ulm university universität ulm

HOCHSCHULE RAVENSBURG-WEINGARTEN UNIVERSITY OF APPLIED SCIENCES

Scientific Computing for Multi-Dimensional Nonlinear Heat Conduction and Control of Multiple Heat Sources

Dissertation zur Erlangung des Doktorgrades Dr. rer. nat.
der Fakultät für Mathematik und Wirtschaftswissenschaften der Universität Ulm

vorgelegt von

Stephan Scholz, geboren in Grimma

2025

Amtierender Dekan:	**Prof. Dr. Gunter Löffler**
Gutachter:	**Prof. Dr. rer. nat. Dirk Lebiedz**
Gutachter:	**Prof. Dr.-Ing. Lothar Berger**, Hochschule Ravensburg-Weingarten
Tag der Promotion:	**25.07.2025**

This document was typeset in LaTeX using the tufte-book document class.
The colors blue, green, purple and red, are part of the official logo of the Julia programming language.

Stephan Scholz, Scientific Computing for Multi-Dimensional Nonlinear Heat Conduction and Control of Multiple Heat Sources, Doctoral Thesis, 2025.

Last revision: 05.10.2025

Abstract

In the production and further processing of modern materials and components, such as metal alloys or semiconductor products, thermal energy is applied to the raw material in order to achieve a temperature increase and thus cause a transformation to the end product. This is achieved by means of actuators, such as laser beams or heating elements, which supply a heat flow on the surface of the treated object.

In this dissertation, we create a mathematical model of such thermal processes for rectangular and cuboidal objects. We consider materials with temperature-dependent parameters and anisotropic thermal conductivity. In addition, we treat cooling effects that take place as heat transfer and thermal radiation at the surface. Thus, we obtain a quasilinear heat equation with nonlinear boundary conditions. We approximate this model with the finite volume method in space and we obtain a large system of nonlinear differential equations. We then discuss the special case with constant material parameters, where we obtain a linear state space model, and we present numerical methods for the temporal integration of the differential equations.

Based on the thermal model, we design a concept for heat supply by means of multiple actuators distributed over the surface. We distinguish between two phases in the heat supply. In the first phase, the measured temperatures should follow a predefined reference. To this end, we develop a model-based control system using the theory of differential flatness and numerical optimization. In the second phase, heat is to be continuously tracked by means of a control system in order to compensate for thermal losses and to keep the measured values at the reference value. Here we take up known approaches from linear-quadratic and model predictive control and we adapt them for our thermal model. Finally, we demonstrate the methods presented in two comprehensive examples.

As part of this work, the author developed the two software packages “Hestia.jl” and “BellBruno.jl” in the Julia programming language and made them freely available. We use “Hestia.jl” to create thermal models in code as differential equations and “BellBruno.jl” to calculate the derivatives of the reference signal for the flatness-based control.

Kurzbeschreibung

Bei der Produktion und weiteren Verarbeitung moderner Werkstoffe und Komponenten, wie Metalllegierungen oder Halbleiterprodukten, wird dem Ausgangsmaterial thermische Energie von außen gezielt zugeführt um eine Temperaturerhöhung zu erreichen und somit eine Transformation hin zum Endprodukt hervorzurufen. Dies geschieht mittels Aktuatoren, wie zum Beispiel Laserstrahlen oder Heizelementen, die einen Wärmefluss an der Oberfläche des behandelten Objektes einbringen.

In dieser Dissertation erstellen wir ein mathematisches Modell solcher thermischen Prozesse für rechteckige und quaderförmige Objekte. Dabei betrachten wir Materialien mit temperaturabhängigen Parametern und anisotroper Wärmeleitung. Außerdem behandeln wir Kühlungseffekte, die als Wärmeübergang und Wärmestrahlung an der Oberfläche stattfinden. Somit erhalten wir eine quasilineare Wärmeleitungsgleichung mit nichtlinearen Randbedingungen. Wir approximieren dieses Modell mit dem Verfahren der finiten Volumen im Raum und erhalten ein großes System nichtlinearer Differentialgleichungen. Anschließend besprechen wir den Spezialfall bei konstanten Materialparametern, bei dem wir ein lineares Zustandsraummodell erhalten, und wir stellen numerische Verfahren zur zeitlichen Integration der Differentialgleichungen vor.

Aufbauend auf dem thermischen Modell entwerfen wir ein Konzept zur Wärmezufuhr mittels einem oder mehreren Aktuatoren, die verteilt auf der Oberfläche wirken. Bei der Wärmezufuhr unterscheiden wir zeitlich zwei Phasen. In der ersten Phase sollen die gemessenen Temperaturen einer vordefinierten Referenz folgen. Dafür entwickeln wir, mit Hilfe der Theorie der differentiellen Flachheit und der numerischen Optimierung, eine modellbasierte Steuerung. In der zweiten Phase soll mittels einer Regelung stetig Wärme nachgeführt werden, um thermische Verluste zu kompensieren und um die Messwerte an dem Referenzwert zu halten. Hier greifen wir bekannte Ansätze aus der linear-quadratischen und modellprädiktiven Regelung auf und passen diese für unser thermisches Modell an. Abschließend demonstrieren wir die vorgestellten Verfahren an Hand von zwei umfassenden Beispielen.

Im Rahmen dieser Arbeit wurden die beiden Softwarepakete „Hestia.jl“ und „BellBruno.jl“ in der Programmiersprache Julia entwickelt und frei zur Verfügung gestellt. Wir nutzen „Hestia.jl“ um thermische Modelle als Differentialgleichung zu erstellen und mit „BellBruno.jl“ berechnen wir die Ableitungen des Referenzsignals für die flachheitsbasierte Steuerung.

Table of Contents

Symbols and Units

Units

Second	s
Meter	m
Centimeter	cm
Kelvin	K
Kilogram	kg
Watt	W

Symbols

Geometry

Number of dimensions	N_d	$\{1,2,3\}$	
Length	L	$\mathbb{R}_{>0}$	in $[m]$
Width	W	$\mathbb{R}_{>0}$	in $[m]$
Height	H	$\mathbb{R}_{>0}$	in $[m]$
Rod	Ω_1	$(0,L)$	in $[m]$
Rectangle	in Ω_2	$(0,L)\times(0,W)$	in $[m^2]$
Cuboid	Ω_3	$(0,L)\times(0,W)\times(0,H)$	in $[m^3]$
Boundary	$\partial\Omega_i := \overline{\Omega}_i \setminus \Omega_i$		
Actuator Boundary	$B_{in} \subseteq \partial\Omega$		
Sensor Boundary	$B_{out} \subseteq \partial\Omega$		

Material

Volumetric mass density	ρ	$\mathbb{R}_{>0}$	in $\left[\frac{kg}{m^3}\right]$
Specific heat capacity	c	$\mathbb{R}_{>0}$	in $\left[\frac{J}{kg\ K}\right]$
Thermal conductivity	λ	$\mathbb{R}_{>0}$	in $\left[\frac{W}{m\ K}\right]$

Heat Equation

Position	x	$\overline{\Omega}_i$	in $[m]$, $[m^2]$, $[m^3]$
Final time	T_{final}	$\mathbb{R}_{>0}$	in $[s]$
Time	t	$[0, T_{final}]$	in $[s]$
Temperature	θ	$\mathbb{R}_{>0}$	in $[K]$
Temperature distribution	ϑ	$[0, T_{final}] \times \Omega$	in $[K]$
Ambient temperature	ϑ_{amb}	$[0, T_{final}] \times \Omega$	in $[K]$
Initial temperature	ϑ_0	Ω	in $[K]$

Boundary Conditions

Total heat flux	ϕ	$[0, T_{final}] \times \partial\Omega$	in $\left[\frac{W}{m^2}\right]$
Supplied heat flux	ϕ_{in}	$[0, T_{final}] \times B_{in}$	in $\left[\frac{W}{m^2}\right]$
Emitted heat flux	ϕ_{em}	$[0, T_{final}] \times \partial\Omega$	in $\left[\frac{W}{m^2}\right]$
Supplied power	P_{in}	$[0, T_{final}]$	in $[W]$
Emitted power	P_{em}	$[0, T_{final}]$	in $[W]$
Heat transfer coefficient	h	$\mathbb{R}_{>0}$	in $\left[\frac{W}{m^2\,K}\right]$
Emissivity	ϵ	$\mathbb{R}_{>0}$	
Stefan-Boltzmann constant	$\sigma \approx 5.67 \cdot 10^{-8}$		in $\left[\frac{W}{m^2\,K^4}\right]$

Remark: The unit of the heat flux and related quantities are specified in physics for a three-dimensional object, here a cuboid Ω_3.

Spatial Approximation

Spatial sampling	$\Delta x_1, \Delta x_2, \Delta x_3$		in $[m]$
Number of cells along x_1	N_j		
Number of cells along x_2	N_m		
Number of cells along x_3	N_k		
Total number of cells	$N_c = N_j \cdot N_m \cdot N_m$		
Local indices in x_1, x_2, x_3	j, m, k		
Global index	i as in Eq. (3.7)		
Discrete temperatures	$\Theta = (\Theta_1, \ldots, \Theta_{N_c})^\top$	$[0, T_{final}]$	in $[K]$
Diffusion matrix	D_1, D_2, D_3		
Emission matrix	E_1, E_2, E_3		
System matrix	A_1, A_2, A_3		

Control Design

Number of actuators	N_u		
Number of sensors	N_y		
Actuator's spatial characteristics	b as in Eq. (6.3)		in $\left[\frac{1}{m^2}\right]$
Sensor's spatial char.	g as in Eq. (6.5)		in $\left[\frac{1}{m^2}\right]$
Input signals	u		in $[W]$
Output signals or temperature measurements	y as in Eq. (6.6)		in $[K]$
Reference signal	r		in $[K]$
State & output weighing matrix	Q	$\mathbb{R}^{N_c \times N_c}$ or $\mathbb{R}^{N_y \times N_y}$	
Input weighing matrix	R	$\mathbb{R}^{N_u \times N_u}$	

Prologue

1

Introduction

"Marzenia zawsze zwyciężą rzeczywistość, gdy im na to pozwolić."

"A dream will always triumph over reality, once it is given a chance."

– Stanisław Lem

Heat conduction is an essential physical process, which describes the transfer of thermal energy in a medium like gas, liquid or solid. The remarkable feature of this process is that the material does not move itself on a macroscopic level, as in case of advection or convection. Energy is only transferred via microscopic activity, e.g. oscillation, and interaction of atoms and molecules. The state of this particle activity is quantified by the temperature: a low value means less and a high value means intensive activity. The temperature is denoted in the units Kelvin, Celsius and Fahrenheit (United States of America), where Kelvin and degree Celsius are SI units, see [1, page 133]. One may say that zero Kelvin corresponds to a physical state at which no particle activity is present. At approximately 273.16 Kelvin (or zero degree Celsius), we have the triple point of water, which is also known as ice point. At this point, all three phases of water (gas, liquid and solid or ice) are in a thermodynamical equilibrium state at atmospheric pressure of approx. 101.325 Pascal, see triple point in [2]. In several of our examples, we assume an ambient temperate of 300 Kelvin or approx. 27° Celsius, which is in a suitable range of the room temperature in Germany.

In this thesis, we consider heat conduction in a solid with a cubic geometry, e.g. a one-dimensional rod, a two-dim. rectangle or a three-dim. cuboid. The one- and two-dim. geometries do not exist in reality but they approximate physical phenomena and they simplify their analysis, simulation and control. We do not specify the solid material, but we assume in our examples metals like aluminum or iron and mixtures of metals or alloys like steel. Heat conduction is described mathematically by the variation of temperature in time and space in form of the heat equation. The standard heat equation is a linear partial differential equation (PDE)[1] with one first order derivative in time and second order derivatives in space. Due to its simplicity, it is an elementary example for the analysis and numerical simulation of PDE, see [3, p. 44] and [4, p. 75]. The PDE only describes the spatio-temporal dynamical behavior inside an object. Additionally, we need to specify the data along all boundary sides for a com-

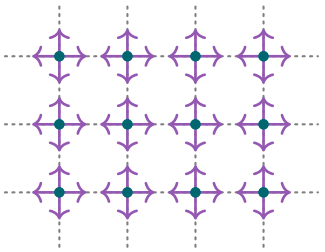
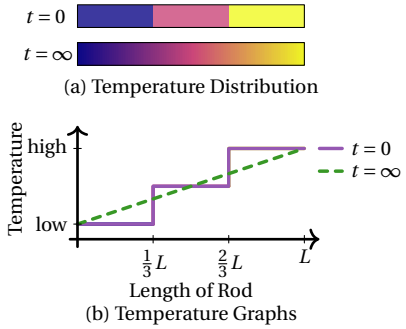


Figure 1.1: Microscopic model of oscillating solid particles in a crystalline grid. The stronger the oscillations the higher is the thermal energy. Thermal energy is transferred via collisions of multiple particles.

[1] We denote the singular and plural form (equation/equations) as PDE.

plete problem formulation. In the analysis of PDE, we distinguish Dirichlet and Neumann-type boundary conditions[2], where the first one fixes the data and the latter one defines a spatial gradient. In case of the heat conduction, the Dirichlet boundary data is a constant or time-varying temperature value and the Neumann boundary condition represents a heat flux, which goes inwards or outwards the object. Dirichlet boundary conditions are easier to understand and implement because they affect the thermal dynamics explicitly. For example: if both sides of a one-dim. rod have a fixed temperature, e.g. low value on the left side and a high value on the right side, then we know that the temperatures inside the rod converge to values between the low and high value on both boundaries, see Fig. 1.2. Dirichlet boundary conditions do not suit for our purposes in this thesis because we are interested in thermal interaction of the object with its surrounding. This interaction is realized via Neumann boundary conditions in form of a heat flux and this exchange of thermal energy along the boundary sides results in a cooling-down or heating-up procedure.

[2] In the literature one may also find Robin boundary conditions, which combine the Dirichlet and Neumann type.



(a) Temperature Distribution

(b) Temperature Graphs

Figure 1.2: Example temperature distribution in one-dim. rod with Dirichlet boundary conditions. The temperature on the left boundary is fixed at a low temperature and the right boundary has a high value.

In the first part of this thesis, we create a numerical model to simulate heat conduction including thermal emissions, which cause a cooling. In the second part, we design a control system to heat up the object such that its surface reaches a desired temperature. On one hand, we supply thermal energy via actuators, like heating elements, to increase the object's temperature and on the other hand we have convective and radiative emissions towards the surrounding, which disturb our control aims. This general concept is embedded in a framework that enables several design options for the geometry and material of the object, the interaction with the surrounding, and the setup of actuators and sensors.

As heat conduction is a very wide field of research with many applications, we present two examples in the subsequent sections: laser welding and semiconductor fabrication. We select these applications because the physical modeling and the considered control approaches may (partially) fit to our proposed heat conduction framework. So, we describe the connections and the differences between these examples and our framework.

1.1 Laser Welding

Laser welding is a central processing step in the production of modern materials and components because it enables us to create objects with complex shapes. The technical procedure of welding is the melting and subsequent solidification of material at the interface of adjacent objects. In other words, the treated material changes its phase from solid to liquid and the resulting weld pool connects the interface of both objects. This procedure is well analyzed in research in order to understand the thermal dynamics and to avoid material fatigue along the weld seam, see [5–7].

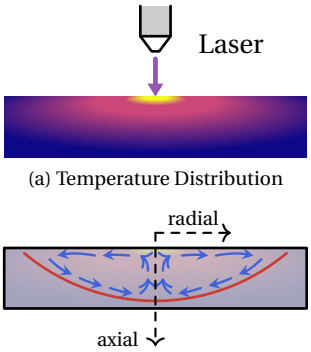
In the subsequent paragraph, we briefly describe the thermal behavior in a single-spot *pulsed* laser welding process according to the article [8] and doctoral thesis [9]. The laser supplies a constant amount of power on a single spot in a short time interval, e.g. 1 to 50 milliseconds [9, page 33]. The temperature at this spot increases and the material changes its phase at the solidus temperature to a mixture of partially solid and liquid. While

the laser treatment continues, the phase at the welding spot changes completely to liquid and the resulting weld pool is growing. When the desired weld pool size is reached (after 1 to 50 ms), the laser intensity is decreased until it is shut down. Thermal conduction and loss via heat transfer and radiation to the environment force the weld pool to cool down and its phase transits back from liquid to solid. We depict the temperature distribution of a laser welding example in Fig. 1.3 (a).

In this laser welding procedure, we find a few physical processes, which we model and simulate in this thesis, too. As the laser supplies a high amount of power to change the solid state into a weld pool, the treated material reaches very high temperature values, e.g. 1000 Kelvin in case of a specific class of aluminum alloys, see [8]. We even find higher temperatures for other materials, e.g. in [5]. These high temperatures lead to intensive thermal emissions via convective and radiative energy transfer. In Section 2.5, we model these emissions and we find the heat radiation as a nonlinear boundary condition. When the solid material turns into a (partially) liquid medium, then a circular convective heat transfer occurs inside the liquid weld pool as depicted in Fig. 1.3 (b), see [9, p. 78]. It moves material from the weld pool center towards its boundary in radial direction, then to the bottom and back. A convection can be modeled as transport equation, see [4, p. 6, 7], but the authors of [8, 9] avoid such a temperature-depending switching of the system model from a pure heat equation to a heat and transport equation.[3] Instead, they consider an anisotropic thermal conductivity, which means that heat transfer operates better towards the radial than axial direction. Furthermore, the phase transition is modeled with material properties, which are designed as functions of the temperature. So, we find in article [8, Fig. 2] a significant difference in the thermal conductivity between the solid and liquid phase. This fact is also noted in [9, p. 76 in Fig. 5.16, p. 78 in Fig. 5.19].

In Section 2.2, we propose an anisotropic thermal conductivity and temperature-dependent material properties, but we do not explicitly consider a phase transition.

Regarding the control of laser welding, we have one laser in article [8] and we can only evaluate the results after its operation. We do not measure the temperature or the weld pool size during the laser treatment and so we cannot apply a feedback controller. Instead, we need to design a feed-forward control approach, which computes the proper input signal based on the full knowledge of the thermodynamical model. In article [8], the authors compute a feed-forward control algorithm with optimization techniques. They formulate and implement an optimal control problem for a shut-down operation of the laser and solve it numerically. Numerical optimization approaches offer a wide range of options, e.g. various norms and hyperparameters, to design the control problem. Thus, we need to evaluate the found input signal and the resulting simulation data to guarantee a proper operation. We can only apply the computed input signal on the real system, here a laser, if the treated system really behaves as desired. Otherwise, we need to recalibrate the optimization options and restart the optimization routine as depicted in Fig. 1.4. Moreover, numerical optimal control is a computationally costly approach, in particular for systems de-



(a) Temperature Distribution

(b) Convective Heat Transfer

Figure 1.3: Visualization of a laser welding example. The temperature distribution in (a) shows a temperature gradient from the hot weld spot towards the cold regions in radial and axial direction. The blue arrows in (b) symbolize a circular convective heat transfer inside the weld pool according to [9, page 78].

[3] As the heat equation is also known as diffusion equation, a model with heat and transport is called diffusion-convection equation.

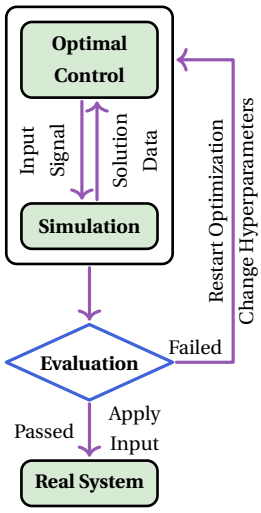


Figure 1.4: Procedure of finding an optimal control approach.

scribed by PDE because it solves an optimization problem iteratively for each time step of the sampled temporal dynamics. In case of the heat equation, we approximate the object in space and sample the time to yield a temperature value for each spatial grid node and time step. Depending on the problem complexity and hardware equipment this cooperation of numerical simulation and optimization of the thermal dynamics may be computationally costly and take a long time.

In Chapter 7, we derive the input signal in two steps to avoid such high computational costs. In a first step, we derive an analytical feed-forward control approach for a simplified heat conduction model, which is close to an applicable solution. In the second step, we transfer the analytical input signal to an optimization-based control approach for the realistic heat conduction problem and solve it.

1.2 Semiconductor Fabrication

In the second application, we present heat conduction scenarios in semiconductor fabrication to produce electronic components like integrated circuits. This technology consists of several complex, highly precise and clean processing steps. Hence, we are not able to describe all thermal treatments, but we select three processes: crystal growth, lithography and rapid thermal processing, which represent a specific thermal treatment and dynamics.

In the first step of semiconductor fabrication, a single crystal in form of a cylinder is produced and afterwards cut into disks. These disks are called wafers and they are treated in subsequent steps physically and chemically in order to establish electrical circuits on a very small scale.

Crystal Growth

In crystal growth the semiconductor raw material is thermally treated in a crucible to yield a single crystal. Here, we briefly discuss the Vertical-Gradient Freeze (VGF) method, where we have very high temperatures, e.g. above 1000 Kelvin, and a phase transition, liquid to solid, similar to the welding example in Section 1.1. This VGF method and its control approaches are described in the articles [10, 11], in the book [12, p. 3] and in the doctoral thesis [13]. In these contributions, the authors describe a plant with heaters on the bottom, on the top and on the jacket of the crucible. These heaters specify a desired temperature gradient in the melt such that the single crystal grows from the bottom to the top, as depicted in Fig. 1.5. This heating process is steered with flatness-based control, see [11], [12, p. 7] and [13, p. 60], and model predictive control in [10].

