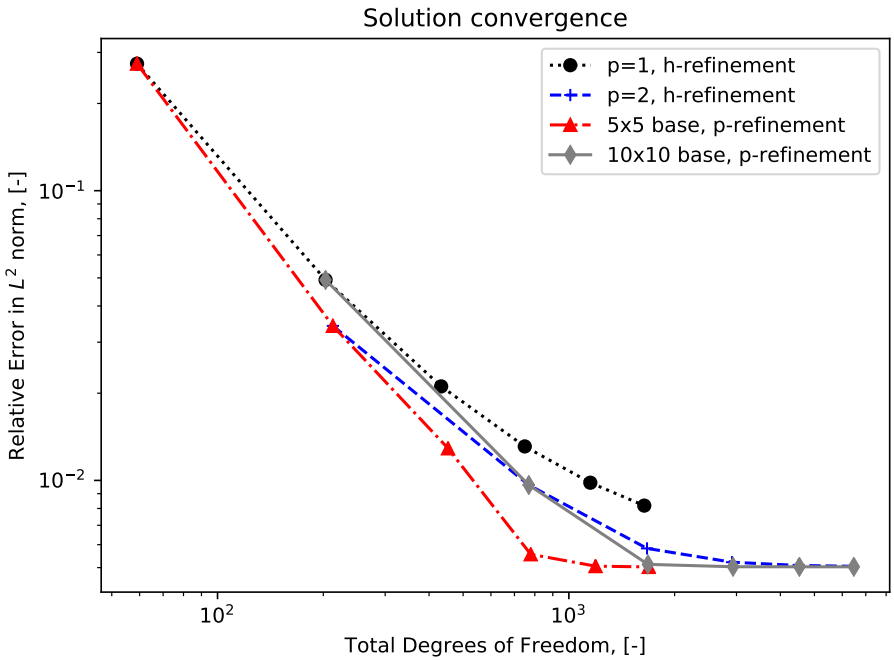


Figure 5.8: Convergence studies for the linear transient example

One-dimensional problem

For later comparison with the space-time approach, a one-dimensional example was also calculated. In this case, the manufactured solution was:

$$T(x,y,t) = \mathbf{e}^{-\left(\frac{(x-t-0.2)^2}{2\times 0.02^2}\right)}, \tag{5.5}$$

which corresponds to the equivalent problem of a traveling peak and employed a total of 40 base elements, 10 of which were overlaid over total of 20 timesteps. Boundary conditions, as before, are taken to be homogeneous Dirichlet and the physical parameters used were:

$$\begin{cases} \kappa(x) = \sin(x) + 10.0 & \text{(5.6a)} \\ \rho(x) = \cos(x) + 5.0 & \text{(5.6b)} \\ c(x) = 1.0. & \text{(5.6c)} \end{cases}$$

The used discretization, as well as some illustrative time step solutions are presented in Figure 5.9. In addition to this, the solution is also depicted in an alternative manner - directly in the space-time domain (see Figure 5.10).

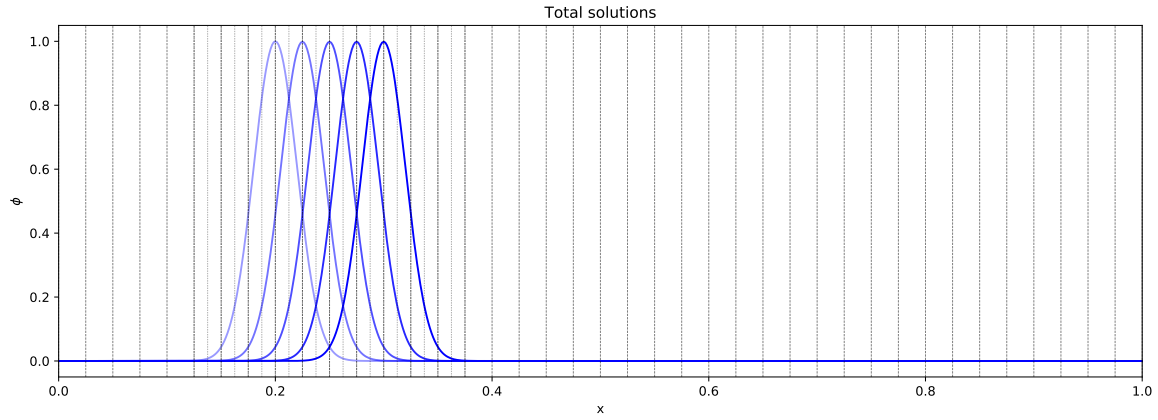


Figure 5.9: Solution for selected time steps for the one-dimensional linear transient test case

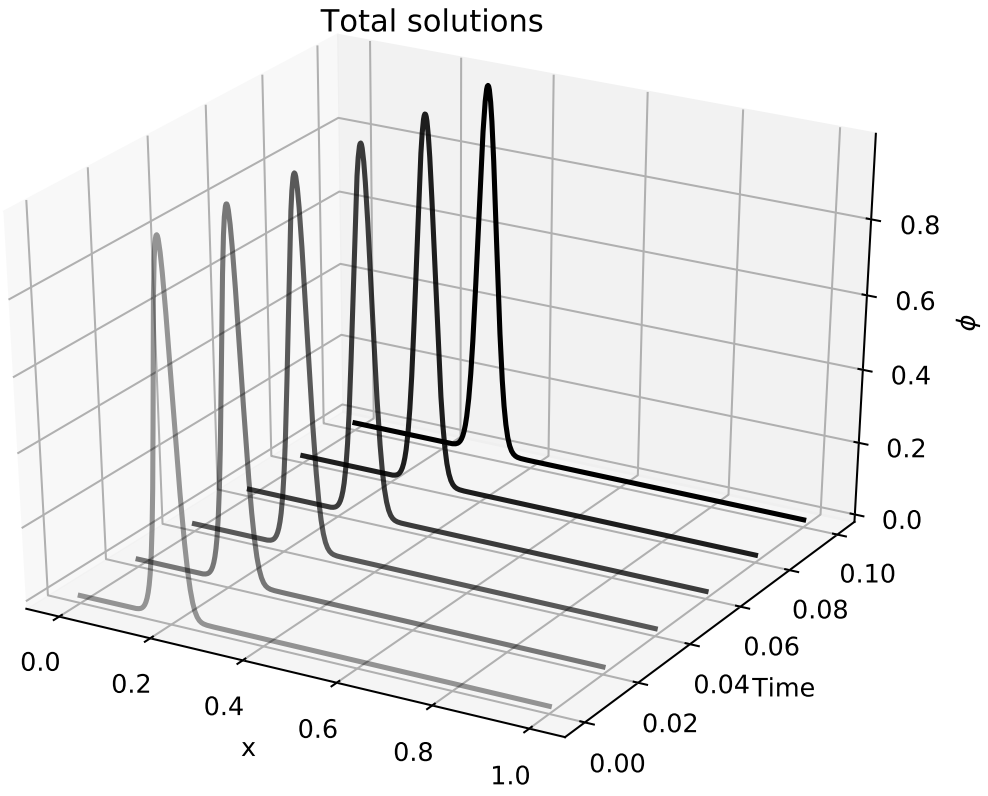


Figure 5.10: Solution over selected time steps plotted in space-time for the one-dimensional linear transient test case

5.3 Nonlinear stationary

Incorporating nonlinear effects, the next problem to be solved was the same as described above in Section 5.1 with the following nonlinear heat conductivity:

$$\kappa(x, y, T) = \sin(T) + 10.0. \tag{5.7}$$

Changing the single value (and the corresponding applied source function), allowed the previous solution to also be used in the verification of the current problem. As indicated, the one-step block Gauss-Seidel-Newton method was used for the solution of the partitioned system. In this case, a final tolerance for the merged iterations was defined at 1×10^{-6}.