In this thesis, we cover both approaches, because they are well-known representatives of open-loop and closed-loop control methods. In Chapter 7, we introduce the flatness-based control and we discuss its application for the prototyping of a feed-forward control system. In Section 8.2, we describe the model predictive control for our heat conduction model framework.

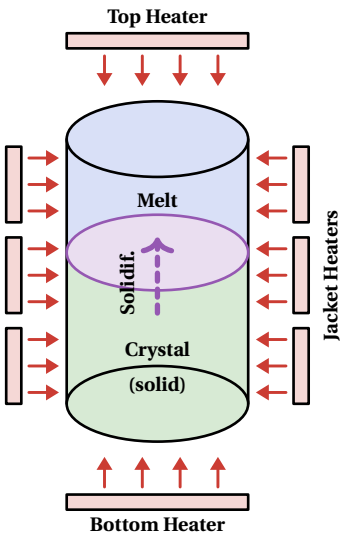
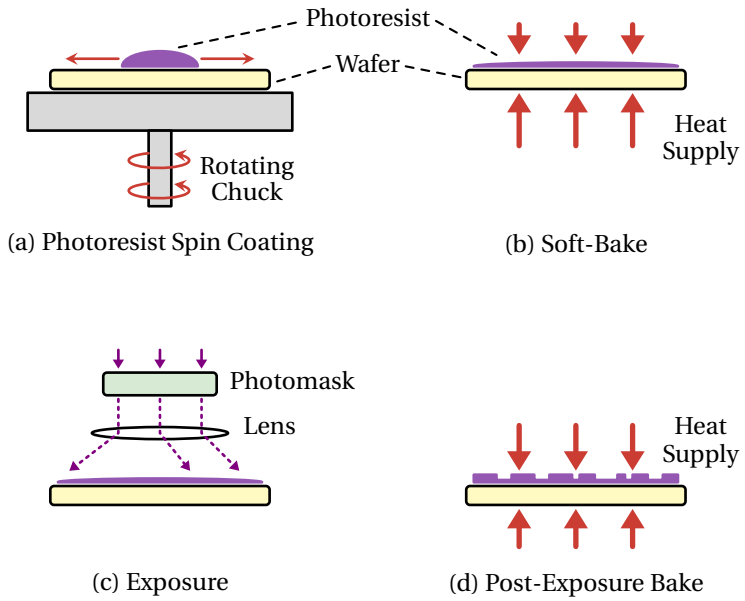


Figure 1.5: Model of a Vertical-Gradient Freeze process according to [10, Fig. 2]. Heaters supply thermal energy to steer the solidification of a melt from bottom to top.



(a) Photoresist Spin Coating (b) Soft-Bake

(c) Exposure (d) Post-Exposure Bake

Figure 1.6: A selection of first processing steps in lithography. The wafer is cleaned and positioned on a rotating chuck. A liquid photoresist is applied as a drop on top of the wafer and the rotation distributes the liquid uniformly in (a) according to [18]. The liquid photoresist is heated to solidify at ca. 100° Celsius in (b). A high energy radiation is guided through a photomask and lens to expose predefined patterns in the photoresist in (c) according to [15, p. 4, Fig. 1.2]. The treated wafer with photoresist are baked at ca. 100° Celsius again in (d) to prepare them for the further processing steps.

Lithography

The produced single crystal is sliced and cleaned for the further processing steps of lithography. Lithography is one of the core technologies to convert a (silicon) wafer into integrated circuits, e.g. microelectronic components. The wafer topside is coated with a photoresist, which is exposed by radiation. The type of lithography is distinguished by the radiation: for example ultraviolet light in photolithography, electron beams or ion beams in charged-particle lithography [14, p. 139]. In this manner, patterns of a photomask are transferred on the photoresist and the resulting prototype pattern structure is treated in subsequent processing steps like the insertion of ions or other material, and etching, see [15, p. 2]. We have baking procedures after the coating (soft bake) to evaporate the solvent from the photoresist, and after the exposure (post-exposure bake) to trigger chemical reactions in the photoresist at the exposed zones, see [16]. These first processing steps of lithography are depicted in Fig. 1.6, further information about it is noted in article [16] and in the books [15, p. 1] and [17, p. 2].

The initial manufacturing steps of wafers and photomasks are similiar. A substrate made of (quartz) glass is coated with resist, followed by patterning with an electron beam, a post-exposure bake and subsequent processes like etching and cleaning, see [19, p. 7] and [20]. Hence, we denote wafers and photomasks subsequently in the general term as substrate. For the post-exposure bake (PEB), the substrate with resist is placed on top of a metal plate with multiple controllable heating zones.[4] In the literature, we find cylindrical forms in [16, 21] and cubic plate shapes in [22–24], where the first one is rather used for wafers and the last one for photomasks. The substrate may be placed on pins, which separate it from the heating plate in close proximity. Additionally, the substrate and heating plate are covered with a lid on top to avoid thermal losses and external disturbances, see patent [24] and datasheet [25].

[4] This plate is called hotplate, heating plate or bake plate.

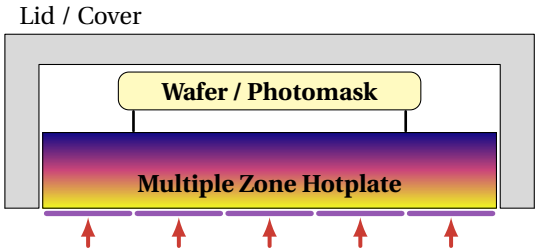


Figure 1.7: Simplified side view of a multiple zone hotplate with wafer or photomask during the post-exposure bake, see [24, Fig. 2]. The wafer or photomask is positioned with pins on the hotplate. Heat is supplied via multiple heating elements on the underside. A metal cover captures the thermal energy inside and avoids disturbances like air fluctuations.

Regarding the heating process, multiple heating elements on the plate's underside supply thermal energy. This heat conducts through the plate and is transferred further towards the substrate. Modern baking devices can be operated up to 230° Celsius [25], whereas we find in the literature PEB temperatures of 95° up to 150° Celsius, see [16, 23] and [21]. We remark that the substrate's temperature shall be steered in this process and we know that the heat transfer from plate to substrate depends on the distance between both objects. As the heating plate with lid is completely closed, one may assume that the substrate reaches the plate's temperature after some time. However, the bake shall operate quick and advantageous to save energy and guarantee a well treatment of the substrate and resist. Hence, a well-performing control system is necessary to steer the thermal process accordingly. Here, we find the issue that temperature sensors are located inside the plate [16, 26]. Thus, only the plate temperature can be measured during the real operation. This problem can be solved in the development and test of the heating plate using a sensor mask instead of a substrate. The sensor mask measures the temperature at several distributed points, e.g. with PT1000 elements as in [22, 23], and transmits the data via a cable to a computer as depicted in Fig. 1.8.

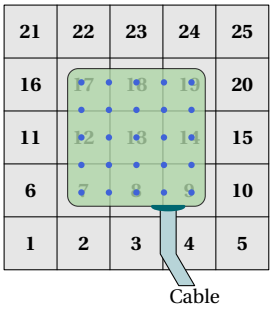


Figure 1.8: Topview of a multiple zone heating plate with sensor photomask on top according to [23]. The 25 zones visualize the heating elements on the hotplate's underside. The sensor mask with 25 PT1000 sensors (blue circles) is mounted on top of the hotplate for calibration, see [23].

In the literature, we find that the thermal dynamics of a heating plate is modeled as linear differential equations, see e.g. [27] and [28, p. 18, 19]. These models are derived via an approximation of the heat conduction using electrical circuit analog models, where the thermal resistance and capacity are replaced by the electrical quantities. Such an electrical circuit model of a heating plate with three segments is exemplified in Fig. 1.9 according to the doctoral thesis [28, p. 19]. In such models, electrical currents describe heat fluxes and voltages correspond to temperature differences. On one hand these simplified models offer an intuitive way to design modern control approaches, e.g. model predictive control in the article [16, 27, 29, 30], but on the other hand they reduce the entire spatial thermal dynamics to a single temperature value, which behaves like the charging and discharging of an electrical capacitor. The latter statement might be explained by the fact that only single, isolated, sensors are installed inside the heating plates. Hence, we are only able to measure isolated temperature points - in contrast to distributed measurements with e.g. thermal imaging. Regarding the control design of heating plates, we also find standard PID control approaches, in which the parameters are found numerically in real experiments using sensor photomasks as depicted in Fig. 1.8, see [22, 23].

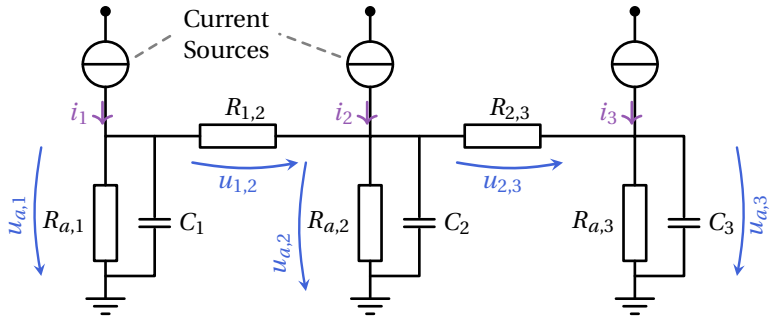


Figure 1.9: Analog electrical circuit model of a cylindrical heating plate consisting of three segments according to [28, p. 19]. The current sources exemplify the heating elements, the electrical currents i corresponds to the heat flux and the voltages u relate to temperatures. The voltages $u_{1,2}$ and $u_{2,3}$ stand for the temperature difference between the plate segments, and $u_{a,1},\dots,u_{a,3}$ are temperature differences between the plate and its surrounding.

In this thesis, we take up various aspects of PEB, but we introduce and treat them in the light of spatially distributed thermal dynamics. In Chapter 2, we describe the shape as a cuboid, where the temporal and spatial temperature evolution takes place. The thermal losses, which are approached as currents through resistances $R_{a,n}$ in Fig. 1.9, are modeled in Sections 2.4 and 2.5 as boundary conditions of the heat equation using heat transfer and heat radiation. Similarly, we describe the heat supply via multiple spatially distributed heating elements in Section 6.1 as heat fluxes. In contrast to the described temperature sensors inside the heating plate, we assume temperature measurements only on the surface of the cuboid. This idea also corresponds to the measurement using a sensor photomask as depicted in Fig. 1.8. Since the modelling and simulation of the spatially distributed thermal dynamics is significantly more complex than a small system of linear differential equations, we focus primarily on feed-forward control in Chapter 7 to design the control system. Subsequently, a predictive feedback control approach is intended in Chapter 8 to stabilize the measured temperature at the reference value while compensating thermal losses.

Rapid Thermal Processing

Finally, we take a look at rapid thermal processing (RTP) to heal defects in the crystal structure of wafers, which are caused by ion implementation, see e.g. [31, p. 316] and [32, p. 5]. In this process, the wafer is heated up quickly for a short time and cooled down afterwards via thermal emission, e.g. convection and radiation. There are different designs of RTP systems, see [31, p. 317, Fig. 31.2]. They have in common that powerful lamps, e.g. (tungsten) halogen lamps [32, p. 9], supply a high amount of thermal energy to the treated wafer and a pyrometer measures its temperature. In Fig. 1.10, we depict a simplified version of one possible RTP design. The wafer reaches very high temperatures, e.g. 1000 Kelvin, but only for a couple of seconds, see [33] and [31, p. 318]. Such high temperatures force the material properties to change. Hence, the thermal conductivity and the heat capacity are modeled in article [33] as functions of the temperature.[5] This fact underlines our temperature-dependent modeling of the material properties in Section 2.2.

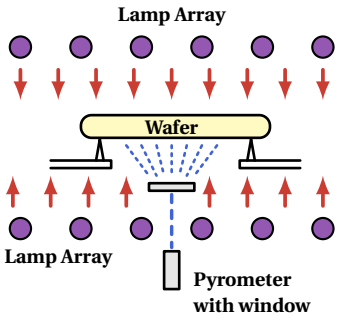


Figure 1.10: Simplified side view of rapid thermal processing according to [31, p. 317, Fig. 31.2 (b)] and [32, p. 10, Fig. 1.2].

[5] In Section 1.1 we noted the temperature-dependent material properties in the laser welding modeling.

1.3 Contribution and Outline

The previous application examples provide a brief overview about the wide field of research in thermal process engineering. As we are not able to cover all these interesting research topics, we choose a few ideas in the domain of modeling, simulation and control of thermal problems and we gather and arrange them in a preferably general and practical framework. In the subsequent paragraphs, we outline the topics in each chapter and we state the scientific contribution including the previously published articles.

Thesis Outline

This thesis is divided into two main parts: firstly modeling and simulation of the heat conduction, and secondly the control design of the heating-up procedure. In the first part, we derive a mathematical heat conduction model and approximate it in space. We analyze the mathematical structure of the approximated thermal model and present numerical methods to solve the large-scale differential equation in time. In the second part, we design a feed-forward control approach to heat up the object and we propose feedback methods to stabilize the reached temperature in presence of thermal emissions.

In Chapter 2, we introduce the geometrical objects with its temperature-dependent material properties. We derive the fundamental quasilinear heat equation and we specify its boundary conditions, which cause a cooling or heating of the object.

In Chapter 3, we approximate the entire heat equation formalism in space using a finite volume method and we obtain a large-scale ordinary differential equation. In case of constant material properties, we yield a linear system consisting of sparse matrices.

In Chapter 4, we describe the algebraic structure of the approximated linear system. We compute the eigenvalues and eigenvectors, which we use to construct an analytical solution.

In Chapter 5, we present the numerical solvers to integrate the approximated heat equation in time. We introduce the Euler integration approaches and Runge-Kutta methods and compare them with respect to an application on the heat equation.

In Chapter 6, we model the spatially distributed multiple actuators and sensors on the boundary faces. We also sketch the heating-up procedure, which is driven by a feed-forward control approach and the subsequent stabilization with a feedback controller.

In Chapter 7, we design a feed-forward control to heat up the object. In a first step we derive an analytical prototype input signal for a simplified heat conduction model and in an second step we adjust the input function with optimization-based methods. The whole feed-forward design approach is exemplified in a comprehensive two-dim. example.

In Chapter 8, we construct a feedback law to stabilize the reached temperature such that the supplied power compensates the thermal emissions. We present a linear-quadratic regulator concept and a model predictive control technique, and we discuss the applicability of both controllers for the considered heat conduction phenomena. Furthermore, we apply the feed-forward and feedback control on an example with a three-dimensional geometry.

In Chapter 9, we summarize the findings of this thesis and we state four promising concepts and methods to enhance the proposed heat conduction framework.

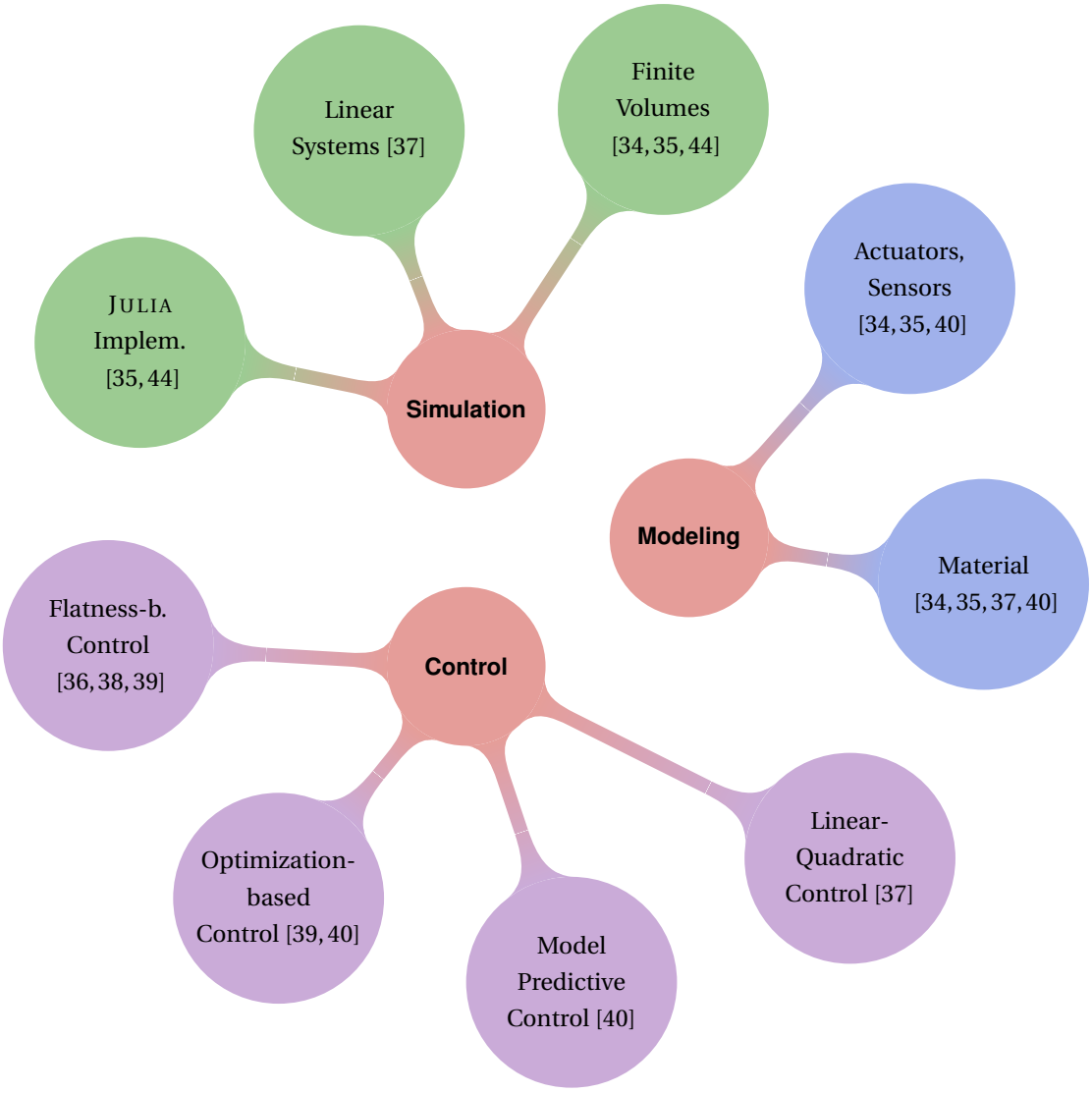
In Appendix A, we state a brief introduction to the analytical solution of the heat equation for Neumann and Dirichlet boundary conditions. The analytical solution for the Neumann problem provides true data for a comparison with the numerical solvers in Chapter 5. Furthermore, we derive the Riccati equation, which is a central fact to compute the linear-quadratic regulation in Chapter 8.

In Appendix B, we list the evaluations of numerical experiments and their corresponding source code listings.

Scientific Contribution

The topics of this thesis were presented in seven articles [34–40]. Moreover, the author developed with the JULIA programming language the software libraries *Hestia.jl* [44] to model and approximate the heat conduction scenarios and *BellBruno.jl* [45] to compute the derivatives of the reference signal in the flatness-based control in Section 7.1. Further available numerical simulations are cited in the mentioned articles. Additionally, the author contributed to the articles [41–43], which discuss the system reconstruction from given simulation or measurement data. This topic is not covered in this thesis.

We visualize the scientific contributions in the mind map below.


Linear Systems [37]
Finite Volumes [34, 35, 44]
JULIA Implem. [35, 44]
Simulation
Actuators, Sensors [34, 35, 40]
Modeling
Material [34, 35, 37, 40]
Flatness-b. Control [36, 38, 39]
Control
Optimization-based Control [39, 40]
Model Predictive Control [40]
Linear-Quadratic Control [37]

Modeling and Simulation

2

Heat Conduction

The thermal dynamics in a solid object is the fundamental phenomenon in this work. It is described by the heat conduction inside the object and the heating and cooling processes on the boundary surfaces of the object. In this chapter, we introduce a heat conduction model in continuous time and space. This model incorporates the geometrical object, the material properties, the heat equation and the boundary conditions. We discuss our heat conduction problems in this work for the one-dimensional rod, the two-dim. rectangle and three-dim. cuboid and so we present these geometries in Section 2.1. The object is further characterized with its material properties in Section 2.2. Their values determine the speed of temperature variation inside the object. From the physical laws of heat transfer, we derive the heat equation in Section 2.3, which contains the core elements for all further ideas regarding the simulation and control. The heat equation operates inside the object and so describe the interaction with the object's surrounding in Section 2.4. Finally, the natural cooling via convective and radiative emissions is explained in Section 2.5.

The heat conduction modeling with cooling and heat supply is based on our article [34].