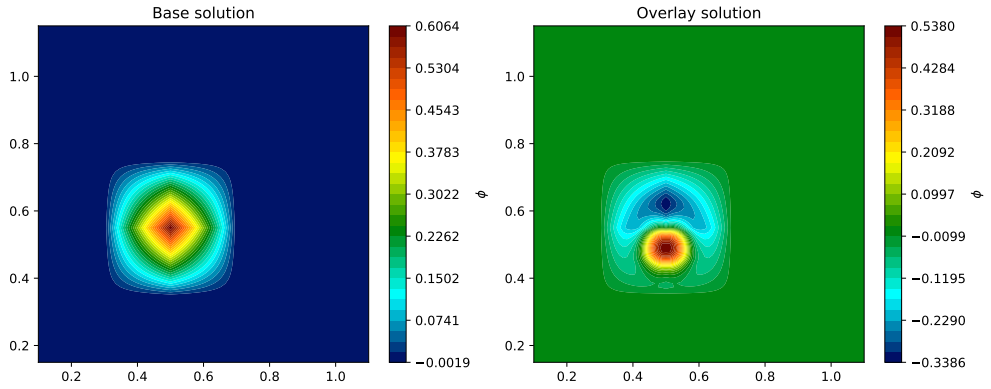


Figure 5.11: Base and overlay solutions of the nonlinear stationary example

As expected, neither the individual solutions of Figure 5.11 nor the total solutions in Figures 5.26 and 5.27 are any worse than those obtained for the linear case once convergence was achieved. Moreover, the convergence tests also demonstrate the expected convergence rates

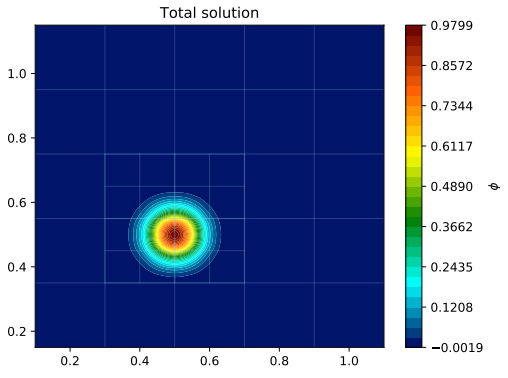


Figure 5.12: Total solution color map of the nonlinear stationary example

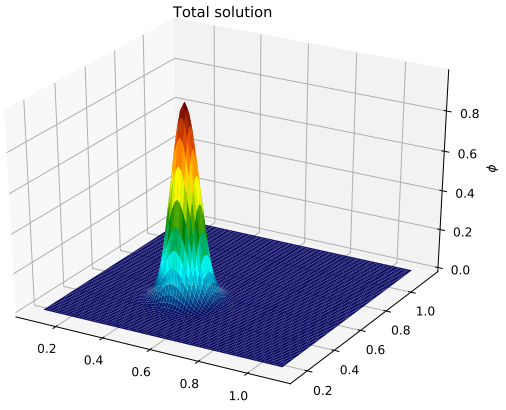


Figure 5.13: Total solution surface plot of the nonlinear stationary example

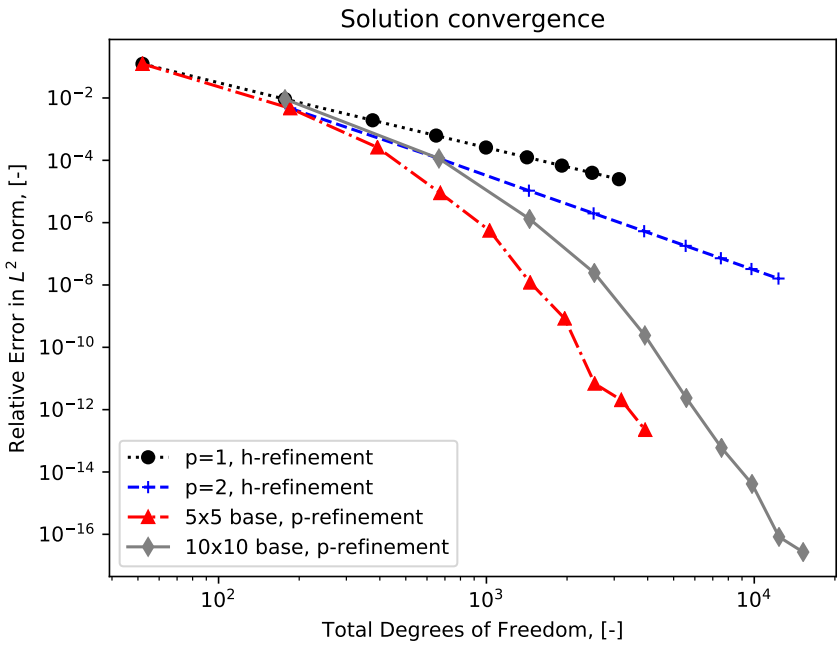


Figure 5.14: Convergence studies for the nonlinear stationary example

for both of the refinement procedures applied: algebraic convergence rate for the h-refinement and exponential for the p-refinement up to machine precision for orders $p = 9, 10$ in the finer starting mesh. The cost of the simulation, however, starts to ramp up for the higher orders (from less than 20 for the linear case to over 150 for $p = 10$), whereas they stay stable or even decrease (between 15 and 29 iterations are required for e.g. $p = 2$) when performing h-refinement.

The complete list of iteration numbers are presented in Table A.3.

5.4 Nonlinear transient

The final test before incorporating the space-time approach is the nonlinear transient case. Rather than repeating the same process as with the linear case, the function chosen for this example was, instead:

$$T(x, y, t) = \mathbf{e}^{-\left(\frac{(x-(0.6-0.1\cos(\pi t)))^2+(y-(0.5+0.1\sin(\pi t)))^2}{2*0.05^2}\right)}, \tag{5.8}$$

which corresponds to a Gaussian curve traveling in a semicircular pattern. In this case, the time interval considered corresponded to 1.0 time units, which were again solved in 5 time steps.

Given the more complex motion expected in the problem, the discretization used was of 6×6 elements for the base mesh, with a 4×3 sub-region refined. The newly defined mesh is

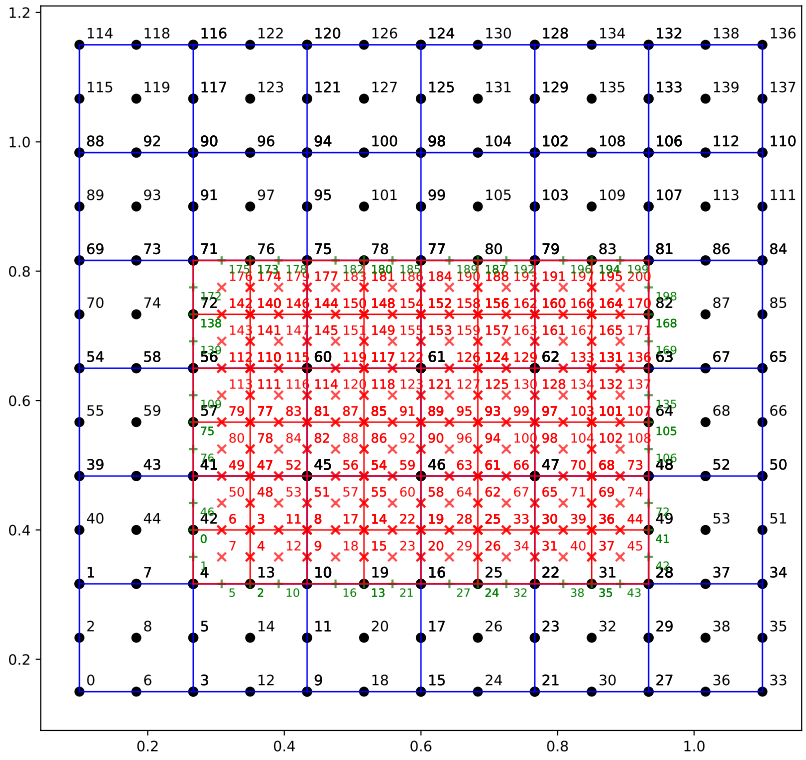


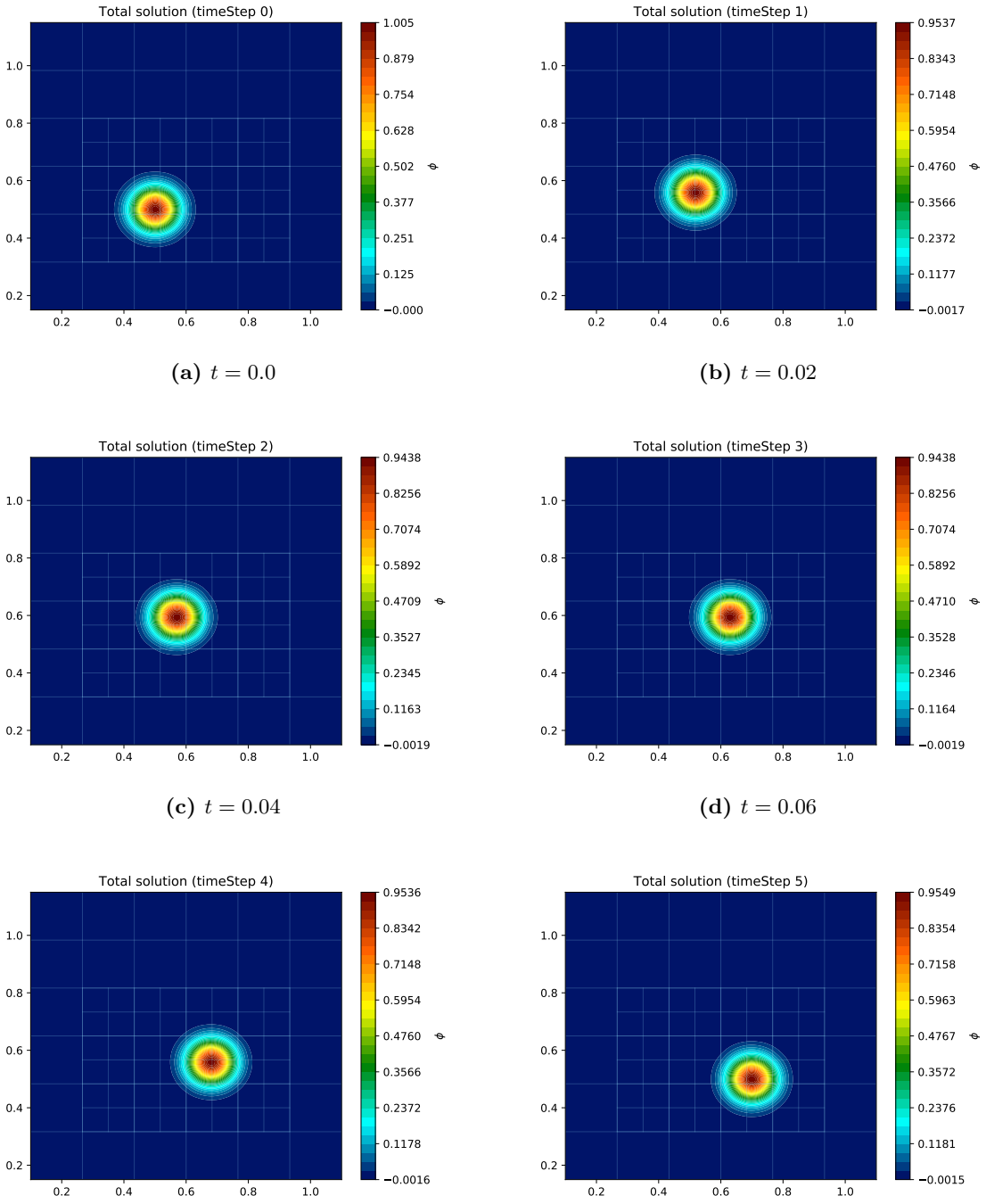
Figure 5.15: Mesh definition for the nonlinear transient case

presented in Figure 5.15. As in the stationary case, the physical parameters:

$$\begin{cases} \kappa(x, y, T) = \sin(T) + 10.0 & \text{(5.9a)} \\ \rho(x, y, T) = \cos(T) + 10.0 & \text{(5.9b)} \\ c(x, y, T) = \sin(T). & \text{(5.9c)} \end{cases}$$

are now all non-linear in order to further increase the complexity of the considered problem. The one-step Gauss-Seidel-Newton method is again employed with a tolerance of 1×10^{-6}.

The solution obtained is, as expected, qualitatively consistent, but displays a lower value at the peak. As in the linear transient case, the convergence analysis from Figure 5.17 also quickly stagnates at a specific error value which cannot be diminished by neither h- nor p-refinement and, therefore, proceeds from another source. Nevertheless, the problem serves to verify the validity of the method, showing that the partitioned approach is still valid, when incorporating all of the terms from Table 4.4, that are involved in the solution of the split/individual problems: original stiffness and mass terms, derived stiffness and mass terms, and coupling stiffness and mass terms.



(a) $t = 0.0$ **(b)** $t = 0.02$

(c) $t = 0.04$ **(d)** $t = 0.06$

(e) $t = 0.08$ **(f)** $t = 0.1$

Figure 5.16: Total solution color map plots of the nonlinear transient example

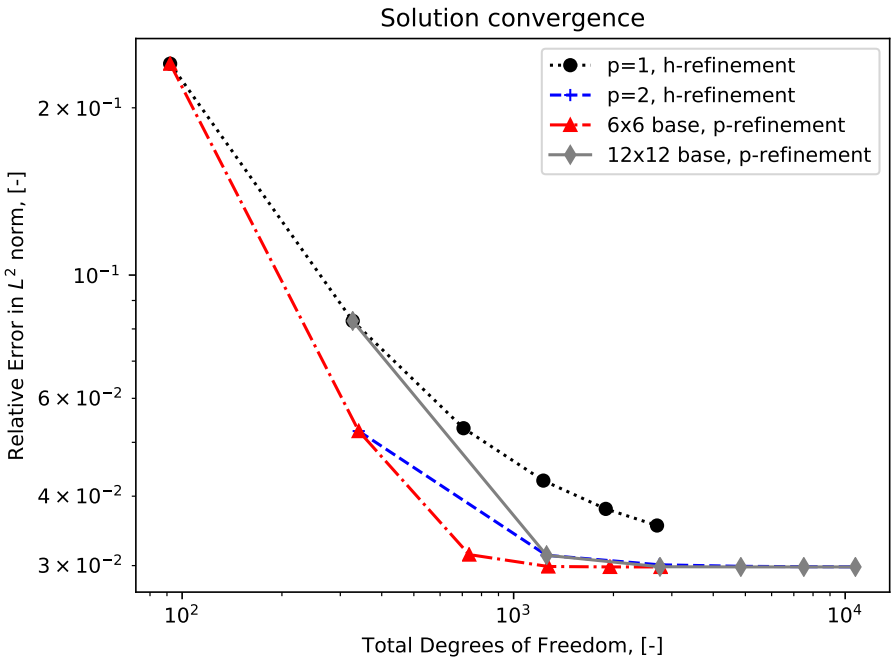


Figure 5.17: Convergence studies for the nonlinear transient example

One-dimensional problem

As for the one-dimensional case, a setup similar to that of Section 5.2 is taken, with the following physical parameters:

$$\begin{cases} \kappa(x,T) = \sin(T) + 10.0 & \text{(5.10a)} \\ \rho(x,T) = \cos(T) + 10.0 & \text{(5.10b)} \\ c(x,T) = \sin(T) + 10.0, & \text{(5.10c)} \end{cases}$$

which lead to the solutions depicted in Figure 5.18 and in the Space-Time domain in Figure 5.19.