2.1 Geometric Cubic Model

In this thesis, we consider three geometries for our heat conduction phenomena:

- a one-dimensional rod $\Omega_1 := (0, L)$,
- a two-dim. rectangle $\Omega_2 := (0, L) \times (0, W)$ and
- a three-dim. cuboid $\Omega_3 := (0, L) \times (0, W) \times (0, H)$

with a fixed length $L > 0$, width $W > 0$ and height $H > 0$. We identify the number of spatial dimensions by $N_d = \{1, 2, 3\}$. In general, the boundary is defined by $\partial\Omega_{N_d} := \overline{\Omega}_{N_d} \setminus \Omega_{N_d}$ where $\overline{\Omega}_{N_d}$ denotes the closed set of Ω_{N_d}. One may think of the boundary as an infinitesimal thin interface between the object and its surrounding. The boundary plays an important role because the supplied and emitted heat is specified on the boundary and it drives the thermal dynamics inside the object. A position inside the ob-

Figure 2.1: Three-dim. cuboid with boundary sides. The boundary sides B_E (east, blue), B_S (south, green) and B_T (topside, purple) are visible. Not visible are B_W (west; opposite to B_E), B_N (north; opposite to B_S) and B_U (underside; opposite to B_T).

ject or on the boundary is defined as $x \in \overline{\Omega}_{N_d}$ with

$$x = \begin{cases} x_1 & \text{if } N_d = 1, \\ (x_1, x_2)^\top & \text{if } N_d = 2, \\ (x_1, x_2, x_3)^\top & \text{if } N_d = 3. \end{cases}$$

All three geometries have a western and eastern boundary side B_W and B_E; the rectangle and the cuboid have a southern and northern side B_S and B_N; and only the cuboid has a underside B_U and topside B_T. The boundary sides of the cuboid are portrayed in Fig. 2.1 and the specifications of all boundary sides are noted in Table 2.1. In case of the one-dim. rod Ω_1, we only have two boundary sides B_W and B_E which are separated points. This simple situation limits significantly the possible boundary specification as we are only able to supply or emit thermal energy on these two sides, see also Section 2.4. In the literature, one-dim. models are assumed

- to study the analytical and numerical behavior of the heat equation, and
- to design control and observer algorithms for thermal systems with one actuator (on one boundary side) and one sensor (on the opposite boundary side)[1], see e.g. [46, 47].

We consider one-dim. heat conduction examples in the modeling, simulation and control design to highlight the discussed physical processes. We illustrate the thermal dynamics inside the rod and on the boundary sides in the subsequent Sections 2.3, 2.4 and 2.5. Furthermore, the one-dim. heat equation helps us to understand the numerical approximation in Chapter 4 and 5, and we note the continuous analytical solution of the one-dim. problem in Appendix A.1. Finally, it is a fundamental system to derive the feed-forward and feedback control algorithms also for two- and three-dim. objects in Chapter 7 and 8.

The two-dim. rectangle has four one-dim. connected boundary sides B_W, B_E, B_S and B_N, see Fig. 2.2. These sides enable us to design simulations with multiple actuators and multiple sensors along the boundary sides, and thermal emissions with a relevant cooling impact. The two-dim. rectangular is still an approximation of the real three-dim. situation. Due to computational aspects it may be useful in several cases to discuss the two-dim. geometry rather than the full three-dim. object because the simulation and optimization in three dimensions require usually more data storage and more computing steps than the two-dim. case.

[1] Such systems are called single-input single-output (SISO) systems.

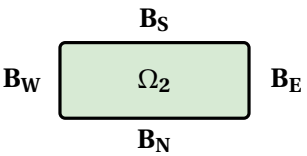


Figure 2.2: Rectangle object with boundary sides B_W (west), B_E (east), B_S (south) and B_N (north).

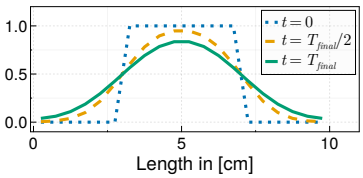
Name	Symbol	Rod	Rectangle	Cuboid
West	B_W	$\{0\}$	$\{0\} \times [0, W]$	$\{0\} \times [0, W] \times [0, H]$
East	B_E	$\{L\}$	$\{L\} \times [0, W]$	$\{L\} \times [0, W] \times [0, H]$
South	B_S		$[0, L] \times \{0\}$	$[0, L] \times \{0\} \times [0, H]$
North	B_N		$[0, L] \times \{W\}$	$[0, L] \times \{W\} \times [0, H]$
Underside	B_U			$[0, L] \times [0, W] \times \{0\}$
Topside	B_T			$[0, L] \times [0, W] \times \{H\}$

Table 2.1: Specification of Boundary Sides.

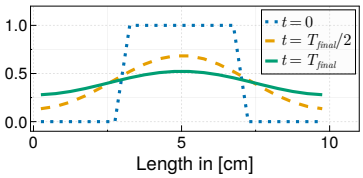
Two- and three-dim. geometries are often utilized in simulations to investigate realistic scenarios like physical or chemical phenomena and experiments. Based on these simulations, optimization-based control strategies can be designed to steer the dynamical system, e.g. the temperature. We consider the cuboid as geometry to formulate the heat equation because it represents appropriately a realistic scenario, such that physical properties, units and laws fit to the mathematical model.

2.2 Material and Physical Properties

We consider a metal or metal alloy as the material of the object and it has the properties: *mass density* ρ, *specific heat capacity* c and *thermal conductivity* λ. These properties specify the ability of an object to store or conduct thermal energy and this means that they influence how fast the temperature varies inside an object. We showcase the speed of thermal conduction in a small numerical experiment, see Fig. 2.3. Here, we assume a one-dim. rod with length $L = 0.1$ meter, simulation time $T_{final} = 10$ seconds, specific heat capacity $c = 1$, density $\rho = 1$ and two different values of the thermal conductivity. In the first simulation, we assume $\lambda = 5 \cdot 10^{-6}$ and we notice only a small temperature variation in Fig. 2.3 (a). In the second simulation, we have $\lambda = 2 \cdot 10^{-5}$ and we obtain a fast conduction Fig. 2.3 (b) such that the temperature is almost in an equilibrium state at the final time.



(a) Slow, $\lambda = 5 \cdot 10^{-6}$



(b) Fast, $\lambda = 2 \cdot 10^{-5}$

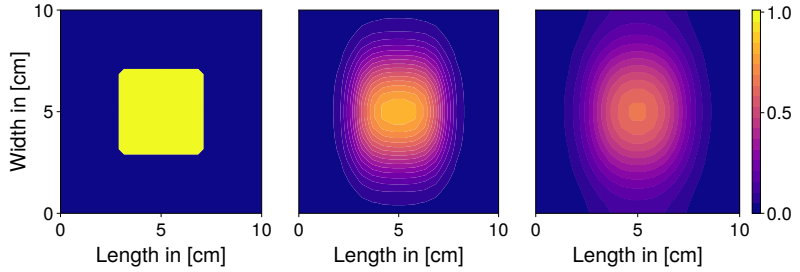
Figure 2.3: Comparison of slow and fast heat conduction in a one-dim. rod with $c = \rho = 1$ and $T_{final} = 20$ seconds. The rod is insulated on both boundary sides, B_W and B_E, as explained in Section 2.4.

The condition of a material may depend on its age, composition (in case of alloys), temperature and further internal and external influences. We neglect most of these dependencies and only consider two facts: the temperature of the material and a possible anisotropy of the thermal conductivity. Hence, we model the material properties with temperature θ as polynomial functions

$$\rho(\theta) := \sum_{n=0}^{N_\rho} \rho_n \, \theta^n \quad \text{and} \tag{2.1}$$

$$c(\theta) := \sum_{n=0}^{N_{cap}} c_n \, \theta^n. \tag{2.2}$$

This general setup shrinks to constant values if $N_\rho = 0$ or $N_{cap} = 0$ and so we have $\rho = \rho_0$ or $c = c_0$. The mass density ρ, the mass m_Ω and the volume V_Ω of object Ω are related via the physical law $m_\Omega = \rho \, V_\Omega$. This means that a change of $\rho(\theta)$ via a variation in θ affects physically either the mass m_Ω



(a) Initial, $t = 0$ (b) Half, $t = \frac{T_{final}}{2}$ (c) Final, $t = T_{final}$

Figure 2.4: Anisotropic heat conduction in a rectangle with $\lambda = \text{diag}(5 \cdot 10^{-6}, 2 \cdot 10^{-5})$, $c = \rho = 1$, and $T_{final} = 10$ seconds. The temperature varies faster in y-direction than in x-direction.

or the volume V_Ω. In this work, we neglect both effects and we assume hereby only (very) small variations of $\rho(\theta)$ and so $\frac{d}{d\theta}\rho(\theta) \approx 0$.

The thermal conductivity is assumed to be depend on the temperature, too.[2] Additionally, we distinguish isotropic and anisotropic heat conduction, see also [49, p. 330]. In the anisotropic case, the thermal conductivity differs for each spatial direction, which is only plausible for geometries in two and three dimensions. Anisotropic heat conduction implies that the temperature varies faster along one spatial orientation than along the other(s). We define the thermal conductivity as diagonal matrix[3]

$$\lambda(\theta) := \begin{pmatrix} \lambda_1(\theta) & & \\ & \lambda_2(\theta) & \\ & & \lambda_3(\theta) \end{pmatrix} \tag{2.3}$$

with the polynomial function

$$\lambda_j(\theta) := \sum_{n=0}^{N_\lambda} \lambda_{j,n}\, \theta^n \quad \text{for} \quad j \in \{1,2,3\} \tag{2.4}$$

similar to the mass density and the specific heat capacity above. If the thermal conductivity does not depend on the spatial orientation as $\lambda_1(\theta) \equiv \lambda_2(\theta)\,[\equiv \lambda_3(\theta)] \equiv \lambda(\theta)$, then we have an isotropic scenario. We visualize in Fig. 2.4 the effect of anisotropic heat conduction for a two-dim. square geometry with $L = W = 0.1$ meter and material properties

$$c = \rho = 1 \quad , \quad \lambda = \begin{pmatrix} 5 \cdot 10^{-6} & \\ & 2 \cdot 10^{-5} \end{pmatrix}.$$

We see that the temperature varies faster in x_2-direction than in x_1-direction because $\lambda_2 > \lambda_1$.

In material science, the identification of these material properties, in particular for a temperature range, is a specialized field of research. We find tables with the material properties for various metals in [50, p. 21] and in the doctoral thesis [51, p. 124]. In the latter contribution [51, p. 120], the author notes the specific heat capacity and thermal conductivity as polynomials of the temperature for some specific types of steel. Furthermore, the document [52] provides a large data set of material properties for various temperatures. In the most of our examples, we assume steel as the treated material but we do not specify the steel.

[2] The concept of temperature-dependent thermal conductivity is also known from the Wiedemann-Franz law $\frac{\lambda}{\sigma} = L \cdot \theta$ with the electrical conductivity σ and the Lorenz number L, see [48].

[3] In case of a rectangle, $N_d = 2$, we have $\lambda(\theta) := \text{diag}(\lambda_1(\theta), \lambda_2(\theta))$.

2.3 Formulation of the Heat Equation

This section provides a basic introduction to the mathematical modeling of heat transfer in solid objects. We explain the fundamental elements of the first law of thermodynamics in accordance with the literature, see [49, p. 118] and [53, p. 54], and we guide step-by-step towards the heat equation in integral and differential form. We consider the three-dim. cuboid Ω_3 for this formulation to yield a proper physical interpretation, and we showcase how to transfer these ideas to the one-dim. heat conduction in the end of this section. The core element of this derivation and further calculations is function

$$\vartheta : [0, T_{final}] \times \overline{\Omega} \to \mathbb{R}_{\geq 0} \tag{2.5}$$

with final time $T_{final} \in \mathbb{R}_{>0}$. It describes the variation in time and space of the temperature distribution inside the geometry and on the boundary sides. Hence, $\vartheta(t,x)$ is the solution of the heat equation. As we introduce several physical properties in this section, we list them in Table 2.2.

Table 2.2: Thermodynamical Variables.

Sym.	Property	Unit
u	Specific int. energy	$\frac{J}{kg}$
U	Internal energy	J
Q	Stored heat	J
W	Supplied work	J
P	Supplied power	W
$\dot{Q}$	Rate of heat flow	W
$\dot{q}$	Heat flux in Ω	$\frac{W}{m^2}$
ϕ	Power density on $\partial\Omega$	$\frac{W}{m^2}$

First of all, we find the specific internal energy u via the integration of the specific heat capacity c over temperature θ as

$$\int_{\theta_0}^{\theta} c(\tilde{\theta})\, d\tilde{\theta} =: u(\theta) \tag{2.6}$$

and we see that u may be noted as polynomial function like c in Eq. (2.2). The specific internal energy expresses the internal energy per mass. So, we find the internal energy $U : [0, T_{final}] \to \mathbb{R}_{\geq 0}$ as we sum up u over each infinitesimal small mass element. The mass equals an integration of density ρ over the volume of Ω, and thus we yield the internal energy

$$U(t) := \int_{\Omega} \rho(\vartheta(t,x))\; u(\vartheta(t,x))\, dx. \tag{2.7}$$

According to the **first law of thermodynamics**, the rate of change of the internal energy ΔU is driven by the stored heat Q and the supplied work W as [4]

$$\Delta U(t) = Q(t) + W(t). \tag{2.8}$$

We assume the net energy transfer W into the system (or object) as positive and from the system as negative. We reformulate Eq. (2.8) in terms of a variation in time as

$$\frac{d}{dt}U(t) \;=\; \frac{d}{dt}Q(t) + P(t) \tag{2.9}$$

with power $P(t) := \frac{d}{dt}W(t)$. We formulate each part of Eq. (2.9) separately in the next steps. We differentiate $U(t)$ in Eq. (2.7) to yield

$$\begin{aligned}\frac{d}{dt}U(t) &= \int_{\Omega} \frac{d\rho(\vartheta)}{d\vartheta}\frac{\partial\vartheta}{\partial t}\; u(\vartheta) + \rho(\vartheta)\; \frac{du(\vartheta)}{d\vartheta}\; \frac{\partial\vartheta}{\partial t}\; dx \\ &= \int_{\Omega} \left[\frac{d\rho(\vartheta)}{d\vartheta}\; u(\vartheta) + \rho(\vartheta)\; c(\vartheta) \right] \frac{\partial}{\partial t}\vartheta(t,x)\; dx\end{aligned}$$

[4] In some contributions, the first law of thermodynamics is noted with the inexact differential δ or đ on the right-hand side as $dU = \delta Q + \delta W$ with dU as the total differential of U. See also [54, p. 81].

with $\vartheta = \vartheta(t,x)$ and $\frac{d}{d\theta}u(\theta) = c(\theta)$ from Eq. (2.6). We neglect a variation of the mass and the volume, $\frac{d}{d\theta}\rho(\theta) \approx 0$, and so we obtain

$$\frac{d}{dt}U(t) = \int_\Omega \rho(\vartheta(t,x))\; c(\vartheta(t,x))\; \frac{\partial}{\partial t}\vartheta(t,x)\; dx. \tag{2.10}$$

On the right-hand side of Eq. (2.9), the rate of heat flow $\frac{d}{dt}Q(t)$ describes how much thermal energy (or heat) is transferred per time in the cuboid. It is defined by

$$\frac{d}{dt}Q(t) \;:=\; -\int_{\partial\Omega} \dot{q}(t,x)\cdot\vec{n}\; dx \tag{2.11}$$

with heat flux[5] $\dot{q}$ and the outer normal vector on the boundary $\vec{n} \perp \partial\Omega$. The heat flux describes motion of heat from warm to cold areas. According to Fourier's law, it is defined as

$$\dot{q}(t,x) := -\lambda(\vartheta(t,x))\; \nabla\vartheta(t,x) \tag{2.12}$$

with the temperature gradient

$$\nabla\vartheta(t,x) := \left(\frac{\partial}{\partial x_1}\vartheta(t,x), \frac{\partial}{\partial x_2}\vartheta(t,x), \frac{\partial}{\partial x_3}\vartheta(t,x)\right)^\top.$$

As the temperature gradient $\nabla\vartheta(t,x)$ points towards the hot regions, the heat flux $\dot{q}$ forces the hot regions to reduce the temperature while the temperature in the cold regions increase. The rate of heat flow describes the thermal dynamics inside the cuboid. Therefore, we apply the divergence theorem [6]

$$\int_{\partial\Omega} v(x)\cdot\vec{n}\; dx = \int_\Omega \operatorname{div}(v(x))\, dx$$

on Eq. (2.11), see [3, p. 20]), and we obtain the rate of heat flow as

$$\begin{aligned}\frac{d}{dt}Q(t) &= -\int_\Omega \operatorname{div}\left[\dot{q}(t,x)\right]\; dx \\ &= \int_\Omega \operatorname{div}\left[\lambda(\vartheta(t,x))\; \nabla\vartheta(t,x)\right]\; dx.\end{aligned} \tag{2.13}$$

The integrand in Eq. (2.13) can be noted as

$$\operatorname{div}\left[\lambda(\vartheta(t,x))\; \nabla\vartheta(t,x)\right] = \sum_{i=1}^{N_d} \frac{\partial}{\partial x_i}\left[\lambda_i(\vartheta(t,x))\; \frac{\partial}{\partial x_i}\vartheta(t,x)\right] \tag{2.14}$$

with $N_d = 3$. If the thermal conductivity is temperature-independent, then we note the integrand as

$$\operatorname{div}\left[\lambda\; \nabla\vartheta(t,x)\right] = \sum_{i=1}^{N_d} \lambda_i \frac{\partial^2}{\partial x_i^2}\vartheta(t,x).$$

The second term on the right-hand side of Eq. (2.9) describes the supplied and emitted power $P(t)$. It expresses the transition of heat from the object to its surrounding and backwards and so it acts on the object's boundary. We define the power analog to the rate of heat flow as

$$P(t) \;:=\; \int_{\partial\Omega} \phi(t,x)\cdot\vec{n}(x)\; dx \tag{2.15}$$

[5] $\dot{q}$ is also known as heat flux density.

[6] The divergence theorem is originally described by and also named after Johann Carl Friedrich Gauß (∗1777, †1855), Mikhail Vasilyevich Ostrogradsky (∗1801, †1862) [55] and George Green (∗1793, †1841).

with outer normal vector on the boundary $\vec{n} \perp \partial\Omega$ which is defined by

$$\vec{n}(x) := \begin{cases} -1 & \text{if } x \in B_W \cup B_S \cup B_U, \\ +1 & \text{if } x \in B_E \cup B_N \cup B_T \end{cases} \tag{2.16}$$

and power density

$$\phi(t,x) := \lambda(\vartheta(t,x))\, \nabla\vartheta(t,x) \tag{2.17}$$

analog to the heat flux in Eq. (2.12). The power density ϕ and the heat flux $\dot{q}$ are equivalent physical objects but we distinguish both as $\dot{q}$ occurs inside the geometrical object Ω and ϕ operates on the boundary $\partial\Omega$. In Section 2.4 we introduce the boundary conditions and discuss the power density ϕ with respect to its cooling and heating behavior. We summarize Eq. (2.15, 2.17) to note the supplied power

$$P(t) \;=\; \int_{\partial\Omega} [\lambda(\vartheta(t,x))\, \nabla\vartheta(t,x)] \cdot \vec{n}(x)\, dx. \tag{2.18}$$

Finally, we assemble Eq. (2.10, 2.13, 2.18) in the first law of thermodynamics (2.9). Thus, we yield the **integral form of the quasilinear heat conduction**

$$\underbrace{\int_\Omega \rho(\vartheta)\, c(\vartheta)\, \frac{\partial}{\partial t}\vartheta(t,x)\, dx}_{\frac{d}{dt}U(t)} \;=\; \underbrace{\int_\Omega \operatorname{div}[\lambda(\vartheta)\, \nabla\vartheta(t,x)]\, dx}_{\frac{d}{dt}Q(t)} + \underbrace{\int_{\partial\Omega} [\lambda(\vartheta)\, \nabla\vartheta(t,x)] \cdot \vec{n}(x) dx}_{P(t)}. \tag{2.19}$$

This integral equation (2.19) provides the core element to derive the finite volume approximation in Chapter 3 and to design energy-based control approaches in Section 7.5 and in Chapter 8.

Now, we reformulate the heat equation (2.19) in differential form. We integrate on both sides over the same volume Ω and so we omit the integral and note the partial differential equation of the heat conduction.

Definition 2.1 (Quasilinear heat equation)
We note the **quasilinear heat equation** as

$$\rho(\vartheta)\, c(\vartheta)\, \frac{\partial}{\partial t}\vartheta(t,x) \;=\; \operatorname{div}[\lambda(\vartheta)\, \nabla\vartheta(t,x)] \tag{2.20a}$$

for $(t,x) \in (0, T_{final}] \times \Omega$ and with boundary condition

$$[\lambda(\vartheta(t,x))\, \nabla\vartheta(t,x)] \cdot \vec{n}(x) \;=\; \phi(t,x) \quad \text{for} \quad x \in \partial\Omega \tag{2.20b}$$

and initial condition

$$\vartheta(0,x) = \vartheta_0(x) \quad \text{for} \quad x \in \Omega. \tag{2.20c}$$

○

A partial differential equation is called quasilinear if it has coefficients with the unknown variable (here: temperature ϑ) and its highest order derivative is linear and lower order derivatives may be nonlinear. This description is not clearly recognizable in Eq. (2.20a) but we find it, if we evaluate the differential operators in Eq. (2.14) and we note the nonlinear expression

$$\operatorname{div}[\lambda(\vartheta(t,x))\,\nabla\vartheta(t,x)] = \sum_{i=1}^{3}\left[\lambda_i(\vartheta(t,x))\underbrace{\frac{\partial^2}{\partial x_i^2}\vartheta(t,x)}_{\text{linear}} + \frac{\partial}{\partial\vartheta}\,\lambda_i(\vartheta(t,x))\underbrace{\left(\frac{\partial}{\partial x_i}\vartheta(t,x)\right)^2}_{\text{nonlinear}}\right].$$

For further information on quasilinear PDE, we refer to the book [3, p. 2] and to the doctoral thesis [56, p. 2]. We remark that the term "quasilinear heat equation" is not unique because some authors denote other types of the heat equation with it.