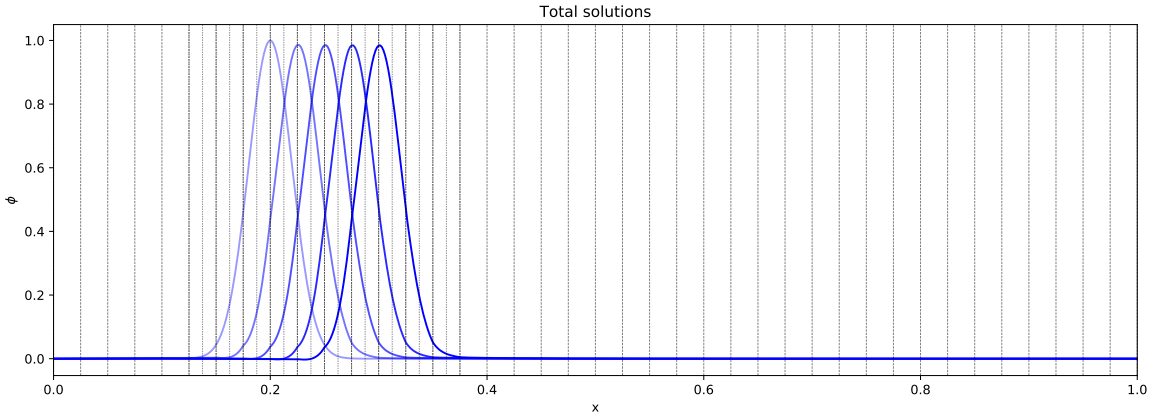


Figure 5.18: Solution over a subset of time steps for the one-dimensional nonlinear transient test case

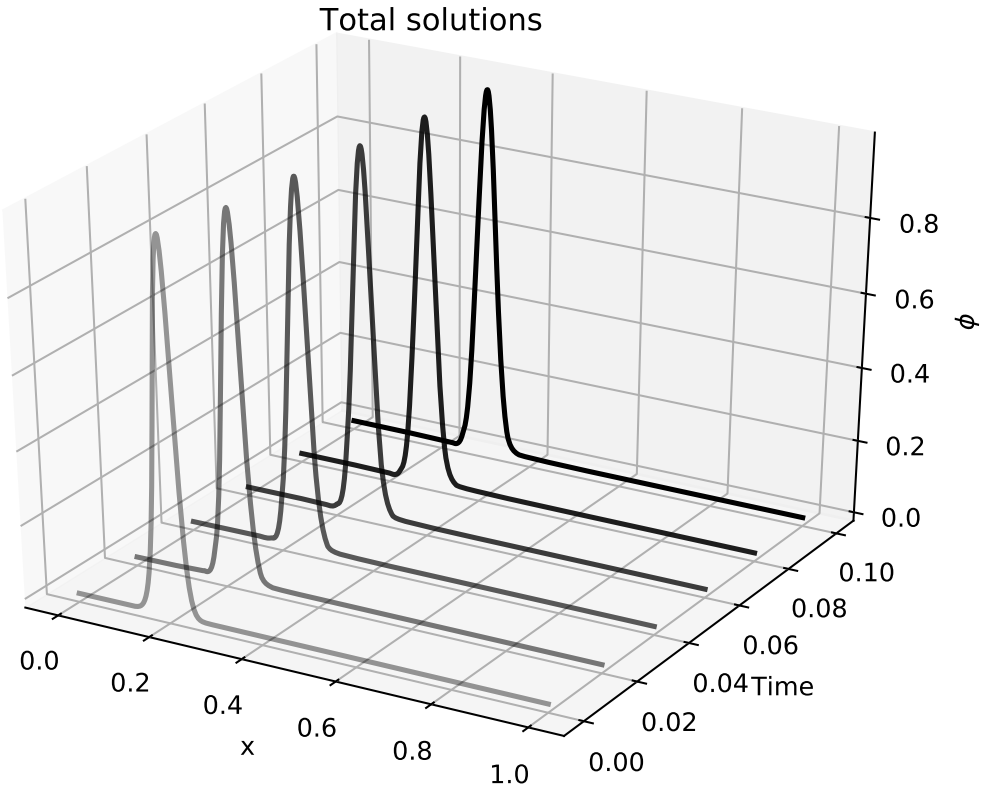


Figure 5.19: Solution over a subset of time steps plotted in space-time for the one-dimensional nonlinear transient test case

5.5 Linear Space-Time

After having verified the applicability of the one-step block Gauss-Seidel-Newton method in the resolution of a transient nonlinear heat transfer problem described with a partitioned multi-level *hp*-FEM through time stepping, the approach is finally also tested for the space-time FEM formulation.

In this case, the problem considered corresponds to the one-dimensional setup described in Section 5.2, which is copied here for convenience.

The manufactured solution corresponds to:

$$T(x, y, t) = \mathbf{e}^{-\left(\frac{(x-t-0.2)^2}{2\times 0.02^2}\right)}, \tag{5.11}$$

and the space-time domain is defined by the box $[0.0, 1.0] \times [0.0, 0.1]$. It is further discretized into 20×10 elements, with a full strip of 4×10 elements being refined, as depicted in Figure 5.20.

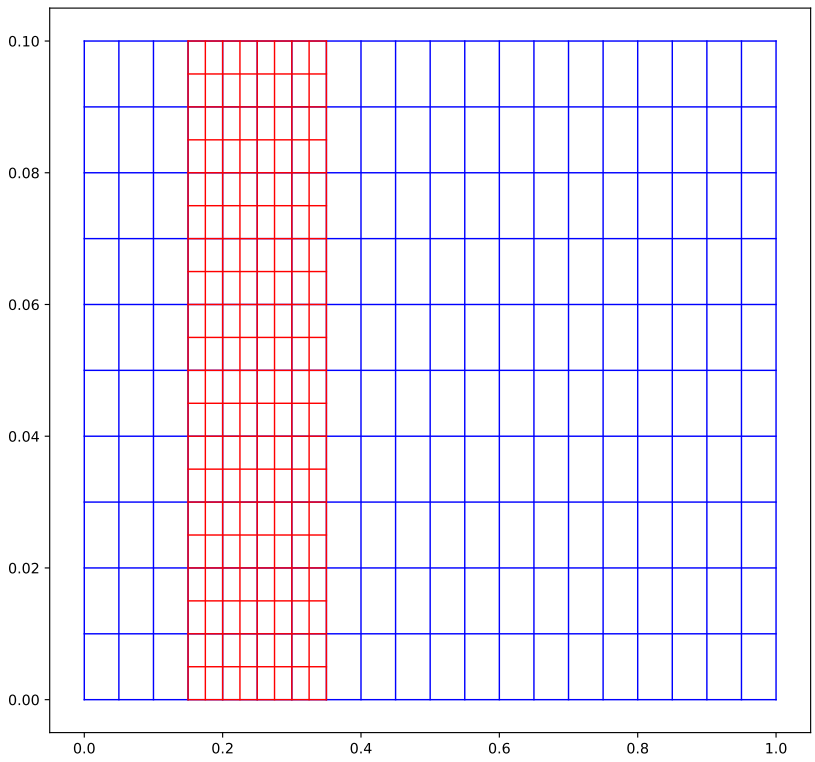


Figure 5.20: Mesh definition for the linear space-time test case

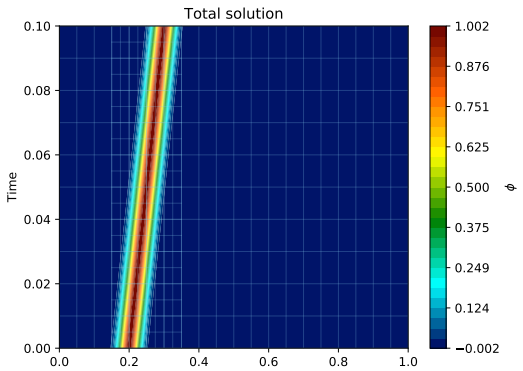


Figure 5.21: Total solution color map of the linear space-time example with Bubnov-Galerkin

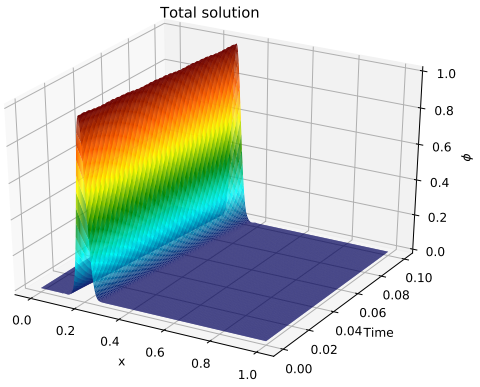


Figure 5.22: Total solution surface plot of the linear space-time test case with Bubnov-Galerkin

The physical parameters, are defined as follows:

$$\begin{cases} \kappa(x) = \sin(x) + 10.0 & \text{(5.12a)} \\ \rho(x) = \cos(x) + 5.0 & \text{(5.12b)} \\ c(x) = 1.0. & \text{(5.12c)} \end{cases}$$

In this case, the lateral boundary is constrained by homogeneous Dirichlet boundary conditions and the initial condition is, naturally, the solution function evaluated at the initial time $t = o$.

In this case, the method considered so far (which applied the Bubnov-Galerkin approach) is able to recover the problem solution, as illustrated by Figures 5.21 and 5.22, even with the limitations discussed in Section 2.4.2. However, further studies were performed with both formulations.

Comparison between Bubnov- and Petrov-Galerkin

Performing a convergence study with the same setup with both the Bubnov-Galerkin and Petrov-Galerkin approaches leads to Figures 5.23 and 5.24, where the expected convergence rates for the different refinements can be recognized. However, in the Bubnov-Galerkin formulation, the solution error oscillates mildly for higher orders. While the behavior is not problematic in itself, it suggests that further analysis be made. In this case, the spectral radius of the system matrices is computed as per Equation 3.3.

In the Bubnov-Galerkin case (Table 5.1), this value is revealed to be near the convergence limit, which also appears to increase the number of iterations required for the solution to convergence. This effect can, however, be diminished either by use of preconditioners or a relaxation parameter (i.e. transitioning into a SOR approach) among others.

In the meantime, the Petrov-Galerkin case (Table 5.2) displays more uniform values across orders and discretizations, with the exception of $p = 2$. For this order, the spectral radius is visibly reduced for one discretization and shoots upwards significantly in the other. In the

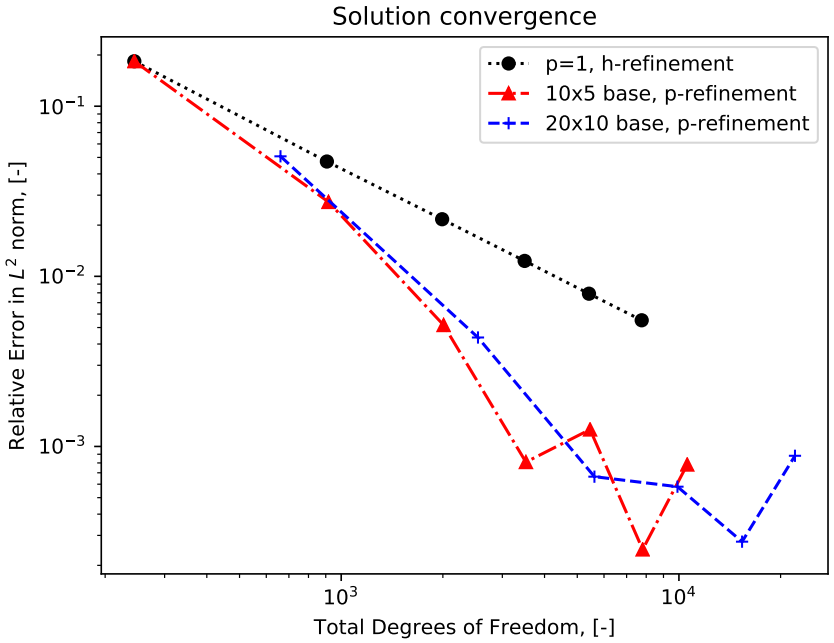


Figure 5.23: Convergence studies for the linear space time example with Bubnov-Galerkin

Table 5.1: Bubnov-Galerkin iterations until convergence and spectral radius results

h :	1	2	3	4	5	6	-
$p = 1$	19	25	25	23	13	18	-
$\rho(\mathbf{H})$	0.786	0.798	0.799	0.799	0.800	0.800	-
p :	1	2	3	4	5	6	7
10×5	19	26	11	36	20	49	30
$\rho(\mathbf{H})$	0.786	0.912	0.939	0.958	0.961	0.971	0.970
20×10	25	28	56	33	58	24	-
$\rho(\mathbf{H})$	0.797	0.922	0.953	0.969	0.975	0.981	-

Table 5.2: Petrov-Galerkin iterations until convergence and spectral radius results

h :	1	2	3	4	5	6	-
$p = 1$	22	22	21	20	19	17	-
$\rho(\mathbf{H})$	0.613	0.650	0.660	0.665	0.668	0.672	-
p :	1	2	3	4	5	6	7
10×5	23	14	19	14	18	14	17
$\rho(\mathbf{H})$	0.624	0.389	0.520	0.418	0.503	0.438	0.495
20×10	21	137	20	14	19	17	-
$\rho(\mathbf{H})$	0.649	0.756	0.585	0.497	0.565	0.514	-

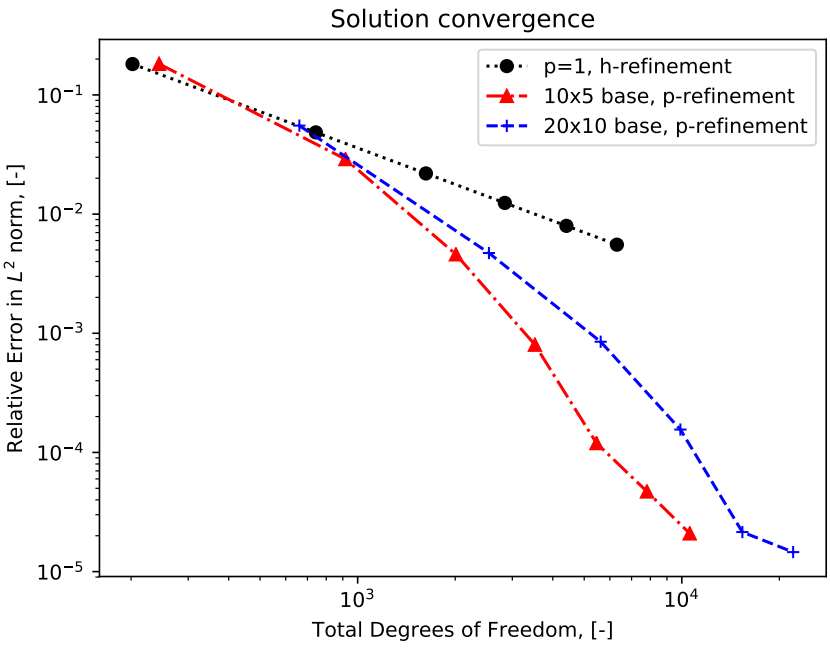


Figure 5.24: Convergence studies for the linear space time example with Petrov-Galerkin

latter, the number of iteration also shoots up by a very high amount. However, further tests seem to suggest that this effect is isolated to orders $p = 2$ and occasionally $p = 4$, while all but vanishing for higher orders.

Comparison to time-stepping

In addition to the internal comparisons between the two formulations of the space-time method, it is also possible to compare the solution obtained with the one from Section 5.2, following a time-stepping approach. In this case, a qualitative assessment of the solutions obtained suggests good agreement between both methodologies as shown in Figure 5.25, which depicts the superposition of the individual curves from the time-stepping approach with the surface obtained in the space-time formulation.

One key difference between the two solutions resides in the interpolation of other intermediate values: while a weighted sum of various time steps may provide a good approximation of the desired values, the same information can be directly extracted from the space-time results and with potentially higher precision. Beyond this observation, it remains to be considered, whether the repeated computation of the lower dimensional system with a finer discretization or the single solution of the coarse higher dimensional system is more expensive. In the linear case, since all matrices can be precomputed outside the iterative loops, the result depends on the overall number of iterations required in each case. Moreover, a time-slab approach can be used in order to partition the work required in the latter scenario.