In this thesis, we do not discuss the analysis of the quasilinear heat equation explicitly because it is out of scope of this work and much more complex than the linear heat equation, see e.g. [56, p. 2, 4]. In contrast, we rather consider the spatially approximated quasilinear heat equation for the controller design, which is introduced in Chapter 3.

If we assume constant material properties, as $\lambda = \operatorname{diag}(\lambda_1, \lambda_2, \lambda_3)$ with $\rho \in \mathbb{R}_{>0}$ and $c \in \mathbb{R}_{>0}$, then we obtain from Eq. (2.14) and (2.20a) the well-known (anisotropic) **linear heat equation**

$$\frac{\partial}{\partial t}\vartheta(t,x) = \frac{1}{c\,\rho}\sum_{l=1}^{N_d}\lambda_l\frac{\partial^2}{\partial x_l^2}\vartheta(t,x). \tag{2.21}$$

In the next chapters, we also note the linear heat equation (2.21) with diffusivity $\alpha_l = \frac{\lambda_l}{c\,\rho}$ for $l \in \{1,2,3\}$. We consider the linear heat equation as an important special case because it helps us to understand the spatial approximation and its numerical behavior, see Section 3.4 and Chapter 4. Furthermore, the linear system is one of the central elements of the flatness-based control design in Chapter 7. In Appendix A.1, we note the analytical solution of the one-dim. heat equation with zero Neumann boundary condition: $\phi(t,x) = 0$.

We refer to the literature [3, p. 44], [4, p. 75] for further information about the mathematical analysis of the linear heat equation.

Example: Temperature-dependent Heat Conduction

As we have a formal description of quasilinear heat conduction now at hand, we apply these ideas on a one-dim. rod model with length $L = 0.2$ meter to showcase the thermal dynamics. Such a reduction of a real three-dim. object to a one-dim. model might be reasonable, if the width and the height are much smaller than the length or if the heat conduction in the directions x_2 and x_3 are not relevant. We assume an object made of steel with a specific heat capacity $c = 400$, a mass density $\rho = 8000$, and a temp.-dependent thermal conductivity as noted in Table 2.3. We have five data samples in Table 2.3 and so we approximate the curve by a quartic function

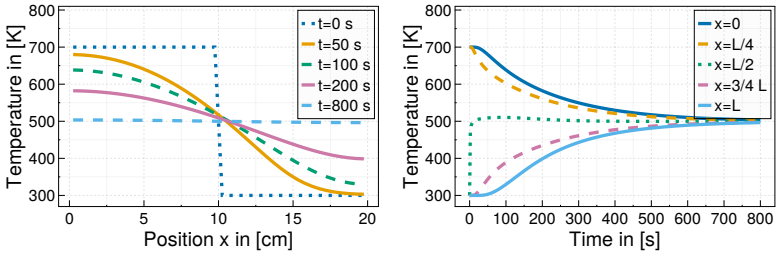
Table 2.3: Th. conductivity data.

Θ in [K]	λ
300	40
400	50
500	70
600	85
700	90

$$\lambda(\theta) = \lambda_0 + \lambda_1\theta + \lambda_2\theta^2 + \lambda_3\theta^3 + \lambda_4\theta^4$$

where we find the approximated parameters as

$$[\lambda_0, \ldots, \lambda_4] = [370, -2.85, 8.458\cdot 10^{-3}, -10^{-5}, 4.1667\cdot 10^{-9}].$$



(a) Temperature distribution (b) Temperature evolution

Figure 2.6: Heat conduction with nonlinear thermal conductivity in a one-dim. rod. The snapshots of the temperature distribution in (a) and the thermal dynamics at five points in (b) show that temperatures converge to the mean value of 500 Kelvin.

The graph of the thermal conductivity function is portrayed in Fig. 2.5. Consequently, we note the one-dim. quasilinear heat equation as

$$\frac{\partial}{\partial t}\vartheta(t,x) \;=\; \frac{1}{\rho\, c}\frac{\partial}{\partial x}\left[\lambda(\vartheta(t,x))\,\frac{\partial}{\partial x}\vartheta(t,x)\right]$$

with thermal conductivity

$$\lambda(\theta) = \; 370 - 2.85\theta + 8.458\cdot 10^{-3}\theta^2 - 10^{-5}\theta^3 + 4.1667\cdot 10^{-9}\theta^4. \qquad (2.22)$$

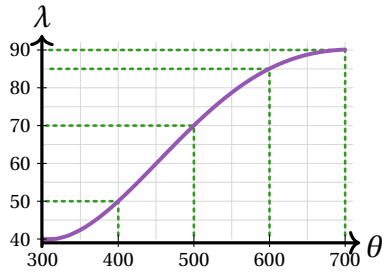


Figure 2.5: Nonlinear thermal conductivity $\lambda(\theta)$ as in Eq. (2.22).

We assume the initial temperature distribution

$$\vartheta(0,x) = \begin{cases} 300 & \text{for } x \in [0, \frac{L}{2}], \\ 700 & \text{for } x \in [\frac{L}{2}, L]. \end{cases}$$

The one-dim. rod has two boundary sides B_W and B_E. We assume that both boundary sides are insulated, which means we have a heat flux or power density of $\phi(t,x) \equiv 0$. We know from identity (2.16) that $\vec{n} = -1$ on B_W and $\vec{n} = +1$ on B_E. So, we yield the boundary conditions

$$-\lambda(\vartheta(t,x))\frac{\partial}{\partial x}\vartheta(t,x) = 0 \quad \text{for } x \in B_W \text{ and}$$
$$\lambda(\vartheta(t,x))\frac{\partial}{\partial x}\vartheta(t,x) = 0 \quad \text{for } x \in B_E.$$

The one-dim. rod is approximated, see Chapter 3, and the heat equation is simulated for $T_{final} = 800$ seconds. The simulation results are visualized in Fig. 2.6, in which Fig. 2.6 (a) portrays the temperature in each position $x \in \Omega$ at five time stamps; and Fig. 2.6 (b) presents the temperature variation in time at five positions. We find that the high temperatures close to boundary B_W decrease faster than the low temperatures close to B_E rise. This behavior is caused by the strong thermal conductivity for high temperatures. All temperatures approach for $t \to \infty$ the mean temperature of 500 Kelvin because both boundary sides are insulated.

2.4 Emitted and Supplied Heat Flux

Boundary condition (2.20b) describes the interaction between the thermal dynamics inside the object and the surrounding. We introduced in Section 2.1 the boundary sides, see Table 2.1, and we stated that boundary $\partial\Omega$ is an infinitesimal thin interface between object Ω and its surrounding. In Section 2.3, we explained that thermal energy can be supplied to or emitted from the object. The overall sum of supplied and emitted power P is noted in Eq. (2.18). According to the first law of thermo-

dynamics, see Eq. (2.8, 2.9), the amount of internal energy $U(t)$ is determined by the rate of heat flow $\frac{d}{dt}Q(t)$ inside the object and the supplied and emitted power $P(t)$ on the boundary sides. We do not have heat sinks or sources inside the geometry, and so only the power via "external" processes $P(t)$ increase or decrease the level of internal energy. This idea includes the fact that $\frac{d}{dt}Q(t) \equiv 0$, but we remark that it does not mean $\operatorname{div}[\lambda(\vartheta(t,x))\,\nabla\vartheta(t,x)] \equiv 0$. If the overall power is positive, then the amount of internal energy and equally the mean temperature increase, whereas a negative power implies a decreasing internal energy and mean temperature. We distinguish the processes on boundary $\partial\Omega$ as

1. $P < 0$ cooling down: thermal emissions cause a temperature drop,

2. $P > 0$ heating up: heat supply leads to a temperature rise.

The emission of heat is assumed to occur *naturally*, which means that it depends on the physical properties of the object and its surrounding. This *natural* process is assumed to be driven by heat transfer and heat radiation as are explained in Section 2.5. In contrast to this, we consider the heat supply as an artificial process, which is carried out by actuators operating on the boundary. We assume thermal actuators like heating elements or lasers. Although some thermal actuators like Peltier elements might be able to heat and to cool, we only consider actuators, which are solely able to heat. The actuators are considered to operate on a subset of the whole boundary, $B_{in} \subseteq \partial\Omega$. This actuator's boundary B_{in} might be identical with a boundary side like B_W, B_E, etc. or an union of boundary sides for example $B_{in} = B_W \cup B_S$. We might have thermal emissions on the actuator's boundary B_{in}, too.

In the previous Section 2.3, we introduced the supplied power as the integral of heat flux or power density ϕ over the boundary. This heat flux consists of thermal emissions ϕ_{em} from the object to the surrounding and of heat supply ϕ_{in} from the actuator to the object. The emitted and supplied heat flux are described below in Def. 2.2.

Definition 2.2 (Emitted and supplied heat flux)
The emitted and the supplied heat flux vary in time t and space x. The emitted heat flux is defined on the whole boundary $\partial\Omega$ to be less than or equal to zero. The supplied heat flux is only defined on the actuator's boundary $B_{in} \subseteq \partial\Omega$ and is considered to be greater than or equal to zero. So, we note emitted heat flux as $\phi_{em} : [0, T_{final}] \times \partial\Omega \to (-\infty, 0]$ and the supplied heat flux as $\phi_{in} : [0, T_{final}] \times B_{in} \to [0, \infty)$.
We summarize these ideas and we note the total heat flux as

$$\phi(t,x) = \begin{cases} \phi_{in}(t,x) + \phi_{em}(t,x) & \text{for } x \in B_{in}, \\ \phi_{em}(t,x) & \text{for } x \in \partial\Omega \setminus B_{in}. \end{cases} \tag{2.23}$$

○

We conclude from Definition 2.2 to distinguish the boundary condition (2.20b) for the actuated boundary side $x \in B_{in}$ as

$$[\lambda(\vartheta(t,x))\,\nabla\vartheta(t,x)] \cdot \vec{n} \;=\; \phi_{in}(t,x) + \phi_{em}(t,x)$$

and for the remaining (not actuated) boundary $x \in \partial\Omega \setminus B_{in}$ as

$$[\lambda(\vartheta(t,x))\,\nabla\vartheta(t,x)]\cdot\vec{n} \;=\; \phi_{em}(t,x).$$

If a boundary side has a vanishing *emitted* heat flux, $\phi_{em}(t,x) \equiv 0$, then we denote it as *insulated*. If all boundary sides are insulated and no heat supply is active then all temperatures converge towards the mean temperature

$$\overline{\vartheta} = \frac{1}{V_\Omega}\int_\Omega \vartheta_0(x)dx$$

with initial values $\vartheta_0(x)$ and volume $V_\Omega = L\cdot W\cdot H$ in the three-dim. case.

Furthermore, we distinguish the **supplied power**

$$P_{in}(t) := \int_{B_{in}} \phi_{in}(t,x)dx \tag{2.24}$$

and the **emitted power**

$$P_{em}(t) := \int_{\partial\Omega} \phi_{em}(t,x)dx. \tag{2.25}$$

We stated in the beginning of this section that the overall power

$$P(t) = P_{in}(t) + P_{em}(t)$$

drives the internal energy and the mean temperature either to increase, to decrease or to hold. In the second part of this thesis, we design control approaches to heat up the object and stabilize the reached temperature. Hence, we need to guarantee that

$$P(t) = P_{in}(t) + P_{em}(t) \begin{cases} > 0 & \text{during the feed-forward control and} \\ = 0 & \text{in the temperature stabilization.} \end{cases}$$

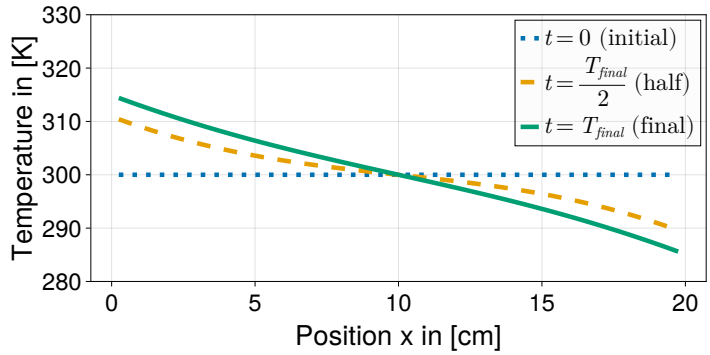
We consider the remaining case $P(t) < 0$ as an undesired behavior because the temperatures leave the desired reference values.

Example: Balanced and Unbalanced Heat Supply and Emission

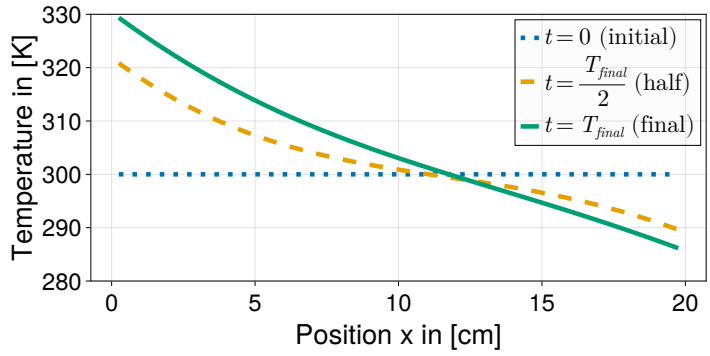
We demonstrate the findings of this section with an example of a one-dim. rod with length $L = 0.2$ and material properties $\lambda = 50$, $c = 400$, $\rho = 8000$. This rod has a pure heat supply ϕ_{in} on boundary B_W and an emission ϕ_{em} on B_E. We study two scenarios: firstly, the amount of supplied and emitted heat is equal $\phi_{in} = -\phi_{em} = 10^4$; and secondly, the supply is higher than the emissions with $\phi_{in} = 2\cdot 10^4$ and $\phi_{em} = -10^4$. The whole rod has an initial temperature of 300 Kelvin. We see in Fig. 2.7 that the temperature rises on the left side (next to B_W) and declines on the right side (next to B_E) in both scenarios. In the first scenario, in Fig. 2.7 (a), the temperature increases on the left side with the same value as it decreases on the right side because the inflow and outflow of heat are equal. So, we see that the average temperature in the rod is constant as

$$\frac{1}{L}\int_0^L \vartheta(t,x)dx \;=\; 300\,\text{Kelvin}$$

at every time $t \in [0, T_{final}]$. We may denote this thermal situation as balanced. In the second scenario, in Fig. 2.7 (b), the temperature increases



(a) Balanced Scenario



(b) Higher Supply than Emission

Figure 2.7: Temperature distribution for heat supply on the left side at $x = 0$ and heat emission on the right side at $x = 0.2$. In the first scenario (above), we assume $\phi_{in} = 10^4$ and $\phi_{em} = -10^4$. In the second scenario (below), we assume $\phi_{in} = 2 \cdot 10^4$ and $\phi_{em} = -10^4$.

stronger on the left side than it drops on the right side. This unbalanced situation also means that the internal energy and the average temperature in the rod increase by time.

The emissive heat flux ϕ_{em} is described next in detail with the linear (convective) heat transfer and nonlinear heat radiation, and the supplied heat flux ϕ_{in} is explained in the second part of this thesis, in Chapter 6.

2.5 Heat Transfer and Heat Radiation

We assume that the cooling process of the object is mainly influenced by heat transfer ϕ_{tr} and heat radiation ϕ_{rad} to the ambient environment as

$$\phi_{em}(t,x) = \phi_{tr}(t,x) + \phi_{rad}(t,x) \tag{2.26}$$

for $x \in \partial\Omega$, $t \in [0, T_{final}]$. In the first part of this section, we introduce the convective heat transfer ϕ_{tr} and second part we discuss the heat radiation ϕ_{rad}. For a comprehensive introduction, we refer to [49, p. 12], [53, p. 19] for heat transfer, and to [49, p. 28], [53, p. 28] for heat radiation.

Convective Heat Transfer

The solid object is surrounded by a quiescent or moving fluid like a liquid or a gas. The boundary $\partial\Omega$ is an interface between two media and so we need to distinguish the emitted heat flux in the solid object[7]

$$\phi_{solid}(t,x) := \lambda_{solid} \nabla\vartheta_{solid}(t,x)$$

and in the fluid

$$\phi_{fluid}(t,x) := \lambda_{fluid} \nabla\vartheta_{fluid}(t,x) \tag{2.27}$$

[7] We neglect the possible temperature-dependency in λ in this paragraph to improve the readability. See also Eq. (2.17).

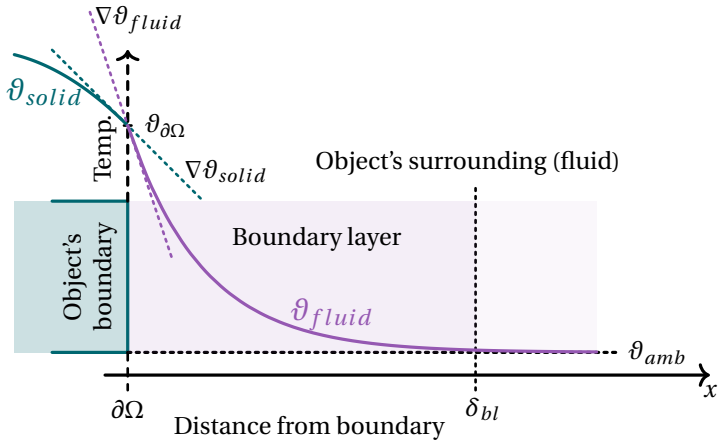


Figure 2.8: The heat is transferred from the solid object Ω to its fluid surrounding. The fluid temperature ϑ_{fluid} decreases from the boundary with $\vartheta_{\partial\Omega}$ until it reaches the ambient temperature ϑ_{amb}. This temperature drop occurs mainly in the boundary layer, which has a distance of δ_{bl} from boundary $\partial\Omega$. This figure is inspired by [49, p. 13, Fig. 1.7].

along the boundary as $x \in \partial\Omega$. We know that both emissions have to be identical as

$$\phi_{tr}(t,x) \equiv \phi_{solid}(t,x) \equiv \phi_{fluid}(t,x) \tag{2.28}$$

but their thermal conductivity values are different, e.g. $\lambda_{solid} \neq \lambda_{fluid}$ because the material (solid / fluid) is different. This fact implies that the temperature gradients are different as

$$\nabla\vartheta_{solid}(t,x) \neq \nabla\vartheta_{fluid}(t,x).$$

Next, we derive the convective heat transfer with the heat flux in the fluid $\phi_{em,fluid}$. Here, we denote the boundary temperature as $\vartheta_{\partial\Omega}$ and the fluid temperature far away from the boundary as the ambient temperature ϑ_{amb}. The temperature in the fluid ϑ_{fluid} does not change suddenly from $\vartheta_{\partial\Omega}$ to ϑ_{amb} because there exists a very thin space, a so called thermal **boundary layer**[8], between the object's boundary and the surrounding with a smooth temperature profile. The thermal boundary layer is defined as the space where the inequality

$$\frac{\vartheta_{fluid} - \vartheta_{amb}}{\vartheta_{\partial\Omega} - \vartheta_{amb}} > 0.01$$

holds, see [53, p. 277]. The temperature profile with the boundary layer in the near field of the boundary is illustrated in Fig. 2.8.

The exact physical description of the fluid's behavior and its thermal interaction with the object may be hard to describe. Thus, the heat transfer emission is approached by the formula

$$\phi_{fluid}(t,x) = -h\,[\vartheta_{\partial\Omega}(t,x) - \vartheta_{amb}]\,, \tag{2.29}$$

see [49, p. 12] and [53, p. 276]. Heat transfer coefficient h sets the intensity of the emission, and it can be determined with Eq. (2.27) as

$$h = -\lambda_{fluid}\frac{\nabla\vartheta_{fluid}(t,x)}{\vartheta_{\partial\Omega}(t,x) - \vartheta_{amb}}.$$

We consider the heat transfer coefficient h and the ambient temperature ϑ_{amb} to depend on the position on the boundary as $h : \partial\Omega \to \mathbb{R}_{\geq 0}$ and $\vartheta_{amb} : \partial\Omega \to \mathbb{R}_{\geq 0}$. We explicitly neglect that the ambient temperature varies in time. This simplification may not be physically accurate as the object's

[8] The boundary layer is firstly discovered and described by Ludwig Prandtl (*1875, †1953) [53, p. 272, 273].

temperature directly influences the ambient temperature. In accordance with identity (2.28), we note the emissions of the heat transfer as

$$\phi_{tr}(t,x) = -h(x)\,[\vartheta(t,x) - \vartheta_{amb}(x)] \tag{2.30}$$

with $(t,x) \in [0, T_{final}] \times \partial\Omega$. If we only consider heat transfer without heat radiation, then we find the boundary condition (2.20b) as

$$[\lambda(\vartheta(t,x))\nabla\vartheta(t,x)]\,\vec{n} = -h(x)\,[\vartheta(t,x) - \vartheta_{amb}(x)] \tag{2.31}$$

with λ as the thermal conductivity of the solid object.