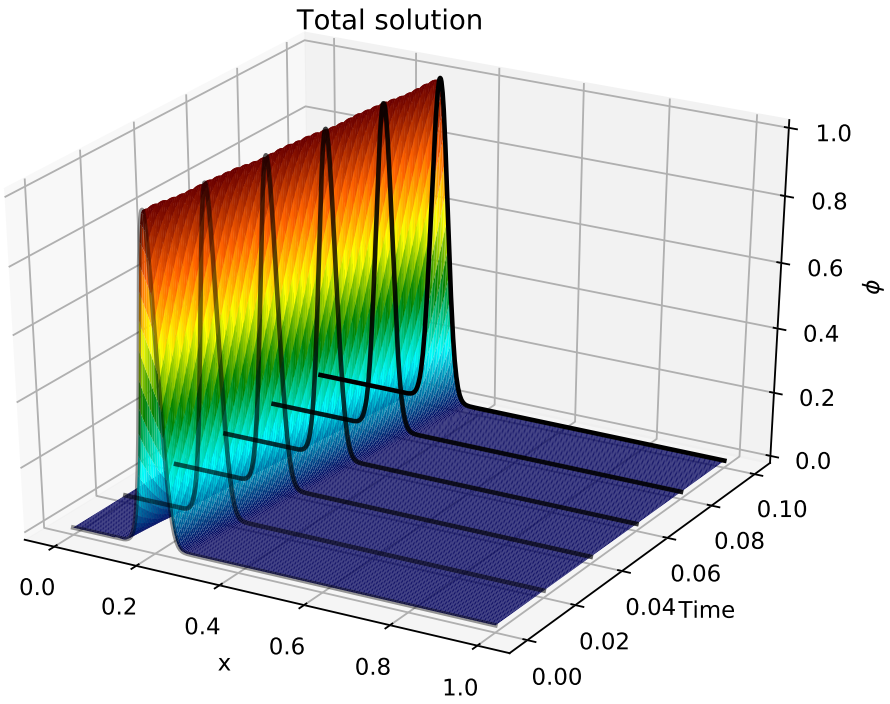


Figure 5.25: Superposition of the linear transient and linear space-time solutions

5.6 Nonlinear Space-Time

The final analyzed case corresponds to the nonlinear space-time formulation. The example considered here is analogous to the one from the linear case, while following the one-dimensional setup described in Section 5.4. Therefore, the physical parameters used are defined as follows:

$$\begin{cases} \kappa(x,T) = \sin(T) + 10.0 & \text{(5.13a)} \\ \rho(x,T) = \cos(T) + 10.0 & \text{(5.13b)} \\ c(x,T) = \sin(T) + 10.0. & \text{(5.13c)} \end{cases}$$

For the defined setup, the Bubnov-Galerkin approach was no longer converging, and indeed its spectral radius was above the threshold of $\rho < 1$. Therefore, alternative discretizations were considered in order to determine whether it was at all possible to apply the formulation to the problem at hand, prior to the application of stabilization techniques. Tables 5.3 and 5.4 summarize the results obtained.

In contrast to the difficulties faced by the Bubnov-Galerkin formulation, which would only converge at particular combinations of polynomial order and overlay discretizations, and Petrov-Galerkin method was convergent for all the cases considered in the tables, with the solution to the original setup illustrated in Figures 5.26 and 5.27. It is, however, possible to encounter rare cases where the the method fails, but it is at the moment unclear what

Table 5.3: Spectral radii for different overlay refinements and orders for a 20×10 base mesh

p	1	2	3	4
2×2 Petrov-Galerkin	0.358	0.209	0.378	0.353
2×2 Bubnov-Galerkin	1.617	1.104	1.040	1.020
3×3 Petrov-Galerkin	0.492	0.313	0.467	0.377
3×3 Bubnov-Galerkin	0.627	1.041	0.901	1.003

Table 5.4: Spectral radii for different overlay refinements and orders for a 20×20 base mesh

p	1	2	3	4
2×2 Petrov-Galerkin	0.496	0.407	0.533	0.547
2×2 Bubnov-Galerkin	1.924	1.101	1.050	1.033
3×3 Petrov-Galerkin	0.620	0.514	0.618	0.552
3×3 Bubnov-Galerkin	0.805	1.049	0.946	1.018

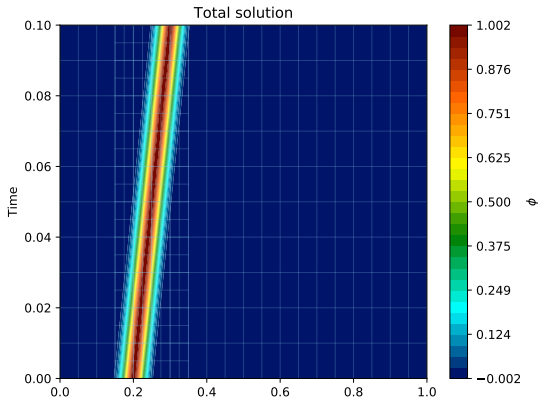


Figure 5.26: Total solution color map of the nonlinear Petrov-Galerkin space-time example

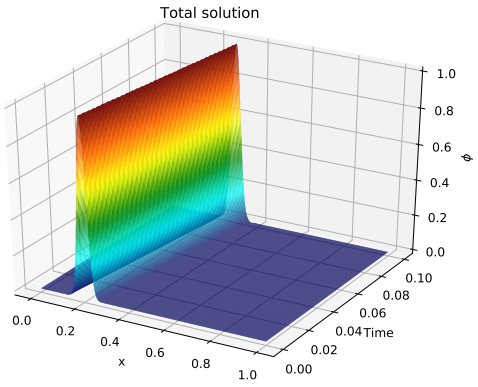


Figure 5.27: Total solution surface plot of the nonlinear Petrov-Galerkin space-time test case

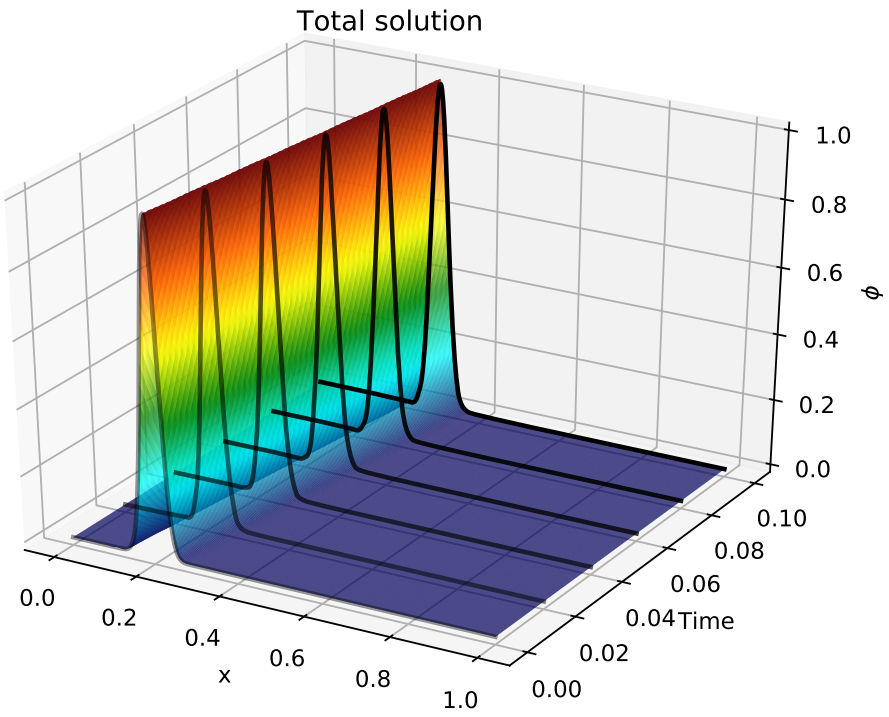


Figure 5.28: Superposition of the nonlinear transient and nonlinear space-time solutions

the exact causes for that are. For the time being, stabilization techniques can be applied to mitigate the potential issues. Needless to say, only a very particular set of illustrative examples were considered and the behavior of the method might vary when considering the whole range of possible problems.

Comparison to time-stepping

One final comparison between the results from the Petrov-Galerkin and the one-dimensional time-stepping solutions was studied, and the superposed results again show a satisfactory agreement between both methodologies (see Figure 5.28). As the system matrices now need to be recomputed after each iteration to account for nonlinearities, the question of whether solving the finely discretized lower dimensional problem or the coarse higher dimensional problem arises again and could be investigated more in-depth.

Chapter 6

Summary and Conclusions

The main goal of the present work was the development of a partitioned hierarchical multi-level space-time *hp*–finite element approach for the solution of nonlinear transient problems. The method aims to diminish the computational costs of solving problems with thermal effects that are local in both space and time, such as those in additive manufacturing. In order to verify the method, a series of incremental examples were devised to evaluate the integration of the individual concepts into a single combined method.