Heat Radiation

Each object which has a temperature above zero Kelvin[9] emits heat radiation in form of electromagnetic waves. The transport of thermal energy via heat radiation does not depend on a (solid or fluid) medium like air or water and so thermal energy can be transmitted through vacuum. The ability to emit heat radiation depends on the material and its surface condition, e.g. the surface color or if it is polished or oxidized. This information is stored in the emissivity value $\varepsilon \in [0,1]$. If the object is unable to emit heat radiation, then we have $\varepsilon = 0$ and on the opposite we have $\varepsilon = 1$ in case of a black body. In real experiments, we face the issue to have several objects in the neighborhood of our test object, and all of these neighbor objects emit heat radiation towards the test object. Here, we neglect all of these neighbors and we only deal with the heat radiation of the considered object. Hence, we define the heat flux of the heat radiation as

$$\phi_{rad}(t,x) := -\sigma\,\varepsilon(x)\,\vartheta(t,x)^4 \tag{2.32}$$

with the Stefan-Boltzmann constant $\sigma \approx 5.67 \cdot 10^{-8} \frac{W}{m^2 K^4}$.[10] We assume that the emissivity depends on the position $x \in \partial\Omega$ because each boundary side may have a different surface condition. A list of emissivity values for certain properties in noted in [53, p. 542]. We summarize the findings of this section in the following definition.

[9] We consider temperatures below zero Kelvin as physically not realizable.

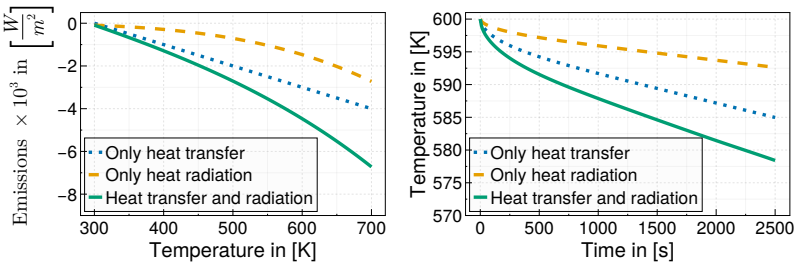
[10] Jožef Štefan (*1835, †1893) and his student Ludwig Boltzmann (*1844,†1906) worked initially on the heat radiation phenomena as in Eq. (2.32) [49, p. 29].

Definition 2.3 (Heat transfer and heat radiation)
The emitted heat flux in Eq. (2.26) consists of a heat transfer term (2.30) and a heat radiation term (2.32). The heat transfer coefficient $h : \partial\Omega \to \mathbb{R}_{\geq 0}$ in Eq. (2.30) and the emissivity $\varepsilon : \partial\Omega \to [0,1]$ in Eq. (2.32) scale the influence of convective heat transfer and heat radiation for each position on the surface $x \in \partial\Omega$. In conclusion, we note the total emitted heat flux as

$$\phi_{em}(t,x) := -h(x)\,[\vartheta(t,x) - \vartheta_{amb}(x)] - \sigma\,\varepsilon(x)\,\vartheta(t,x)^4 \tag{2.33}$$

with the ambient temperature $\vartheta_{amb} : \partial\Omega \to \mathbb{R}_{\geq 0}$ and the Stefan-Boltzmann constant $\sigma \approx 5.67 \cdot 10^{-8} \frac{W}{m^2 K^4}$. ◯

We remark that we find nonlinear expressions in two parts of our heat conduction problem: in the quasilinear diffusion $\operatorname{div}[\lambda(\vartheta(t,x))\,\nabla\vartheta(t,x)]$ and in the heat radiation (2.32). Both facts imply the need to approximate



(a) Thermal Emissions

(b) Temperature drop at B_E

Figure 2.9: Comparison of heat transfer with $h = 10$, $\vartheta_{amb} = 300$ versus heat radiation with $\varepsilon = 0.2$. The linear behavior of the heat transfer ϕ_{tr} and the nonlinear heat radiation ϕ_{rad} are visualized on the left side (a). The cooling-down process on boundary B_E of the one-dim. rod is plotted on the right side (b).

the heat equation in space and time properly. In case of constant material properties and no heat radiation, we are able to find an approximated closed-form solution, see Section 4.3.

Example: Comparison of Heat Transfer and Heat Radiation

In the end of this section, we present a small simulation example of heat transfer and heat radiation. We assume a one-dim. rod with length $L = 0.2$, material properties $\lambda = 50$, $c = 400$, $\rho = 8000$ and an initial temperature of $\vartheta_0(x) = 600$ Kelvin for $x \in \overline{\Omega} = [0, L]$. The rod is assumed to be insulated on the left side, $\phi_{em}(x) = 0$ for $x \in B_W$, and non-insulated on the right side, $\phi_{em}(x) \geq 0$ for $x \in B_E$. We distinguish three scenarios of emissions:

1. pure heat transfer as $\phi_{em}(t,x) = \phi_{tr}(t,x) = -h\ [\vartheta(t,x) - \vartheta_{amb}]$ with $h = 5$ and $\vartheta_{amb} = 300$,

2. pure heat radiation as $\phi_{em}(t,x) = \phi_{rad}(t,x) = -\sigma\varepsilon\ \vartheta(t,x)^4$ with $\varepsilon = 0.2$ and

3. heat transfer and heat radiation as $\phi_{em}(x) = \phi_{tr}(x) + \phi_{rad}(x)$ with $h = 5$, $\vartheta_{amb} = 300$ and $\varepsilon = 0.2$.

We evaluate these three emissions for a temperature range of 300 to 700 Kelvin in Figure 2.9 (a). We notice that the heat radiation plays an important role in particular for high temperatures, e.g. above 500 Kelvin. This implies that we cannot neglect the nonlinear heat radiation when we simulate heat conduction phenomena with high temperatures. In Fig. 2.9 (b), the temperature on boundary B_E of the one-dim. rod drops stronger for the heat transfer than for the heat radiation. So, the heat transfer influences mainly the cooling-down process but the heat radiation has a significant impact, too.

We notice that this example shall only demonstrate the heat transfer and heat radiation. It does not provide a qualitative statement like “heat transfer is always stronger than heat radiation” because both physical processes depend on the condition of the object and its surrounding.

3

Spatial Approximation

Partial differential equations like the heat equation need to be solved in time and space. For some simple scenarios, we are able to find an analytical solution. For example, in appendix A.1, we derive an analytical solution for the one-dim. linear heat equation with zero Neumann boundary conditions.[1] However, we usually need to find a numerical solution of the (partial) differential equation. Due to the wide range of types and specifications of partial differential equations, there exist a lot of numerical methods to solve a them: for example the well-known finite difference, finite volume and finite element methods as well as

- radial basis function methods [57, 58],
- pseudo-spectral methods [59] and
- physics-informed neural networks (PINN) [60, 61].[2,3]

[1] Insulated boundaries as $\phi(t,x) \equiv 0$.

[2] The article [60] occured also as long preprint version in two parts [62, 63].

[3] In Chapter 9, we state a short outlook on the use of PINN to solve heat conduction problems.

In this work, we approximate the integral equation (2.19) with finite volumes because it preserve the temperature-dependent heat conduction and we can implement it with a simple meshing. This spatial discretization leads to a large scale (nonlinear) ordinary differential equation (ODE) which is solved with numerical integration approaches like Runge-Kutta methods, see also Chapter 5. The finite volume approach is noted for a two-dim. model in our article [34] and implemented in *Hestia.jl*, see [35, 44].

General Formulation of the Finite Volume Method

Finite volume methods are designed originally to solve partial differential equations of the type[4]

$$\frac{\partial}{\partial t} z(t,x) + \operatorname{div}\big(f(z,t,x)\big) + g(t,x) = 0 \tag{3.1}$$

[4] Such differential equations are also denoted as conservation laws and *hyperbolic* partial differential equations [66].

for $(t,x) \in (0,T_f) \times \Omega$. The state $z : [0,T_f] \times \overline{\Omega} \to \mathbb{R}$ corresponds to a physical quantity like mass or energy, see, $f : \mathbb{R} \times [0,T_f] \times \Omega \to \mathbb{R}^d$ is called flux function with dimension $d \in \{1,2,3\}$, and $g : [0,T_f] \times \overline{\Omega} \to \mathbb{R}$ might be interpreted as a source term. We refer for a brief introduction to the online article [64] and for detailed explanations to article [65] and book [66]. We omit to specify a certain boundary condition for Eq. (3.1) here because it is less relevant for our further explanations. We integrate Eq. (3.1) over the

whole space Ω, apply the divergence theorem and obtain

$$\int_\Omega \frac{\partial}{\partial t} z(t,x) + \operatorname{div}\big(f(z,t,x)\big) + g(t,x)\, dx$$
$$= \frac{\partial}{\partial t}\int_\Omega z(t,x)\, dx + \int_{\partial\Omega} f(z,t,x)\cdot\vec{n}\, dx + \int_\Omega g(t,x)\, dx = 0 \qquad (3.2)$$

Now, we subdivide the space Ω in $N_c > 0$ *finite volumes* Ω_i and cell boundaries $\partial\Omega_i$, and we say that Eq. (3.1) holds in each finite volume Ω_i. The sum of all finite volumes is the geometry as $\Omega = \bigcup_{i=1}^{N_c} \Omega_i$. The sum of all cell boundaries is more than the boundary $\partial\Omega$ because cell boundary $\partial\Omega_i$ is the interface of each cell to its neighbors and so we find it on the boundary sides $\partial\Omega$ and inside the geometry Ω. We formulate the integral equation (3.2) for a finite volume as

$$\frac{\partial}{\partial t}\int_{\Omega_i} z(t,x)\, dx + \int_{\partial\Omega_i} f(z(t,x),t,x)\cdot\vec{n}\, dx + \int_{\Omega_i} g(t,x)\, dx = 0 \qquad (3.3)$$

with index $i \in \{1,2,\ldots,N_c\}$. An example of a single cell with its fluxes is sketched in Fig. 3.1. The finite volume Ω_i is also called *control volume* or *cell* and it might be realized via quadrilateral [67], triangular [68] and other meshing types [69]. The approximation of flux $f(z,t,x)$ at the cell boundaries $\partial\Omega_i$ is a key factor to ensure proper numerical results. We approximate each term of Eq. (3.3) and we yield the ODE

$$\frac{\partial}{\partial t}\tilde{z}_i(t) + \tilde{f}_i(\tilde{z}_i(t),t) + \tilde{g}_i(t) = 0 \qquad (3.4)$$

for the i-th finite volume with the spatial approximations $\tilde{z}_i$, $\tilde{f}_i$ and $\tilde{g}_i$.

In Section 2.3, we derived the heat equation with flux

$$f(\vartheta(t,x)) = \lambda(\vartheta(t,x))\,\nabla\vartheta(t,x)$$

to describe the heat flux inside the object, see Fourier law in Eq. (2.12), and the supplied and emitted thermal energy on the boundary sides in Eq. (2.17). As a result of this derivation, we noted the quasilinear heat equation in integral form in Eq. (2.19). We compare the quasilinear heat equation (2.19) and Eq. (3.2) and we find that the source term is zero: $g(t,x) \equiv 0$ and we need to split integral $\int_{\partial\Omega} f(z,t,x)\cdot\vec{n}dx$ into two parts: heat flux inside Ω and thermal emission and power supply on $\partial\Omega$. Hence, we derive the ODE for the inner domain of Ω in Section 3.2, and we approximate the exchange of thermal energy along the boundaries in Section 3.3 with the supplied and emitted heat flux ϕ, see also Definition 2.2.

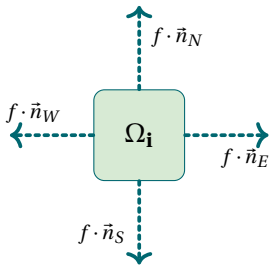


Figure 3.1: A single finite volume with flux f on cell boundaries and the outer normal vectors $\vec{n}_W$, $\vec{n}_E$, $\vec{n}_S$, $\vec{n}_N$.

3.1 Meshing with Finite Volumes

In this section, we describe the spatial approximation of the geometric shapes from Section 2.1: one-dim. rod, two-dim. rectangular, three-dim. cuboid. These objects are subdivided in many small cells and we assume that each cell contains a certain thermal energy. Such a cell is an interval in case of a rod, an area in case of a rectangular or a volume in case of a cuboid. The subsequent derivation of the finite volume approximation is explained for the three-dim. cuboid, but might be easily reduced to the

one- or two-dim. case by neglecting the corresponding dimension(s). A cuboid has a length $L > 0$, width $W > 0$ and height $H > 0$ and so we note the total volume $|\Omega| = L \cdot W \cdot H$. This total volume is subdivided in small finite volumes $\Omega_{j,m,k}$ at position $(j,m,k) \in \mathscr{J} \times \mathscr{M} \times \mathscr{K}$ with sets

$$\mathscr{J} := \{1,2,\cdots,N_j\}\ ,\ \mathscr{M} := \{1,2,\cdots,N_m\}\ ,\ \mathscr{K} := \{1,2,\cdots,N_k\}\,.$$

Along each axis we have the dimensions and the numbers of cells as

- $\Delta x_1 > 0$ and $N_j \in \mathbb{N}$ for x_1,
- $\Delta x_2 \geq 0$ and $N_m \in \mathbb{N}$ for x_2 and
- $\Delta x_3 \geq 0$ and $N_k \in \mathbb{N}$ for x_3.

Table 3.1: Size of a finite volume.

Length	$\Delta x_1 := \frac{L}{N_j}$
Width	$\Delta x_2 := \frac{W}{N_m}$
Height	$\Delta x_3 := \frac{H}{N_k}$

We note the relations in Table 3.1. These properties are reduced in the one-dim. case as $\Delta x_2 = 0$, $N_m = 1$ and $\Delta x_3 = 0$, $N_k = 1$, and in the two-dim case as $\Delta x_3 = 0$, $N_k = 1$. We find the volume of a cell as

$$|\Omega_{j,m,k}| \ = \ \Delta x_1\, \Delta x_2\, \Delta x_3 = \frac{L \cdot W \cdot H}{N_j \cdot N_m \cdot N_k} = \frac{|\Omega|}{N_c}. \tag{3.5}$$

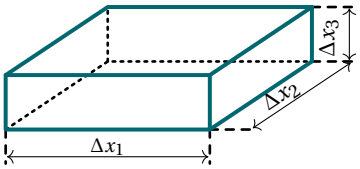


Figure 3.2: Finite volume $\Omega_{j,m,k}$

with the total number of cells $N_c = N_j \cdot N_m \cdot N_k$. We define the finite volume at position (j,m,k) as

$$\begin{aligned}\Omega_{j,m,k} := \big[j\,\Delta x_1, (j+1)\,\Delta x_1\big] &\times [m\,\Delta x_2, (m+1)\,\Delta x_2] \\ &\times [k\,\Delta x_3, (k+1)\,\Delta x_3]\,.\end{aligned} \tag{3.6}$$

and we note the corresponding position of its central point as

$$x^{j,m,k} := \begin{pmatrix} x_1^j \\ x_2^m \\ x_3^k \end{pmatrix} = \begin{pmatrix} \big[j - \frac{1}{2}\big]\,\Delta x_1 \\ \big[m - \frac{1}{2}\big]\,\Delta x_2 \\ \big[k - \frac{1}{2}\big]\,\Delta x_3 \end{pmatrix}.$$

We call a cell via its central point $x^{j,m,k}$ in the subsequently when we derive the numerical approximation of the quasilinear heat equation. The temperature values in all cells $\Omega_{j,m,k}$ are stored in a vector.[5] To call one element of this vector, we use the global identifier

$$i(j,m,k) \ = \ j + (m-1) \cdot N_j + (k-1) \cdot N_m \cdot N_j. \tag{3.7}$$

[5] We store the temperature data in vectors because we use CPU-based algorithms. In case of GPU-based computations, we recommend to store the data in matrices, tensors or multidimensional arrays.

Inversely, we find the local position (j,m,k) as

$$\begin{aligned} j &= (i-1) \mod N_j + 1 \quad , \\ m &= \frac{i-j}{N_j} \mod N_m + 1 \quad \text{and} \\ k &= \frac{i-j-(m-1)\,N_j}{N_j \cdot N_m} + 1 \end{aligned}$$

where expression mod denotes the modulo operation. A grid of finite volumes is depicted in Fig. 3.3 to exemplify the relation between position (j,m,k) and its corresponding global identifier i. In the next section, we discuss the finite volumes at following positions in detail:

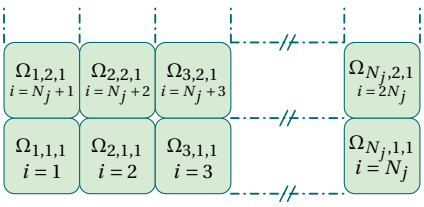


Figure 3.3: A grid of finite volumes with the relation between global index i and position (j,m,k).

Name	Symbol	j	m	k
West	$\mathscr{S}_W$	1	$\in \mathscr{M}$	$\in \mathscr{K}$
East	$\mathscr{S}_E$	N_j	$\in \mathscr{M}$	$\in \mathscr{K}$
South	$\mathscr{S}_S$	$\in \mathscr{J}$	1	$\in \mathscr{K}$
North	$\mathscr{S}_N$	$\in \mathscr{J}$	N_m	$\in \mathscr{K}$
Underside	$\mathscr{S}_U$	$\in \mathscr{J}$	$\in \mathscr{M}$	1
Topside	$\mathscr{S}_T$	$\in \mathscr{J}$	$\in \mathscr{M}$	N_k

Table 3.2: Index set of finite volumes next to boundary sides.

$$
\begin{aligned}
i(j-1,m,k) &= i(j,m,k)-1 && \text{for } j \in \{2,\dots,N_j\},\\
i(j+1,m,k) &= i(j,m,k)+1 && \text{for } j \in \{1,\dots,N_j-1\},\\
i(j,m-1,k) &= i(j,m,k)-N_j && \text{for } m \in \{2,\dots,N_m\},\\
i(j,m+1,k) &= i(j,m,k)+N_j && \text{for } m \in \{1,\dots,N_m-1\},\\
i(j,m,k-1) &= i(j,m,k)-N_j \cdot N_m && \text{for } k \in \{2,\dots,N_k\},\\
i(j,m,k+1) &= i(j,m,k)+N_j \cdot N_m && \text{for } k \in \{1,\dots,N_k-1\}.
\end{aligned}
$$

At the remaining positions, e.g. $i(j-1,m,k)$ for $j=1$, we assume "virtual" cells to derive the approximated boundary conditions in Section 3.3.

We distinguish the cells inside the object versus the cells at the boundary sides. The index set of all finite volumes is defined by

$$\mathscr{S} := \left\{ i(j,m,k) \mid j \in \mathscr{J},\ m \in \mathscr{M},\ k \in \mathscr{K} \right\} \tag{3.8}$$

and the indices of inner domain are stored as

$$\mathring{\mathscr{S}} := \left\{ i(j,m,k) \mid j \in \mathscr{J} \setminus \{1,N_j\},\ m \in \mathscr{M} \setminus \{1,N_m\},\ k \in \mathscr{K} \setminus \{1,N_k\} \right\}.$$

Consequently, the set of indices of all cells next to the boundary sides is found as $\mathscr{S} \setminus \mathring{\mathscr{S}}$. Table 3.2 lists the index sets for each boundary side separately. We remark that the i-th index may occur in multiple sets of the boundary sides because the corners and edges intersect, which means

$$
\begin{aligned}
(\mathscr{S}_W \cap \mathscr{S}_S) \cup (\mathscr{S}_W \cap \mathscr{S}_N) \cup (\mathscr{S}_W \cap \mathscr{S}_U) \cup (\mathscr{S}_W \cap \mathscr{S}_U) \neq \{\} \text{ and}\\
(\mathscr{S}_E \cap \mathscr{S}_S) \cup (\mathscr{S}_E \cap \mathscr{S}_N) \cup (\mathscr{S}_E \cap \mathscr{S}_U) \cup (\mathscr{S}_E \cap \mathscr{S}_U) \neq \{\}.
\end{aligned}
$$

We have the cardinality of the index set as

$$|\mathscr{S}| = N_j\ N_m\ N_k = N_c.$$

This box-shaped meshing with finite volumes of the same size provides us an intuitive approach to approximate the heat equation in the next section. Though, we need to remark that this approach leads to high computational costs because the number of cells grow cubically, see the cardinality above. We visualize the finite volumes as two-dim. boxes in the next sections but we consider small three-dim. cuboids.

3.2 The Finite Volume Method

We consider a physical quantity $z : [0, T_{final}] \times \overline{\Omega} \to \mathbb{R}_{\geq 0}$, which represents for example the thermal energy or temperature. We consider the value of z in the cell (j, m, k) as the average

$$z(t, x^{j,m,k}) = \frac{1}{|\Omega_{j,m,k}|} \int_{x_3^{k-\frac{1}{2}}}^{x_3^{k+\frac{1}{2}}} \int_{x_2^{m-\frac{1}{2}}}^{x_2^{m+\frac{1}{2}}} \int_{x_1^{j-\frac{1}{2}}}^{x_1^{j+\frac{1}{2}}} z(t,x) dx_1 dx_2 dx_3$$
$$= \frac{1}{|\Omega_{j,m,k}|} \int_{\Omega_{j,m,k}} z(t,x) dx \tag{3.9}$$

with $|\Omega_{j,m,k}|$ as in Eq. (3.5). This averaging approach is visualized in Fig. 3.4, and it is applied on the Eq. (2.19) to yield the integral form of the heat equation in each cell (j, m, k) as

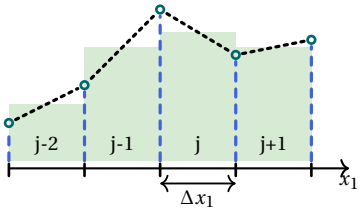


Figure 3.4: Averaging in one-dim. finite volume as in Eq. (3.9).

$$\underbrace{\frac{1}{|\Omega_{j,m,k}|} \int_{\Omega_{j,m,k}} \rho(\vartheta)\, c(\vartheta)\, \frac{\partial}{\partial t} \vartheta(t,x)\, dx}_{\frac{d}{dt} U_{j,m,k}(t)} =$$
$$\underbrace{\frac{1}{|\Omega_{j,m,k}|} \int_{\Omega_{j,m,k}} \operatorname{div}\left[\lambda(\vartheta)\, \nabla \vartheta(t,x)\right]\, dx}_{\frac{d}{dt} Q_{j,m,k}(t)}$$
$$+ \underbrace{\frac{1}{|\Omega_{j,m,k}|} \int_{\partial\Omega_{j,m,k}} \left[\lambda(\vartheta)\, \nabla \vartheta(t,x)\right] \cdot \vec{n}\, dx}_{P_{j,m,k}(t)}. \tag{3.10}$$

We see in Eq. (3.10) that the first and second term, $\frac{d}{dt} U_{j,m,k}(t)$ and $\frac{d}{dt} Q_{j,m,k}(t)$ affect all cells, but the third term $P_{j,m,k}(t)$ only affects boundary cells, see Table 3.2. This implies that $P_{j,m,k}(t) \equiv 0$ for all cells of the inner domain.

The left-hand side of Eq. (3.10), describes only the variation in time and not in space. Therefore, we find its approximation as

$$\frac{1}{|\Omega_{j,m,k}|} \int_{\Omega_{j,m,k}} \rho(\vartheta(t,x))\, c(\vartheta(t,x))\, \frac{\partial}{\partial t} \vartheta(t,x) dx$$
$$\approx \rho(\vartheta(t,x))\, c(\vartheta(t,x))\, \frac{\partial}{\partial t} \vartheta(t,x) \Big|_{x = x^{j,m,k}}. \tag{3.11}$$

In the next step, we evaluate the term $\frac{d}{dt} Q_{j,m,k}(t)$ on the right-hand side of Eq. (3.10). We recapitulate from Section 2.3 that we derived $\frac{d}{dt} Q$ with the heat flux $\dot{q}$ and divergence

$$\operatorname{div}(\dot{q}(t,x)) = \sum_{l=1}^{3} \frac{\partial}{\partial x_l} \dot{q}(t,x)$$

in Eq. (2.13). Here, we approximate the derivatives $\frac{\partial}{\partial x_l} \dot{q}_l$ for axis $l \in \{1,2,3\}$ in a first step and afterwards we approximate the temperature gradient

$\frac{\partial}{\partial x_l}\vartheta(t,x)$ in

$$\dot{q}_l(\vartheta(t,x)) = -\lambda_l(\vartheta(t,x))\frac{\partial}{\partial x_l}\vartheta(t,x), \tag{3.12}$$

see also Eq. (2.12).

We find the finite volume approach of $\frac{d}{dt}Q(t)$ in Eq. (3.10) as

$$\begin{aligned}
&\frac{1}{|\Omega_{j,m,k}|}\int_{\Omega_{j,m,k}} \operatorname{div}[\lambda(\vartheta(t,x))\nabla\vartheta(t,x)]\,dx \\
&\quad= \frac{-1}{|\Omega_{j,m,k}|}\int_{\Omega_{j,m,k}} \operatorname{div}\left[\dot{q}(\vartheta(t,x))\right]dx \\
&\quad= \frac{-1}{|\Omega_{j,m,k}|}\int_{\Omega_{j,m,k}} \sum_{l=1}^{3}\frac{\partial}{\partial x_l}\dot{q}_l(\vartheta(t,x))dx \\
&\quad= \frac{-1}{|\Omega_{j,m,k}|}\sum_{l=1}^{3}\int_{\Omega_{j,m,k}} \frac{\partial}{\partial x_l}\dot{q}_l(\vartheta(t,x))dx.
\end{aligned} \tag{3.13}$$

In accordance with the fundamental theorem of calculus, we solve the latter integral as

$$\begin{aligned}
&\int_{\Omega_{j,m,k}} \frac{\partial}{\partial x_l}\dot{q}_l(\vartheta(t,x))dx \\
&\quad= \Delta x_{l_1}\,\Delta x_{l_2}\left[\dot{q}_l(\vartheta\left(t,\tilde{x}+\frac{\delta x_l}{2}\right)) - \dot{q}_l(\vartheta\left(t,\tilde{x}-\frac{\delta x_l}{2}\right))\right]
\end{aligned} \tag{3.14}$$

with the central point $\tilde{x} := x^{j,m,k}$, distance $\delta x_l = \Delta x_l\, e_l$ and standard basis vector $e_l \in \mathbb{R}^3$ for $l \in \{1,2,3\}$ and indices

$$l_1 := [l \mod 3] + 1 \quad \text{and} \quad l_2 := [(l+1) \mod 3] + 1,$$

which determine orthogonal directions of e_l. We see that $\Delta x_{l_1}\cdot\Delta x_{l_2}$ denotes an area and we find $\frac{\Delta x_{l_1}\cdot\Delta x_{l_2}}{|\Omega_{j,m,k}|} = \frac{1}{\Delta x_l}$. We continue our ideas from Eq. (3.13) with the latest findings in Eq. (3.14) as

$$\begin{aligned}
&\frac{-1}{|\Omega_{j,m,k}|}\sum_{l=1}^{3}\int_{\Omega_{j,m,k}} \frac{\partial}{\partial x_l}\dot{q}_l(\vartheta(t,x))dx \\
&\quad= -1\sum_{l=1}^{3}\frac{1}{\Delta x_l}\left[\dot{q}_l(\vartheta\left(t,\tilde{x}+\frac{\delta x_l}{2}\right)) - \dot{q}_l(\vartheta\left(t,\tilde{x}-\frac{\delta x_l}{2}\right))\right]
\end{aligned}$$

and we replace $\dot{q}_l$ as in Eq. (3.12) to yield

$$\begin{aligned}
&\frac{1}{|\Omega_{j,m,k}|}\int_{\Omega_{j,m,k}} \operatorname{div}[\lambda(\vartheta(t,x))\nabla\vartheta(t,x)]\,dx \\
&\quad= \sum_{l=1}^{3}\frac{1}{\Delta x_l}\left[\lambda_l\left(\vartheta\left(t,\tilde{x}+\frac{\delta x_l}{2}\right)\right)\quad\frac{\partial}{\partial x_l}\vartheta\left(t,\tilde{x}+\frac{\delta x_l}{2}\right)\right. \\
&\qquad\left.-\lambda_l\left(\vartheta\left(t,\tilde{x}-\frac{\delta x_l}{2}\right)\right)\quad\frac{\partial}{\partial x_l}\vartheta\left(t,\tilde{x}-\frac{\delta x_l}{2}\right)\right]
\end{aligned} \tag{3.15}$$

The derivative $\frac{\partial}{\partial x_l}\vartheta\left(t,\tilde{x}\pm\frac{\delta x_l}{2}\right)$ in Eq. (3.15) is approximated with a centered *finite difference* approach as

$$\frac{\partial}{\partial x_l}f(x) = \frac{1}{\Delta x_l}\left[f\left(x+\frac{\delta x_l}{2}\right) - f\left(x-\frac{\delta x_l}{2}\right)\right] + \mathcal{O}(\|\Delta x_l\|^2)\,,$$

which we derive via Taylor series approximation, see [71, p. 3], and we find the finite difference approximation at position $x \pm \frac{\delta x_l}{2}$ as

$$\frac{\partial}{\partial x_l} f\left(x \pm \frac{\delta x_l}{2}\right) \approx \frac{1}{\Delta x_l}\left[f\left(x \pm \delta x_l\right) \mp f\left(x\right)\right]. \tag{3.16}$$

We apply the finite difference stencil in Eq. (3.16) on derivative $\frac{\partial}{\partial x_l}\vartheta$ in Eq. (3.15) and we conclude

$$\begin{aligned}
&\frac{1}{|\Omega_{j,m,k}|}\int_{\Omega_{j,m,k}} \operatorname{div}\left[\lambda(\vartheta(t,x))\nabla\vartheta(t,x)\right] dx \\
&\approx \sum_{l=1}^{3} \frac{1}{\Delta x_l^2}\Bigg[\lambda_l(\vartheta\left(t,\tilde{x}+\frac{\delta x_l}{2}\right))\,\vartheta\left(t,\tilde{x}+\delta x_l\right) \\
&\qquad +\lambda_l(\vartheta\left(t,\tilde{x}-\frac{\delta x_l}{2}\right))\,\vartheta\left(t,\tilde{x}-\delta x_l\right) \\
&\qquad -\left[\lambda_l(\vartheta\left(t,\tilde{x}+\frac{\delta x_l}{2}\right)) + \lambda_l(\vartheta\left(t,\tilde{x}-\frac{\delta x_l}{2}\right))\right]\,\vartheta(t,\tilde{x})\Bigg].
\end{aligned} \tag{3.17}$$

We do not have access to the temperature $\vartheta(t,\tilde{x}\pm\frac{\delta x_l}{2})$, which occur inside the thermal conductivity in Eq. (3.17), and so we approximate it via

$$\vartheta\left(t,\tilde{x}\pm\frac{\delta x_l}{2}\right) = \frac{1}{2}\left[\vartheta(t,\tilde{x}) + \vartheta(t,\tilde{x}\pm\delta x_l)\right].$$

To improve the readability, we change the notation from position (j,m,k) to global identifier $i(j,m,k)$, see Eq. (3.7), and we note the cell temperatures as

$$\Theta_i(t) := \vartheta\left(t,\tilde{x}\right) \quad \text{and} \quad \Theta_{i\pm\mu}(t) := \vartheta\left(t,\tilde{x}\pm\delta x_l\right) \tag{3.18}$$

with offset

$$\mu = \begin{cases} 1 & \text{if } l=1, \\ N_j & \text{if } l=2, \\ N_j\cdot N_m & \text{if } l=3. \end{cases} \tag{3.19}$$

Temperatures with index $i\pm\mu$ are geometrically adjacent to the i-th temperature as portrayed in Fig. 3.5, but for $l\in\{2,3\}$ in Eq. (3.19) they are not adjacent in the vector of stored temperatures

$$\Theta := \Bigg(\underbrace{\Theta_1,\ldots,\Theta_{N_j}}_{m=1},\underbrace{\Theta_{N_j+1},\ldots,\Theta_{2N_j}}_{m=2},\ldots,,\Theta_{m\,N_j},\ldots,\Theta_{N_jN_m}\Bigg)^{\top}.$$

Furthermore, we define

$$\tilde{\lambda}_l(w_1,w_2) := \lambda_l([w_1+w_2]/2) \tag{3.20}$$

as the thermal conductivity along a cell boundary and we note

$$\lambda_l(\vartheta\left(t,\tilde{x}\pm\frac{\delta x_l}{2}\right)) \approx \lambda_l\left(\frac{\vartheta\left(t,\tilde{x}\right)+\vartheta\left(t,\tilde{x}\pm\delta x_l\right)}{2}\right) := \tilde{\lambda}_l(\Theta_i,\Theta_{i\pm\mu}).$$

Consequently, we formulate Eq. (3.17) in terms of Θ_i, $\Theta_{i\pm\mu}$ and $\tilde{\lambda}$ as

$$\begin{aligned}
&\frac{1}{|\Omega_{j,m,k}|}\int_{\Omega_{j,m,k}} \operatorname{div}\left[\lambda(\vartheta(t,x))\nabla\vartheta(t,x)\right] dx \\
&\approx \sum_{l=1}^{3} \frac{1}{\Delta x_l^2}\left[\tilde{\lambda}_l(\Theta_i,\Theta_{i+\mu})\,\Theta_{i+\mu}(t) + \tilde{\lambda}_l(\Theta_i,\Theta_{i-\mu})\,\Theta_{i-\mu}(t)\right. \\
&\qquad \left. -\left[\tilde{\lambda}_l(\Theta_i,\Theta_{i+\mu}) + \tilde{\lambda}_l(\Theta_i,\Theta_{i-\mu})\right]\,\Theta_i(t)\right].
\end{aligned} \tag{3.21}$$

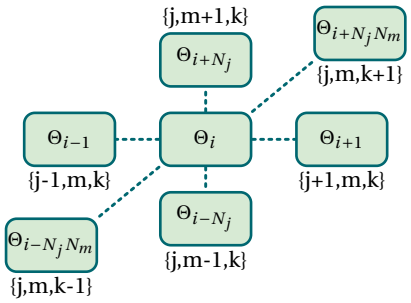


Figure 3.5: Neighboring temperatures of the i-th cell inside the object.

Summarizing the results of the spatial approximation in Eq. (3.11) and (3.21), we find diffusion for all cells of the inner domain $i \in \mathring{\mathcal{S}}$ as

$$\begin{aligned}\rho(\Theta_i)\, c(\Theta_i)\, \frac{\partial}{\partial t}\Theta_i(t) = \\ \sum_{l=1}^{3} \frac{1}{\Delta x_l^2} \big[\tilde{\lambda}_l(\Theta_i, \Theta_{i+\mu})\, \Theta_{i+\mu}(t) + \tilde{\lambda}_l(\Theta_i, \Theta_{i-\mu})\, \Theta_{i-\mu}(t) \\ - \big[\tilde{\lambda}_l(\Theta_i, \Theta_{i+\mu}) + \tilde{\lambda}_l(\Theta_i, \Theta_{i-\mu})\big]\, \Theta_i(t)\big]. \end{aligned} \tag{3.22}$$

3.3 Spatial Approximation of Boundary Conditions

The thermal dynamics inside the cuboid is described by Eq. (3.22) in terms of $\frac{d}{dt}U_{j,m,k}(t)$ and $\frac{d}{dt}Q_{j,m,k}(t)$ in Eq. (3.10). Additionally, we need to describe the influence along the boundary sides with $P_{j,m,k}(t)$ as in Eq. (3.10), because the temperatures $\Theta_{i+\mu}$ and $\Theta_{i-\mu}$ are not known for $i \in S_E \cup S_N \cup S_T$ and $i \in S_W \cup S_S \cup S_U$, respectively. For this purpose, we assume virtual (or ghost) cells outside, which are adjacent to the cuboid as depicted in Fig. 3.6 and 3.7. So, we calculate the temperature gradients between the inner cell and the virtual cell, see Definition 2.1. We begin with the boundary condition

$$\lambda(\vartheta(t,x))\nabla\vartheta(t,x)\cdot\vec{n}|_{x=\partial\Omega} = \phi(t,x)$$

where $\phi : [0,T]\times\partial\Omega \to \mathbb{R}$ represents the supplied and emitted energy flux as noted in Definition 2.2.

The outer normal vector $\vec{n}$ is orthogonal to the boundary side and is positive if it is parallel to x_1, x_2 or x_3, and negative if it is antiparallel to these directions. Accordingly, we note the gradients on the boundary sides as

$$\nabla\vartheta(t,x)\cdot\vec{n} = \begin{cases} -\frac{\partial\vartheta(t,x)}{\partial x_1} & \text{for } x \in B_W, \\ \frac{\partial\vartheta(t,x)}{\partial x_1} & \text{for } x \in B_E, \\ -\frac{\partial\vartheta(t,x)}{\partial x_2} & \text{for } x \in B_S, \\ \frac{\partial\vartheta(t,x)}{\partial x_2} & \text{for } x \in B_N, \\ -\frac{\partial\vartheta(t,x)}{\partial x_3} & \text{for } x \in B_U, \\ \frac{\partial\vartheta(t,x)}{\partial x_3} & \text{for } x \in B_T. \end{cases}$$

We approximate the boundary condition with finite differences in case of the negative outer normal vector as

$$\lambda_l(\vartheta(t,\tilde{x}-\delta x_l/2))\, \frac{1}{\Delta x_l}\, [\vartheta(t,\tilde{x}-\delta x_l) - \vartheta(t,\tilde{x})] = \phi_l(t,\tilde{x}) \tag{3.23a}$$

for $(\tilde{x} - \delta x_l/2) \in B_W \cup B_S \cup B_U$; and in case of the positive outer normal vector as

$$\lambda_l(\vartheta(t,\tilde{x}+\delta x_l/2))\, \frac{1}{\Delta x_l}\, [\vartheta(t,\tilde{x}+\delta x_l) - \vartheta(t,\tilde{x})] = \phi_l(t,\tilde{x}) \tag{3.23b}$$

for $(\tilde{x} + \delta x_l/2) \in B_E \cup B_N \cup B_T$. In Eq. (3.23), we consider ϕ_l at position $\tilde{x}$ because we claim

$$\phi_l(t,\tilde{x}\mp\delta x_l/2) \approx \phi_l(t,\tilde{x}).$$

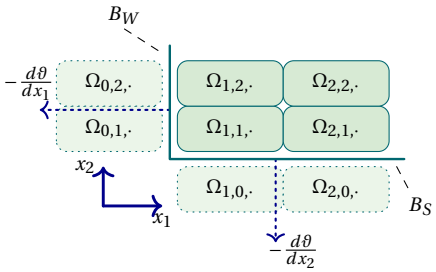


Figure 3.6: Cells next to boundary sides B_W and B_S for an arbitrary k-th index. The temperature gradients $-\frac{d\vartheta}{dx_1}$ and $-\frac{d\vartheta}{dx_2}$ are antiparallel to x_1 and x_2.

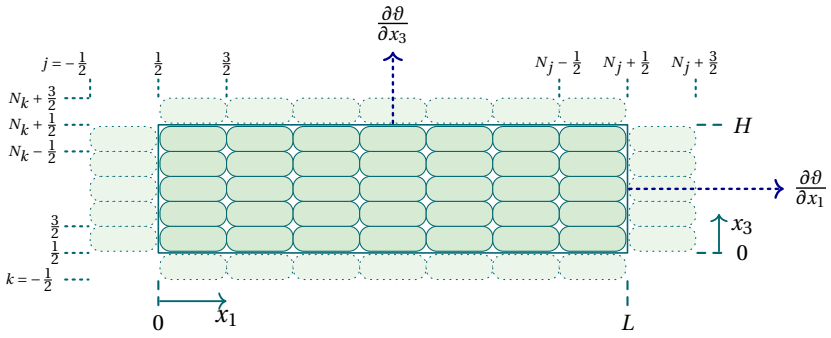


Figure 3.7: Side view on the cuboid, on boundary side B_S, with finite volume cells inside object and virtual cells outside. The vectors $\frac{\partial\vartheta}{\partial x_1}$ and $\frac{\partial\vartheta}{\partial x_3}$ represent the temperature gradients on boundary sides B_E and B_T.