In a first scenario, a regular *p*–FEM method using traditional time-stepping was considered, which served not only as a point of comparison with previous work by Korshunova [18], but extending the partitioned solution strategy to higher order shape functions and two-dimensional problems in conjunction with the multi-level *hp*–method instead of the multiscale $hp-d$. The method was then further extended by the application of the space-time approach, and tested using both the Bubnov-Galerkin and Petrov-Galerkin methods.

First, the linear stationary case was used to verify the applicability of the partitioned approach to the multi-level *hp*–method including its stiffness coupling terms, since the involved systems are in general non-orthogonal. The solution itself was verified against manufactured solutions and monolithic *hp*–solutions from AdhoC++. The higher order shape functions and dimensionality do not affect the performance of the method, and the switch to the multi-level *hp*–refinement is valid.

Following this, the linear transient case was simulated to analyze the influence of the transient effects on the method, such as the additional mass coupling terms. While the time discretization employed was rather coarse, the results obtained match up with those from AdhoC++ and, therefore, it was confirmed that the mass coupling terms were handled properly. Additionally, a 1D case was computed for comparison with the space-time results.

Next, a nonlinear stationary example was used to verify the performance of the partitioned approach in conjunction with a nonlinear solver. Out of the possible strategies, the one-step block-partitioned nonlinear Gauss-Seidel-Newton method was chosen for the test. With again satisfactory results, the scheme was kept for the further nonlinear examples. Convergence results again show algebraic and exponential convergence rates for h- and p-refinements respectively.

The nonlinear transient study combined the previously defined hybrid (nonlinear + parti-

tioned) system solver with the time-stepping approach, again adding mass terms to the equation to solve. The solution was satisfactory when compared to the AdhoC++ results, and a one-dimensional case was computed for comparison with the nonlinear space-time approach.

Finally, introducing the space-time approach affected the stability of the method itself. While the linear problem is still solvable under the defined setup, it was verified that the type of finite element approach (namely Bubnov-Galerkin or Petrov-Galerkin in this case), as well as the choice of partitioned system solvers affect the method's stability. A comparative analysis of Bubnov-Galerkin and Petrov-Galerkin was performed with promising results for the latter. In addition to this, the solution was shown to match the time-stepping solution computed for the linear transient case. The partitioned nonlinear Bubnov-Galerkin space-time method was, however, not always stable at the desired discretizations. Instead, the Petrov-Galerkin formulation was employed with a generally more stable behavior, being capable of computing the defined problems more reliably. Nonetheless, the stability issue is still present for coarser and low even order discretizations, namely $p = 2$ and in some cases $p = 4$. It should be noted, in addition, that the code has not yet been optimized.

While the main goal of the project was successfully attained, close analysis of the resulting method opened a broad outlook. Further investigation is required in the analysis on the method's execution, as well as closer understanding of its stability and convergence criteria. Therefore, its limitations and possible changes which may help to overcome them are to be investigated. Additionally, some possible extensions to the method are:

- usage of error estimators (to adaptively coarsen and refine the simulated space-time domain)
- usage of more general overlays (including variable polynomial orders, higher numbers of overlays, and more general overlay shapes)
- extending the formulation to other combinations of nonlinear and partitioned system solvers
- usage of alternative mathematical models which consider more of the physical processes occurring (e.g. the phase change which takes place in the melt pool and during solidification).

Appendix A

Convergence tables for the individual cases evaluated

Linear stationary heat problem

Table A.1: Linear stationary convergence study and iteration numbers

h :	1	2	3	4	5	6	-
$p = 1$	0.124	9.28E-3	1.95E-3	6.30E-4	2.61E-4	1.27E-4	-
dofs.	52	177	376	649	996	1417	-
iters.	17	19	18	16	15	13	-
$p = 2$	4.72E-3	1.14E-4	1.07E-5	1.98E-6	5.31E-7	1.80E-7	-
dofs.	185	665	1441	2513	3881	5545	-
iters.	26	29	28	25	22	20	-
p :	1	2	3	4	5	6	7
5×5	0.124	4.72E-3	2.59E-4	8.88E-6	5.57E-7	1.20E-8	8.37E-10
dofs.	52	185	392	673	1028	1457	1960
iters.	17	26	38	52	68	82	104
10×10	9.28E-3	1.14E-4	1.31E-6	2.44E-8	2.41E-10	2.36E-12	6.02E-14
dofs.	177	665	1449	2529	3905	5577	7545
iters.	19	29	31	43	63	72	95

Linear transient heat problem

Table A.2: Linear transient convergence study and iteration numbers

h :	1	2	3	4	5	6
$p = 1$	0.364	0.117	0.081	0.069	0.064	0.062
dofs.	59	203	433	749	1151	1639
iters.	27	15	9	8	7	7
$p = 2$	0.084	0.058	0.056	0.056	0.056	0.056
dofs.	213	769	1669	2913	4501	6433
iters.	45	21	14	12	10	9
p :	1	2	3	4	5	6
5×5	0.364	0.084	0.058	0.056	0.056	0.056
dofs.	59	213	453	779	1191	1689
iters.	27	45	55	61	66	70
10×10	0.117	0.058	0.069	0.064	0.062	0.056
dofs.	203	769	1679	2933	4531	6473
iters.	15	21	25	27	28	30

Nonlinear stationary heat problem

Table A.3: Nonlinear stationary convergence study and iteration numbers

h :	1	2	3	4	5	6	7	8	9	-
$p = 1$	0.124	9.17E-3	1.93E-3	6.22E-4	2.57E-4	1.25E-4	6.76E-5	3.97E-5	2.48E-5	-
dofs.	52	177	376	649	996	1417	1912	2481	3124	-
iters.	17	18	18	16	15	13	12	11	11	-
$p = 2$	4.67E-3	1.14E-4	1.06E-5	1.98E-6	5.31E-7	1.80E-7	7.21E-8	3.25E-8	1.61E-8	-
dofs.	185	665	1441	2513	3881	5545	7505	9761	12313	-
iters.	25	29	28	25	22	20	18	16	15	-
p :	1	2	3	4	5	6	7	8	9	10
5×5	0.124	4.67E-3	2.59E-4	8.88E-6	5.55E-7	1.19E-8	8.37E-10	6.83E-12	2.03E-12	2.24E-13
dofs.	52	185	392	673	1028	1457	1960	2537	3188	3913
iters.	17	25	38	54	70	85	108	122	151	164
10×10	9.17E-3	1.14E-4	1.31E-6	2.43E-8	2.41E-10	2.36E-12	5.97E-14	4.14E-15	8.29E-17	2.69E-17
dofs.	177	665	1449	2529	3905	5577	7545	9809	12369	15225
iters.	18	29	34	43	60	71	91	103	124	137

Nonlinear transient heat problem

Table A.4: Nonlinear transient convergence study and iteration numbers

h :	1	2	3	4	5	6
$p = 1$	0.213	0.087	0.057	0.047	0.043	0.040
dofs.	92	327	706	1229	1896	2707
iters.	11	8	8	7	6	6
$p = 2$	0.062	0.036	0.035	0.035	0.035	0.035
dofs.	341	1257	2749	4817	7461	10681
iters.	15	13	11	9	8	7
p :	1	2	3	4	5	6
6×6	0.213	0.062	0.036	0.035	0.035	0.035
dofs.	92	341	734	1271	1952	2777
iters.	11	15	16	17	18	19
12×12	0.087	0.036	0.035	0.035	0.035	0.035
dofs.	327	1257	2763	4845	7503	10737
iters.	8	13	15	16	17	18

Linear space-time heat problem

Table A.5: Linear space-time convergence study and iteration numbers under the Bubnov-Galerkin formulation

h :	1	2	3	4	5	6	-
$p = 1$	0.184	4.73E-2	2.17E-2	1.23E-2	7.90E-3	5.52E-3	-
dofs.	244	907	1990	3493	5416	7759	-
iters.	19	25	25	23	13	18	-
p :	1	2	3	4	5	6	7
10×5	0.184	2.73E-2	5.17E-3	8.09E-4	1.26E-3	2.48E-4	7.82E-4
dofs.	244	917	2010	3523	5456	7809	10582
iters.	19	26	11	36	20	49	30
20×10	5.08E-2	4.37E-3	6.64E-4	5.80E-4	2.76E-4	8.81E-4	-
dofs.	661	2541	5621	9901	15381	22061	-
iters.	8	13	15	16	17	18	-