Furthermore, we distinguish ϕ_l for axis $l \in \{1,2,3\}$ because cells with two or three boundary surfaces have different fluxes for each side. Now, we step over to the spatially discrete case where all nodes $\tilde{x} = x^i = x^{j,m,k}$ are inside the object. Here, we lose the unique relation between the position on the boundary and its outer normal vector because a cell may have two or three boundary sides. Thus, we need a decision variable to connect the position and its associated direction as

$$\text{pos}(l,i) = \begin{cases} 1 & \text{if } (l,i) \in \{1\}\times\mathscr{S}_W \cup \{2\}\times\mathscr{S}_S \cup \{3\}\times\mathscr{S}_U, \\ 2 & \text{if } (l,i) \in \{1\}\times\mathscr{S}_E \cup \{2\}\times\mathscr{S}_N \cup \{3\}\times\mathscr{S}_T, \\ 0 & \text{else.} \end{cases}$$

We identify the cell temperatures as $\Theta_i(t) = \vartheta(t, \tilde{x} \pm \delta x_l)$, see Eq. (3.18) and the averaged thermal conductivity in Eq. (3.20), and we formulate

$$\Theta_{i-\mu}(t) \;=\; \Theta_i(t) + \frac{\Delta x_l \;\phi_l(t,x^i)}{\tilde{\lambda}_l(\Theta_i,\Theta_{i-\mu})} \tag{3.24a}$$

with $x^i = x^{(j,m,k)}$, μ as in Eq. (3.19), $l \in \{1,2,3\}$ and all $i \in S_W \cup S_S \cup S_U$, which guarantee $\text{pos}(l,i) = 1$ and

$$\Theta_{i+\mu}(t) \;=\; \Theta_i(t) + \frac{\Delta x_l \;\phi_l(t,x^i)}{\tilde{\lambda}_l(\Theta_i,\Theta_{i+\mu})} \tag{3.24b}$$

for $l \in \{1,2,3\}$ and $i \in S_E \cup S_N \cup S_T$ such that $\text{pos}(l,i) = 2$. The approximation of the supplied and emitted heat flux for the l-th direction can be condensed as a mapping

$$\begin{aligned} \phi_1 &: [0, T_{final}) \to \mathbb{R}^{2N_mN_k} \quad , \\ \phi_2 &: [0, T_{final}) \to \mathbb{R}^{2N_jN_k} \quad \text{and} \\ \phi_3 &: [0, T_{final}) \to \mathbb{R}^{2N_jN_m} \quad . \end{aligned}$$

with the heat flux vectors[6] as

[6] We drop the time-dependency of $\phi_l(t)$ for a better readability.

$$\phi_1 = \begin{pmatrix} \phi_{1,i(1,1,1)} \\ \phi_{1,i(N_j,1,1)} \\ \hline \phi_{1,i(1,2,1)} \\ \phi_{1,i(N_j,2,1)} \\ \hline \vdots \\ \hline \phi_{1,i(1,m,k)} \\ \phi_{1,i(N_j,m,k)} \\ \hline \vdots \\ \hline \phi_{1,i(1,N_m,N_k)} \\ \phi_{1,i(N_j,N_m,N_k)} \end{pmatrix}, \quad \phi_2 = \begin{pmatrix} \phi_{2,i(1,1,1)} \\ \cdots \\ \phi_{2,i(N_j,1,1)} \\ \hline \phi_{2,i(1,N_m,1)} \\ \cdots \\ \phi_{2,i(N_j,N_m,1)} \\ \hline \vdots \\ \hline \phi_{2,i(1,1,k)} \\ \cdots \\ \phi_{2,i(N_j,1,k)} \\ \hline \phi_{2,i(1,N_m,k)} \\ \cdots \\ \phi_{2,i(N_j,N_m,k)} \\ \hline \vdots \\ \hline \phi_{2,i(1,1,N_k)} \\ \cdots \\ \phi_{2,i(N_j,1,N_k)} \\ \hline \phi_{2,i(1,N_m,N_k)} \\ \cdots \\ \phi_{2,i(N_j,N_m,N_k)} \end{pmatrix} \quad \text{and} \quad \phi_3 = \begin{pmatrix} \phi_{3,i(1,1,1)} \\ \cdots \\ \phi_{3,i(j,m,1)} \\ \cdots \\ \phi_{3,i(N_j,N_m,1)} \\ \hline \phi_{3,i(1,1,N_k)} \\ \cdots \\ \phi_{3,i(j,m,N_k)} \\ \cdots \\ \phi_{3,i(N_j,N_m,N_k)} \end{pmatrix} \tag{3.25}$$

in which $\phi_{l,i(j,m,k)}$ is a short notation for

$$\phi_{l,i(j,m,k)} \hat{=} \phi_l(t, x^{j,m,k}) = \phi_l(t, x^i)$$

with global index $i = i(j,m,k)$. If ϕ_l represents the thermal emissions as noted in Def. 2.3, we note the approximated heat flux as

$$\phi_{em,l}(t, x^i) := -h_l(x^i)\,[\Theta_i(t) - \vartheta_{amb,l}(x^i)] - \sigma\,\varepsilon_l(x^i)\,\Theta_i(t)^4 \tag{3.26}$$

and we distinguish here h_l, ϵ_n and $\vartheta_{amb,l}$ for each direction $l \in \{1,2,3\}$. According to Eq. (3.22), the diffusion in the l-th direction is approximated by

$$\left[\tilde{\lambda}_l(\Theta_i, \Theta_{i+\mu})\,\Theta_{i+\mu}(t) + \tilde{\lambda}_l(\Theta_i, \Theta_{i-\mu})\,\Theta_{i-\mu}(t) - \left[\tilde{\lambda}_l(\Theta_i, \Theta_{i+\mu}) + \tilde{\lambda}_l(\Theta_i, \Theta_{i-\mu})\right]\,\Theta_i(t)\right] / \Delta x_l^2$$

and we identify the unknown temperatures at $i \pm \mu$ with the identities (3.24). We find the diffusion in each direction $l \in \{1,2,3\}$ as

$$\mathscr{D}_l(\Theta_i, \Theta_{i-\mu}, \Theta_{i+\mu}) := \begin{cases} \tilde{\lambda}_l(\Theta_i, \Theta_{i+\mu})\,\left(\Theta_{i+\mu} - \Theta_i\right) / \Delta x_l^2 & \text{if } \mathrm{pos}(l,i) = 1, \\ \tilde{\lambda}_l(\Theta_i, \Theta_{i-\mu})\,\left(\Theta_{i-\mu} - \Theta_i\right) / \Delta x_l^2 & \text{if } \mathrm{pos}(l,i) = 2, \\ \left[\tilde{\lambda}_l(\Theta_i, \Theta_{i+\mu})\Theta_{i+\mu} + \tilde{\lambda}_l(\Theta_i, \Theta_{i-\mu})\Theta_{i-\mu}\right. & \\ \qquad \left. - \left[\tilde{\lambda}_l(\Theta_i, \Theta_{i+\mu}) + \tilde{\lambda}_l(\Theta_i, \Theta_{i-\mu})\right]\Theta_i\right] / \Delta x_l^2 & \text{else} \end{cases} \tag{3.27}$$

and we note the "external" processes on the boundary

$$\mathscr{E}_l(t, x^i) = \begin{cases} \phi_l(t, x^i) / \Delta x_l & \text{if } \mathrm{pos}(l,i) \in \{1,2\}, \\ 0 & \text{else.} \end{cases} \tag{3.28}$$

We conclude this section by summarizing the numerical approximation of the quasilinear heat conduction in the following definition.

Definition 3.1 (Spatially approximated quasilinear heat conduction)
We consider an object with length $L > 0$, width $W \geq$ and height $H \geq 0$. This object is discretized with $N_c = N_j \cdot N_m \cdot N_k$ finite volumes with the dimensions $\Delta x_1 = \frac{L}{N_j}$, $\Delta x_2 = \frac{W}{N_m}$ and $\Delta x_3 = \frac{H}{N_k}$, see Table 3.1. We note the cell temperatures as $\Theta \in \mathbb{R}^{N_c}$ and we approximate the thermal conductivity at the cell boundaries $\tilde{\lambda}_l$ as in Eq. (3.20). The left-hand side of the quasilinear heat equation (2.20) is approximated as in Eq. (3.11). The diffusion inside the object Ω as in Eq. (3.22) is equipped with the boundary conditions and the temperatures in the virtual cells is found as in Eq. (3.24). In conclusion, we formulate the spatially approximated quasilinear heat conduction as

$$c(\Theta_i)\,\rho(\Theta_i)\,\frac{d}{dt}\Theta_i(t) = \sum_{l=1}^{3}\Big[\mathscr{D}_l(\Theta_i,\Theta_{i-\mu},\Theta_{i+\mu}) + \mathscr{E}_l(t,x^i)\Big]. \tag{3.29}$$

with offset μ in Eq. (3.19), approximated diffusion $\mathscr{D}_l$ in Eq. (3.27) and "external" processes $\mathscr{E}_l$ in (3.28). ○

3.4 Sparse Representation of the Linear System

In the end of Section 2.3, we introduced the linear heat equation (2.21) with constant material properties $\lambda = \text{diag}(\lambda_1,\lambda_2,\lambda_3)$, $\rho > 0$ and $c > 0$. Here, we approximate the linear heat equation with the finite volume approach and we note the ODE (3.29) in matrix-vector notation as

$$c\,\rho\,\frac{d}{dt}\Theta(t) = \sum_{l=1}^{N_d}\frac{\lambda_l}{\Delta x_l^2}D_l\,\Theta(t) + E_l\frac{\phi_l(t)}{\Delta x_l} \tag{3.30}$$

with $N_d \in \{1,2,3\}$, diffusion matrices $D_l \in \mathbb{R}^{N_c\times N_c}$, temperature vector $\Theta : [0,T_{final}] \to \mathbb{R}^{N_c}$ and number of finite volume cells $N_c = N_j \cdot N_m \cdot N_k$. The approximated boundary conditions are specified by

$$\begin{aligned}
E_1 &\in \mathbb{R}^{N_c\times 2N_mN_k}, & \phi_1 &: [0,T_{final}) \to \mathbb{R}^{2N_mN_k},\\
E_2 &\in \mathbb{R}^{N_c\times 2N_jN_k}, & \phi_2 &: [0,T_{final}) \to \mathbb{R}^{2N_jN_k},\\
E_3 &\in \mathbb{R}^{N_c\times 2N_jN_m}, & \phi_3 &: [0,T_{final}) \to \mathbb{R}^{2N_jN_m}.
\end{aligned}$$

The heat flux vectors ϕ_l are specified in Eq. (3.25), and the sparse[7] matrices D_l and E_l are described in detail next. We recommend to compare the definition of the index set in Eq. (3.8) and the corresponding Table 3.2 for boundary cells to follow the subsequent ideas. In Eq. (3.27) with $\mu = 1$, we have the diffusion at the boundary sides as

$$\begin{aligned}
\Theta_{i+1} - \Theta_i &= (-1,1)\cdot\begin{pmatrix}\Theta_i\\ \Theta_{i+1}\end{pmatrix} \quad \text{for } i \in \mathscr{S}_W,\\
\Theta_{i-1} - \Theta_i &= (1,-1)\cdot\begin{pmatrix}\Theta_{i-1}\\ \Theta_i\end{pmatrix} \quad \text{for } i \in \mathscr{S}_E,
\end{aligned}$$

and elsewhere as

$$\Theta_{i+1} + \Theta_{i-1} - 2\Theta_i = (1,-2,1)\cdot\begin{pmatrix}\Theta_{i-1}\\ \Theta_i\\ \Theta_{i+1}\end{pmatrix}.$$

[7] The term *sparse* means that a vector or matrix consists of many zero entries. This fact may be used to reduce the storage or to accelerate the computation of a matrix-vector multiplication.

We formulate these iterations as the matrix

$$\tilde{D}_1 = \begin{array}{c} j: \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \dots & N_j \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -1 \end{pmatrix} \end{array}, \tag{3.31}$$

which is stacked to form the diffusion along direction x_1 as

$$D_1 = \operatorname{diag}(\underbrace{\tilde{D}_1, \dots, \tilde{D}_1}_{N_m N_k \text{ blocks}}).$$

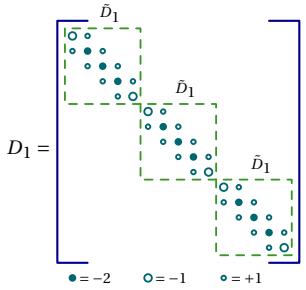


Figure 3.8: Sparse pattern of matrix D_1 to express the diffusion in x_1-direction.

The sparse pattern of matrix D_1 is visualized in Fig. 3.8. The position of flux ϕ_1 at $\mathscr{S}_W$ and $\mathscr{S}_E$ corresponds to $j = 1$ and $j = N_j$ and so we note the matrices

$$\tilde{E}_1 = \begin{pmatrix} 1 & 0 \\ 0 & \vdots \\ \vdots & 0 \\ 0 & 1 \end{pmatrix} \quad , \quad E_1 = \operatorname{diag}(\underbrace{\tilde{E}_1, \dots, \tilde{E}_1}_{N_m N_k \text{ blocks}}).$$

We continue with direction x_2 and we find the boundary conditions in Eq. (3.27) with $\mu = N_j$ as

$$\Theta_{i+N_j} - \Theta_i = (-1, 0_{N_j-1}, 1) \begin{pmatrix} \Theta_i \\ \Theta_{i+1} \\ \vdots \\ \Theta_{i+N_j} \end{pmatrix} \quad \text{for } i \in \mathscr{S}_S \,,$$

$$\Theta_{i-N_j} - \Theta_i = (1, 0_{N_j-1}, -1) \begin{pmatrix} \Theta_{i-N_j} \\ \vdots \\ \Theta_{i-1} \\ \Theta_i \end{pmatrix} \quad \text{for } i \in \mathscr{S}_N$$

and for all other indices $i \in \mathscr{S} \setminus \mathscr{S}_S \cup \mathscr{S}_N$, we note the diffusion

$$\Theta_{i+N_j} + \Theta_{i-N_j} - 2\Theta_i = (1, 0_{N_j-1}, -2, 0_{N_j-1}, 1) \begin{pmatrix} \Theta_{i-N_j} \\ \vdots \\ \Theta_{i-1} \\ \Theta_i \\ \Theta_{i+1} \\ \vdots \\ \Theta_{i+N_j} \end{pmatrix}.$$

We iterate over $j \in \{1, \dots, N_j\}$, $m \in \{1\} \cup \{2, \dots, N_m - 1\} \cup \{N_m\}$ to yield the

matrix blocks

$$\tilde{D}_2 = \begin{array}{c} m: \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \dots & N_m \end{array} \\ \begin{pmatrix} -I_{N_j} & I_{N_j} & 0_{N_j} & \dots & 0_{N_j} \\ I_{N_j} & -2I_{N_j} & I_{N_j} & \ddots & \vdots \\ 0_{N_j} & \ddots & \ddots & \ddots & 0_{N_j} \\ \vdots & \ddots & I_{N_j} & -2I_{N_j} & I_{N_j} \\ 0_{N_j} & \dots & 0_{N_j} & I_{N_j} & -I_{N_j} \end{pmatrix} \end{array}, \tag{3.32}$$

which are summarized via an iteration over $k \in \{1, \dots, N_k\}$ as

$$D_2 = \text{diag}(\underbrace{\tilde{D}_2, \dots, \tilde{D}_2}_{N_k \text{ blocks}}).$$

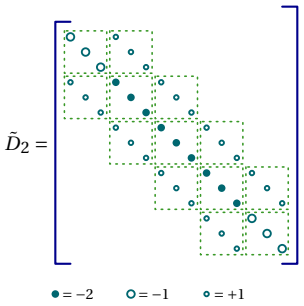


Figure 3.9: Sparse pattern of matrix $D_{2,k}$ to express the diffusion in x_2-direction in layer $k \in \{1, \dots, N_k\}$.

An example of the sparse pattern of the sub-matrices $D_{2,k}$ is expressed in Fig. 3.9. The heat fluxes occur at $m = 1$ and $m = N_m$ for N_j cells in each "layer" $k \in \{1, \dots, N_k\}$ and so we note the matrices

$$\tilde{E}_2 = \begin{pmatrix} I_{N_j} & 0_{N_j} \\ 0_{N_j} & \vdots \\ \vdots & 0_{N_j} \\ 0_{N_j} & I_{N_j} \end{pmatrix} \quad , \quad E_2 = \text{diag}(\underbrace{\tilde{E}_2, \dots, \tilde{E}_2}_{N_k \text{ blocks}}).$$

The diffusion in x_3-direction is noted in Eq. (3.27) with $\mu = N_j \cdot N_m$ for the boundary sides as

$$\Theta_{i+N_j N_m} - \Theta_i = (-1, 0_{N_j N_m - 1}, 1) \begin{pmatrix} \Theta_i \\ \Theta_{i+1} \\ \vdots \\ \Theta_{i+N_j N_m} \end{pmatrix} \quad \text{for } i \in \mathscr{S}_U \,,$$

$$\Theta_{i-N_j N_m} - \Theta_i = (1, 0_{N_j N_m - 1}, -1) \begin{pmatrix} \Theta_{i-N_j N_m} \\ \vdots \\ \Theta_{i-1} \\ \Theta_i \end{pmatrix} \quad \text{for } i \in \mathscr{S}_T$$

and we find for all other indices $i \in \mathscr{S} \setminus \mathscr{S}_U \cup \mathscr{S}_T$

$$\Theta_{i+N_j N_m} + \Theta_{i-N_j N_m} - 2\Theta_i = (1, 0_{N_j N_m - 1}, -2, 0_{N_j N_m - 1}, 1) \begin{pmatrix} \Theta_{i-N_j N_m} \\ \vdots \\ \Theta_{i-1} \\ \Theta_i \\ \Theta_{i+1} \\ \vdots \\ \Theta_{i+N_j N_m} \end{pmatrix}.$$

The boundary conditions are active at "layer" $k = 1$ and $k = N_k$ for all $(j, m) \in \{1, \dots, N_j\} \times \{1, \dots, N_m\}$, and for all other layers $k \in \{2, \dots, N_k - 1\}$ we have the diffusion matrix blocks $(I_{N_j N_m}, -2I_{N_j N_m}, I_{N_j N_m})$. Consequently,

we note the matrices

$$D_3 = \begin{array}{c} k: \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \dots & N_k \end{array} \\ \begin{pmatrix} -I_{N_jN_m} & I_{N_jN_m} & 0_{N_jN_m} & \dots & 0_{N_jN_m} \\ I_{N_jN_m} & -2I_{N_jN_m} & I_{N_jN_m} & \ddots & \vdots \\ 0_{N_jN_m} & \ddots & \ddots & \ddots & 0_{N_jN_m} \\ \vdots & \ddots & I_{N_jN_m} & -2I_{N_jN_m} & I_{N_jN_m} \\ 0_{N_jN_m} & \dots & 0_{N_jN_m} & I_{N_jN_m} & -I_{N_jN_m} \end{pmatrix} \end{array} \tag{3.33}$$

and

$$E_3 = \begin{pmatrix} I_{N_jN_m} & 0_{N_jN_m} \\ 0_{N_jN_m} & \vdots \\ \vdots & 0_{N_jN_m} \\ 0_{N_jN_m} & I_{N_jN_m} \end{pmatrix}.$$

We see that D_l and E_l with $l \in \{1,2,3\}$ are large-scale matrices with only few nonzero entries and so its summation in Eq. (3.30) leads to a large-scale sparse matrix again.

Due to these large-scale and sparse matrices, the evaluation of Eq. (3.30) should not be implemented as s matrix vector operations in a CPU-based computation because of potentially high computational costs. However, the linear system formulation in Eq. (3.30) provides a suitable form to analyze the eigenvalues, eigenvectors, and related properties like the analytical and numerical stability, stiffness, etc. of the linear system in Chapter 4. Furthermore, we consider the linear system for the design of the open-loop and closed-loop control in Chapter 7 and 8.

We conclude this section by summarizing the diffusion matrices D_1, D_2 and D_3 to formulate the system matrix A for the one-, two- and three-dim. case.

Definition 3.2 (State space formulation of the free system)
We consider the spatially approximated heat equation (3.30) with thermally insulated boundary sides and without actuation, e.g. $\phi_l(t) \equiv 0$ for $l \in \{1,2,3\}$. We denote the state space formulation as

$$\frac{d}{dt}\Theta(t) = A_{N_d}\Theta(t) \tag{3.34}$$

with $N_d \in \{1,2,3\}$, system matrix

$$A_{N_d} := \sum_{l=1}^{N_d} \frac{\alpha_l}{\Delta x_l^2} D_l \tag{3.35}$$

and diffusivity $\alpha_l := \frac{\lambda_l}{c\,\rho}$. We distinguish the temperature vector for each geometry as temperature states

$$\begin{aligned} &\Theta : [0, T_{final}] \to \mathbb{R}^{N_j} && \text{if } N_d = 1\,, \\ &\Theta : [0, T_{final}] \to \mathbb{R}^{N_jN_m} && \text{if } N_d = 2 \text{ and} \\ &\Theta : [0, T_{final}] \to \mathbb{R}^{N_jN_mN_k} && \text{if } N_d = 3. \end{aligned}$$

Subsequently, we formulate A_{N_d} for each $N_d \in \{1,2,3\}$. In the one-dim.

case ($N_d = 1$), we find the tridiagonal system matrix

$$A_1 = \frac{\alpha_1}{\Delta x_1^2}\tilde{D}_1 = \frac{\alpha_1}{\Delta x_1^2}\begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{N_j \times N_j} \tag{3.36}$$

with the tridiagonal matrix $\tilde{D}_1$ as in Eq. (3.31).We formulate the state-space of the two-dim. heat equation ($N_d = 2$) as

$$\begin{aligned} A_2 &= \frac{\alpha_1}{\Delta x_1^2}\,\mathrm{diag}(\underbrace{\tilde{D}_1, \ldots, \tilde{D}_1}_{N_m \text{ blocks}}) + \frac{\alpha_2}{\Delta x_2^2}\tilde{D}_2 \\ &= \begin{pmatrix} \tilde{A}_{2,0} & \tilde{A}_{2,2} & & & \\ \tilde{A}_{2,2} & \tilde{A}_{2,1} & \tilde{A}_{2,2} & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{A}_{2,2} & \tilde{A}_{2,1} & \tilde{A}_{2,2} \\ & & & \tilde{A}_{2,2} & \tilde{A}_{2,0} \end{pmatrix} \in \mathbb{R}^{N_j N_m \times N_j N_m} \end{aligned} \tag{3.37}$$

with $\tilde{D}_2$ as in Eq. (3.32) and the matrix blocks

$$\begin{aligned} \tilde{A}_{2,0} &= A_1 - \frac{\alpha_2}{\Delta x_2^2} I = \frac{\alpha_1}{\Delta x_1^2}\tilde{D}_1 - \frac{\alpha_2}{\Delta x_2^2} I \quad , \\ \tilde{A}_{2,1} &= A_1 - \frac{2\,\alpha_2}{\Delta x_2^2} I = \frac{\alpha_1}{\Delta x_1^2}\tilde{D}_1 - \frac{2\,\alpha_2}{\Delta x_2^2} I \quad \text{and} \\ \tilde{A}_{2,2} &= \frac{\alpha_2}{\Delta x_2^2} I. \end{aligned}$$