Table A.6: Linear space-time convergence study and iteration numbers under the Petrov-Galerkin formulation

h :	1	2	3	4	5	6	-
$p = 1$	0.182	4.85E-2	2.19E-2	1.24E-2	7.97E-3	5.54E-3	-
dofs.	202	743	1624	2845	4406	6307	-
iters.	22	22	21	20	19	17	-
p :	1	2	3	4	5	6	7
10×5	0.182	2.89E-2	4.59E-3	8.01E-4	1.20E-4	4.70E-5	2.09E-5
dofs.	244	917	2010	3523	5456	7809	10582
iters.	23	14	19	14	18	14	17
20×10	5.53E-2	4.71E-3	8.49E-4	1.56E-4	2.15E-5	1.46E-5	-
dofs.	661	2541	5621	9901	15381	22061	-
iters.	21	137	20	14	19	17	-

Appendix B

Git repository

The GitLab repository located at:

https://gitlab.lrz.de/ga59vup/HaydenMasterThesis.git

includes the following contents:

- Python multi-level hp code for one- and two-dimensional heat transfer problems
- additional auxiliary scripts which can be relevant to the usage of the code
- this documentation, written in LaTeX together with the respective figures.

List of Figures

List of Tables

Bibliography

[1] Andreev, R. (2012). Space-time discretization of the heat equation. a concise matlab implementation. *arXiv preprint arXiv:1212.6037*.

[2] Bathe, K. (2007). *Finite element procedures*. Prentice Hall, New Jersey.

[3] Bell, J., Berger, M., Saltzman, J., and Welcome, M. (1994). Three dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM Journal of Scientific Computing*, 15:127–138.

[4] Bonet, J. and Wood, R. (2008). *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, Cambridge (UK), New York.

[5] Cottrell, J. A., Hughes, T. J., and Bazilevs, Y. (2009). *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons.

[6] Daneshgar, M., Habibi, S., Daneshgar, E., and Daneshgar, A. (2016). Analysis of residual stresses and angular distortion in stiffened cylindrical shell fillet welds using finite element method. *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, 10(4):538 – 548.

[7] Demkowicz, L. (2006). *Computing with HP-adaptive Finite Elements: One and Two Dimensional Elliptic and Maxwell Problems, Volume 1*.

[8] Düster, A. (2008). High-Order FEM. Lectures notes, Chair for Computation in Engineering, Technische Universität München.

[9] Fish, J. and Markolefas, S. (1993). Adaptive s-method for linear elastostatics. *Computer Methods in Applied Mechanics and Engineering*, 104(3):363–396.

[10] Fries, T., Byfut, A., Alizada, A., Cheng, K. W., and Schröder, A. (2011). Hanging nodes and xfem. *International Journal for Numerical Methods in Engineering*, 86(4-5):404–430.

[11] Huang, S., Liu, P., Mokasdar, A., and Hou, L. (2013). Additive manufacturing and its societal impact: a literature review. *The International Journal of Advanced Manufacturing Technology*, 67(5):1191–1203.

[12] Hughes, T. J. (2012). *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation.

[13] Hughes, T. J. and Hulbert, G. M. (1988). Space-time finite element methods for elastodynamics: formulations and error estimates. *Computer methods in applied mechanics and engineering*, 66(3):339–363.

[14] Hulbert, G. M. and Hughes, T. J. (1990). Space-time finite element methods for second-order hyperbolic equations. *Computer methods in applied mechanics and engineering*, 84(3):327–348.

[15] Hussain, S., Schieweck, F., and Turek, S. (2011). Higher order galerkin time discretizations and fast multigrid solvers for the heat equation. *Journal of Numerical Mathematics*, 19(1):41–61.

[16] Joulaian, M., Hubrich, S., and Düster, A. (2016). Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Computational Mechanics*, 57(6):979–999.

[17] Khairallah, S. A., Anderson, A. T., Rubenchik, A., and King, W. E. (2016). Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones. *Acta Materialia*, 108:36–45.

[18] Korshunova, N. (2016). Partitioned hp-d approach for multiscale transient heat problems. Masterarbeit, Technische Universität München.

[19] Langtangen, H. (2015). Solving nonlinear ode and pde problems. Available at http://hplgit.github.io/num-methods-for-PDEs/doc/pub/nonlin/pdf/nonlin-4print.pdf (2016/09/31).

[20] Larson, M., B. F. (2013). *The Finite Element Method: Theory, Implementation, and Applications*. Springer-Verlag Berlin Heidelberg.

[21] Levy, G., Schindel, R., and Kruth, J. (2003). Rapid manufacturing and rapid tooling with layer manufacturing (lm) technologies, state of the art and future perspectives. *CIRP Annals - Manufacturing Technology*, 52(2):589 – 609.

[22] Lü, L., Fuh, J., and Wong, Y. (2001). *Laser-Induced Materials and Processes for Rapid Prototyping*. Springer US, 1 edition.

[23] Masud, A. and Hughes, T. J. (1997). A space-time galerkin/least-squares finite element formulation of the navier-stokes equations for moving domain problems. *Computer Methods in Applied Mechanics and Engineering*, 146(1-2):91–126.

[24] Melenk, J. M. and Babuška, I. (1996). The partition of unity finite element method: basic theory and applications. *Computer methods in applied mechanics and engineering*, 139(1-4):289–314.

[25] Neugebauer, F., Keller, N., Ploshikhin, V., Feuerhahn, F., and Köhler, H. (2014). Multi scale fem simulation for distortion calculation in additive manufacturing of hardening stainless steel. In *Proceedings of the IWOTE'14: International Workshop on Thermal Forming and Welding Distortion.*, pages 13 – 23. BIAS Verlag.

[26] Ortega, J. and Rheinboldt, W. (2000). *Iterative Solution of Nonlinear Equations in Several Variables*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

[27] Rank, E. (1992a). Adaptive remeshing and h-p domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 101:299–313.

[28] Rank, E. (1992b). A combination of hp-version finite elements and a domain decomposition method. *Proceedings of the First European Conference on Numerical Methods in Engineering*.

[29] Rivara, M. (1984). Mesh refinement processes based on the generalized bisection of simplices. *SIAM Journal on Numerical Analysis*, 21(3):604–613.

[30] Schieweck, F. (2010). A-stable discontinuous galerkin–petrov time discretization of higher order. *Journal of Numerical Mathematics*, 18(1):25–57.

[31] Schillinger, D. and Rank, E. (2011). An unfitted hp-adaptive finite element method based on hierarchical b-splines for interface problems of complex geometry. *Computer Methods in Applied Mechanics and Engineering*, 200(47–48):3358–3380.

[32] Schoinochoritis, B., Chantzis, D., and Salonitis, K. (2017). Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(1):96–117.

[33] Szabo, B. and Babuska, I. (1991). *Finite element analysis*. Wiley, New York. A Wiley-interscience publication.

[34] Yu, J. and Hsu, T. (1985). Analysis of heat conduction in solids by space-time finite element method. *International journal for numerical methods in engineering*, 21(11):2001–2012.

[35] Zander, N., Bog, T., Kollmannsberger, S., Schillinger, D., and Rank, E. (2015). Multi-level hp-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, 55(3):499–517.

[36] Zander, N. D. (2017). *Multi-level hp-FEM: dynamically changing high-order mesh refinement with arbitrary hanging nodes*. Dissertation, Technische Universität München.

[37] Zhang, L. and Michaleris, P. (2004). Investigation of lagrangian and eulerian finite element methods for modeling the laser forming process. *Finite Elem. Anal. Des.*, 40(4):383–405.

[38] Zienkiewicz, O. C., Taylor, R. L., and Zhu, J. Z. (2005). *The finite element method: its basis and fundamentals*. Elsevier.