We continue our previous ideas for the three-dim. scenario ($N_d = 3$) and we formulate the system matrix as

$$\begin{aligned} A_3 &= \frac{\alpha_1}{\Delta x_1^2} D_1 + \frac{\alpha_2}{\Delta x_2^2} D_2 + \frac{\alpha_3}{\Delta x_3^2} D_3 \\ &= \begin{pmatrix} \tilde{A}_{3,0} & \tilde{A}_{3,2} & & & \\ \tilde{A}_{3,2} & \tilde{A}_{3,1} & \tilde{A}_{3,2} & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{A}_{3,2} & \tilde{A}_{3,1} & \tilde{A}_{3,2} \\ & & & \tilde{A}_{3,2} & \tilde{A}_{3,0} \end{pmatrix} \in \mathbb{R}^{N_j N_m N_k \times N_j N_m N_k} \end{aligned} \tag{3.38}$$

with $\tilde{D}_2$ as in Eq. (3.32) and the matrix blocks

$$\begin{aligned} \tilde{A}_{3,0} &= A_2 - \frac{\alpha_3}{\Delta x_3^2} I \quad , \\ &= \frac{\alpha_1}{\Delta x_1^2}\,\mathrm{diag}(\underbrace{\tilde{D}_1, \ldots, \tilde{D}_1}_{N_m \text{ blocks}}) + \frac{\alpha_2}{\Delta x_2^2}\tilde{D}_2 - \frac{\alpha_3}{\Delta x_3^2} I \quad , \\ \tilde{A}_{3,1} &= A_2 - \frac{2\,\alpha_3}{\Delta x_3^2} I \\ &= \frac{\alpha_1}{\Delta x_1^2}\,\mathrm{diag}(\underbrace{\tilde{D}_1, \ldots, \tilde{D}_1}_{N_m \text{ blocks}}) + \frac{\alpha_2}{\Delta x_2^2}\tilde{D}_2 - \frac{2\,\alpha_3}{\Delta x_3^2} I \quad \text{and} \\ \tilde{A}_{3,2} &= \frac{\alpha_3}{\Delta x_3^2} I. \end{aligned}$$

○

4

Approximated Linear System

In this chapter, we discuss the system properties of the linear heat conduction phenomena as described in Definition 3.2. In particular, we compute the eigenvalues[1] $\mu_n \in \mathbb{C}$ and eigenvectors $\psi \in \mathbb{C}^N$ of the system matrix A_{N_d} in Eq. (3.34) for $N_d \in \{1,2,3\}$. A general linear differential equations

[1] The common eigenvalue symbol λ is reserved for the thermal conductivity.

$$\frac{d}{dt}z(t) = Az(t) \tag{4.1}$$

with states $z : [0, T_f] \to \mathbb{R}^N$, system matrix $A \in \mathbb{R}^{N\times N}$ and number of states $N \in \mathbb{N}$, represents a heat conduction problem with zero-Neumann boundary conditions as in Eq. (3.34). In this setting, we have a system matrix $A := A_{N_d}$, states or temperature values $z(t) := \Theta(t)$ and the number states of finite volume cells $N := N_c$. We find the eigenvalues μ_n of matrix A in Eq. (4.1) by solving the well-known eigenvalue problem

$$A\psi = \mu\psi \quad \text{or equivalently} \quad (A-\mu I)\psi = 0. \tag{4.2}$$

In case of small-scale systems, e.g. $N \in \{1,2,3,4\}$, we may find the eigenvalues through manually solving the characteristic polynomial

$$p(\mu) := \det(A-\mu I) \overset{!}{=} 0.$$

In case of larger systems, such a computation is usually much more complicated for arbitrary matrices and need be evaluated numerically. However, in some cases we may derive the eigenvalues directly: for example in case of diagonal and upper or lower triangular matrices as

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,N} \\ & a_{1,2} & a_{2,3} & \dots & a_{2,N} \\ & & \ddots & & \vdots \\ & & & a_{N-1,N-1} & a_{N-1,N} \\ & & & & a_{N,N} \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} a_{1,1} & & & & \\ a_{2,1} & a_{2,2} & & & \\ a_{3,1} & a_{3,2} & \ddots & & \\ \vdots & \vdots & \ddots & a_{N-1,N-1} & \\ a_{N,1} & a_{N,2} & \dots & a_{N,N-1} & a_{N,N} \end{pmatrix}$$

we yield the eigenvalues as diagonal entries $\mu_n = a_{n,n}$. If matrix A is not in a triangular or diagonal form then it may be transformed to such a form.

System matrix A_{N_d} is not diagonal or triangular, but it is a tridiagonal matrix for $N_d = 1$ and a tridiagonal block matrix for $N_d \in \{2,3\}$, see Definition 3.2. Thus, the eigenvalues are not the diagonal elements and they need to be calculated numerically as discussed in Section 4.1. In Section

4.2, we analyze the matrix properties of A_{N_d} and the system behavior of the related differential equation using the found eigenvalues and eigenvectors. We continue these ideas in order to derive the solution of differential equation (3.30) in Section 4.3 and we exemplify our findings with small-scale simulations.

4.1 Computation of Eigenvalues and Eigenvectors

In Definition 3.2, we formulated the system matrix A_{N_d} for each number of dimension $N_d \in \{1,2,3\}$. In the one-dim. case, we have $A_1 = \frac{\alpha_1}{\Delta x_1^2}\tilde{D}_1$ and we see that $\tilde{D}_1$ contains ones on the upper an lower sub-diagonal and -2 on almost all diagonal elements, except the first and last row and column. We call such a matrix shape tridiagonal. In the two- and three-dim. cases, the system matrices A_2 and A_3 contain matrix blocks on the diagonal and sub-diagonal and these matrix blocks have a similar tridiagonal shape as in the one-dim. case. These matrices do not match the previously described triangular or diagonal form, which have the eigenvalues as diagonal entries. In general, these matrices are too large to calculate the eigenvalues manually and so they are usually computed numerically. Standard eigenvalue solvers provide useful results, but they prone to small numerical errors and the computational costs increase by the matrix size. In this section, we provide an approach to compute the eigenvalues and eigenvectors exactly for one-, two- and three-dim. geometries. To reach this goal, we firstly estimate the range of eigenvalues with the Gershgorin Circle Theorem and secondly, we compute the eigenvalue and eigenvectors exactly with cosine expressions. We prove the correctness of the found eigenvalues and eigenvectors and we show that the eigenvalues are in fact inside the Gershgorin circles.

Gershgorin Circle Theorem

We approximate the eigenvalues of an arbitrary matrix $A \in \mathbb{R}^{N\times N}$ with the Gershgorin circle theorem[2], see also [71, p. 277]. For the further explanations, we have an eigenvalue μ and the related eigenvector $\psi = (\psi_1,\dots,\psi_N)^\top$, which is normed as

[2] These ideas are based on the work of Semyon Aronovich Gershgorin (*1901, †1933) [70].

$$\|\psi\|_\infty = \max(|\psi_1|,\dots,|\psi_N|) = 1$$

such that its largest element is one at index i with $i \in \{1,2,\dots,N\}$. From Eq. (4.2), we derive for each row

$$\mu\psi_i = \sum_{j=1}^{N} a_{i,j}\psi_j = \left[\sum_{j=1\wedge j\neq i}^{N} a_{i,j}\psi_j\right] + a_{i,i}\psi_i$$

and we subtract the i-th component on the right side as

$$(\mu - a_{i,i})\psi_i = \sum_{j=1\wedge j\neq i}^{N} a_{i,j}\psi_j. \tag{4.3}$$

Next, we consider the absolute value on both sides and see that

$$|\mu - a_{i,i}||\psi_i| = |\mu - a_{i,i}|$$

because $|\psi_i| = 1$. We apply the triangle inequality on Eq. (4.3) and we note

$$|\mu - a_{i,i}| \leq \left| \sum_{j=1 \wedge j \neq i}^{N} a_{i,j} \psi_j \right| \leq \sum_{j=1 \wedge j \neq i}^{N} |a_{i,j}||\psi_j| \leq \sum_{j=1 \wedge j \neq i}^{N} |a_{i,j}|$$

because $|\psi_j| \leq 1$ for $j \neq i$. We define the radius of the i-th diagonal element $a_{i,i}$ of matrix A as

$$r_i := \sum_{j=1 \wedge j \neq i}^{N} |a_{i,j}|$$

and the related Gershgorin discs as

$$d(z, r) := \{\xi \in \mathbb{C} : |\xi - z| < r\}$$

which implies $|\mu - a_{i,i}| = d(a_{i,i}, r_i)$. Therefore, we find all eigenvalue μ_n inside or on the boundary of the union of all Gershgorin discs as

$$\mu_n \in \bigcup_{i}^{N} \overline{d(a_{i,i}, r_i)} \quad \text{for } n \in \{1, \ldots, N\}.$$

This result is called Gershgorin circle theorem. We illustrate this concept with the small example matrix[3]

$$A = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{pmatrix},$$

which has the Gershgorin discs $d_1(-1, 1)$ for the first and last row and $d_2(-2, 2)$ for the second row. We find the true eigenvalues of A as the roots of the characteristic polynomial

$$p(\mu) = \det(\mu - A) = (\mu + 2)\,(\mu + 1)\,\mu$$

as $\mu \in \{-3, -1, 0\}$. All eigenvalues are inside the union of the closed Gershgorin discs as $\mu \in \overline{d_1(-2, 2)} \cup \overline{d_2(-2, 2)}$, and the Gershgorin discs and the eigenvalues are visualized in Fig. 4.1.

If matrix $A \in \mathbb{R}^{N \times N}$ is decomposable as $A = \sum_{n=1} p_n M_n$ with coefficient $p_n \in \mathbb{R}$ and $M_n \in \mathbb{R}^{N \times N}$, then we find the Gershgorin discs

$$d(a_{i,i}, r_i) = d\left(\sum_{n=1} p_n m_{n,i,i}, r_i \right)$$

with radius

$$r_i = \sum_{n=1} p_n \sum_{j=1 \wedge j \neq i} |m_{n,i,j}|$$

in which $m_{n,i,j}$ denotes the (i, j)-th entry of matrix M_n. In this way, we apply the Gershgorin circle theorem on each (partial) diffusion matrix $\tilde{D}_1$, $\tilde{D}_2$ and D_3 as in Eq. (3.31, 3.32, 3.33) separately and we find for each matrix the diagonal entries -1 and -2 and the radius $r = 1$ and $r = 2$. Thus, we note the same Gershgorin discs $d(-1, 1)$ and $d(-2, 2)$ for each (partial) diffusion matrix, see also Fig. 4.1 above. In case of a full system matrix A_{N_d} as in Eq. (3.35), we find four scenarios for the diagonal entries and the corresponding radii, which depend on the index $i \in \mathscr{S}$ of the temperature cell. If the cell is completely inside the object as $i \in \mathring{\mathscr{S}}$, then we note the diagonal entries and its radius as

$$a_{N_d,i,i} = -2 \sum_{l=1}^{N_d} \frac{\alpha_l}{\Delta x_l^2} \quad \text{and} \quad r_i = 2 \sum_{l}^{N_d} \frac{\alpha_l}{\Delta x_l^2}.$$

[3] Example matrix A equals to matrix $D_{1,n}$ in identity (3.31) for $N_j = 3$.

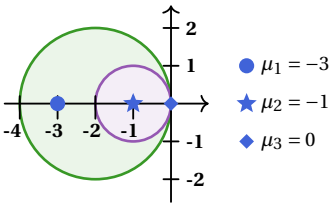


Figure 4.1: Gershgorin discs $d(-1, 1)$, $d(-2, 2)$ and the true eigenvalues $\mu \in \{-3, -1, 0\}$ of the example matrix A.

with diffusivity $\alpha_l = \frac{\lambda_l}{c\,\rho}$. If a cell is close to a boundary side, then we note the index of the corresponding direction as

$$\tilde{l} := \begin{cases} 1 & \text{if } i \in \mathscr{S}_W \cup \mathscr{S}_E, \\ 2 & \text{if } i \in \mathscr{S}_S \cup \mathscr{S}_N, \\ 3 & \text{if } i \in \mathscr{S}_U \cup \mathscr{S}_T. \end{cases}$$

If a cell is close to one boundary side, then we find diagonal entries and its radius

$$a_{N_d,i,i} = \left[-2 \sum_{l \neq \tilde{l}} \frac{\alpha_l}{\Delta x_l^2} + (-1)\, \frac{\alpha_{\tilde{l}}}{\Delta x_{\tilde{l}}^2} \right], \; r_i = \left[2 \sum_{l \neq \tilde{l}} \frac{\alpha_l}{\Delta x_l^2} + \frac{\alpha_{\tilde{l}}}{\Delta x_{\tilde{l}}^2} \right].$$

If a cell is close to two boundary sides, then we continue with

$$a_{N_d,i,i} = \left[-2 \frac{\alpha_l}{\Delta x_l^2} + (-1) \sum_{\tilde{l} \neq l} \frac{\alpha_{\tilde{l}}}{\Delta x_{\tilde{l}}^2} \right], \; r_i = \left[2 \frac{\alpha_l}{\Delta x_l^2} + \sum_{\tilde{l} \neq l} \frac{\alpha_{\tilde{l}}}{\Delta x_{\tilde{l}}^2} \right]$$

and if a cell is close to three boundary sides, then we note

$$a_{N_d,i,i} = -\sum_{\tilde{l}=1}^{N_d} \frac{\alpha_{\tilde{l}}}{\Delta x_{\tilde{l}}^2} \quad \text{and} \quad r_i = \sum_{\tilde{l}=1}^{N_d} \frac{\alpha_{\tilde{l}}}{\Delta x_{\tilde{l}}^2}.$$

We exemplify these findings with a simple three-dim. heat conduction example. We assume the material properties $\lambda_n = 1$ and $c = \rho = 1$, and the spatial discretization $\Delta x_n = 1$ for $n \in \{1,2,3\}$. So, we note the diffusion matrix

$$A_{N_d} = D_1 + D_2 + D_3$$

for an arbitrary size of $D^{N_c \times N_c}$ with $N_c \geq 9$. In accordance with the previous ideas, we derive the Gershgorin discs

$$d_4(-6,6)\,, \quad d_3(-5,5)\,, \quad d_2(-4,4)\,, \quad d_1(-3,3)$$

and we see that the smallest possible eigenvalue is at $\mu_{min} = -12$ and the largest possible eigenvalue is at $\mu_{max} = 0$.

Eigenvalues and Eigenvectors in the One-Dimensional Case

We continue with the exact computation of eigenvalues and eigenvectors for the one-dim. linear heat equation. We return to our standard notation of the spatially approximated linear one-dim. heat equation

$$\frac{d}{dt} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_{N_j-1} \\ \Theta_{N_j} \end{pmatrix} = p_1 \underbrace{\begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}}_{=:D_1} \begin{pmatrix} \Theta_1 \\ \Theta_2 \\ \vdots \\ \Theta_{N_j-1} \\ \Theta_{N_j} \end{pmatrix} \tag{4.4}$$

with insulated boundary sides, coefficient $p_1 := \frac{\alpha}{\Delta x^2}$ and initial conditions $\Theta(0) = \Theta_0$, see also Def. 3.2. Here, we have diffusion matrix $D_1 = \tilde{D}_1$ and we notice that D_1 looks *almost* like a Toeplitz matrix[4]

[4] Described by and named after Otto Toeplitz (*1881 , †1940) [72].

$$\begin{pmatrix} a_0 & a_{-1} & a_{-2} & \dots & a_{-(N_j-1)} \\ a_1 & a_0 & a_{-1} & \dots & a_{-(N_j-2)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & a_1 & a_0 & a_{-1} \\ a_{N_j-1} & \dots & a_2 & a_1 & a_0 \end{pmatrix},$$

which has only non-zero entries a_0 on the diagonal, and a_{-1} on the upper and a_1 on the lower sub-diagonal. We remark that D_1 is only almost a Toeplitz matrix because a_0 in the first and last row differ to the other diagonal elements. The eigenvalue computation of Toeplitz matrices in general[5] and tridiagonal Toeplitz matrices in particular is well studied in the literature, see [73–75]. The eigenvalues of a tridiagonal Toeplitz matrix

$$A = \begin{pmatrix} b & c & & & \\ a & b & c & & \\ & \ddots & \ddots & \ddots & \\ & & a & b & c \\ & & & a & b \end{pmatrix} \tag{4.5}$$

are noted as[6]

$$\mu_j = b - 2\sqrt{ac}\cos\left(\frac{n\pi}{N_j+1}\right)$$

for $j \in \{1,\dots,N_j\}$. We choose $(a,b,c) = (1,-2,1)$ to note a matrix which looks *almost* like the diffusion matrix D_1 in Eq. (3.31) and we find the eigenvalues

$$\mu_j = -2 - 2\cos\left(\frac{j\pi}{N_j+1}\right) \tag{4.6}$$

for $j \in \{1,\dots,N_j\}$. The position of the eigenvalues in Eq. (4.6) for $N = 10$ are portrayed in Fig. 4.2. However, the eigenvalues in Eq. (4.6) are *not exactly* the eigenvalues of diffusion matrix D_1 in Eq. (4.4) because matrix D_1 is not exactly a tridiagonal Toeplitz matrix as the first and the last diagonal entry of D_1 are not -2 but -1 due to the Neumann boundary condition.

We take these differences of the diagonal elements into account and we note the tridiagonal matrix

$$A = \begin{pmatrix} b-\alpha & c & & & \\ a & b & c & & \\ & \ddots & \ddots & \ddots & \\ & & a & b & c \\ & & & a & b-\beta \end{pmatrix}. \tag{4.7}$$

According to article [78], we find the eigenvalues of the tridiagonal matrix in Eq. (4.7) with $\alpha = \beta = -\sqrt{ac} \neq 0$[7] as

$$\mu_j = b + 2\sqrt{ac}\cos\left(\frac{(j-1)\pi}{N_j}\right)$$

for $j \in \{1,\dots,N_j\}$. The corresponding eigenvectors

$$\psi_j = (\psi_{j,1},\dots,\psi_{j,n_j},\dots,\psi_{j,N_j})^\top$$

[5] More details about Toeplitz matrices are noted in [76].

[6] There exist different ways how to derive these eigenvalues, see also this blog post on StackExchange [77].

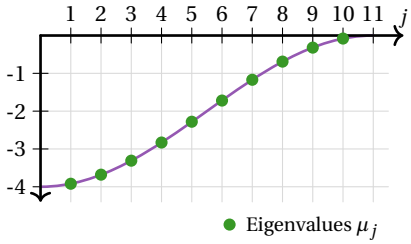


Figure 4.2: Continuous version of the eigenvalue distribution as in Eq. (4.6) for $N = 10$. The discrete eigenvalues μ_j for $j \in \{1,\dots,10\}$ are noted as red dots.

[7] Coefficient α is **not** the diffusivity here.

have the elements

$$\psi_{j,n_j} = \varrho^{n_j-1} \cos\left(\frac{(j-1)(2n_j-1)\pi}{2N_j}\right) \tag{4.8}$$

with $\varrho = \sqrt{a/c}$, see [78]. We remark that these eigenvectors ψ_j are not normalized. This eigenvalue and eigenvector computation is also discussed and extended in article [79]. Now, we choose again $(a,b,c) = (1,-2,1)$ and $\alpha = \beta = -1$ such that we formulate diffusion matrix D_1 as in Eq. (4.4). In this way, we yield the eigenvalues of the one-dimensional linear system as

$$\mu_j = -2 + 2\cos\left(\frac{(j-1)\pi}{N_j}\right) \tag{4.9}$$

for $j \in \{1,\ldots,N_j\}$. An example of the eigenvalue distribution for $N = 10$ is visualized in Fig. 4.3. We highlight that all eigenvalues are inside the interval $[-4,0]$ as computed previously with the Gershgorin discs. We note the eigenvectors elements with Eq. (4.8) as

$$\psi_{j,n_j} = \cos\left(\frac{(j-1)(2n_j-1)\pi}{2N_j}\right) \tag{4.10}$$

for $j \in \{1,\ldots,N_j\}$ because $\varrho = \sqrt{\frac{a}{c}} = \sqrt{1} = 1$. We highlight the case of $j = 1$, where we have eigenvalue $\mu_1 = 0$ and eigenvector $\psi_1 = (1,\ldots,1)^\top$. In Fig. 4.4, we the visualize the eigenvector elements ψ_{j,n_j} of Eq. (4.10) for $N_j = 10$ and $N_j \in \{2,5,7,10\}$.

In the original one-dim. linear heat conduction problem in Eq. (4.4), the diffusion matrix is multiplied with coefficient $p_1 = \frac{\alpha_1}{\Delta x^2}$. Hence, we need to include p_1 in equation (4.9) as

$$\mu_j = -2\frac{\alpha}{\Delta x^2}\left[1 - \cos\left(\frac{(j-1)\pi}{N_j}\right)\right]. \tag{4.11}$$

So, we find all eigenvalues to be inside the interval $[-4\frac{\alpha}{\Delta x^2}, 0]$. This fact implies that the linear differential equation (4.4) is analytically stable in the sense of Lyapunov for all choices of $\alpha > 0$ and $\Delta x > 0$. The single zero eigenvalue μ_1 does not disturb the stability property practically. We summarize our findings on the computation of eigenvalues and eigenvectors in the subsequent lemma and we prove that they solve the eigenvalue problem (4.2).

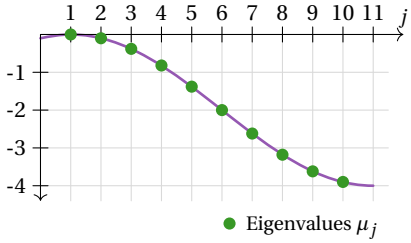


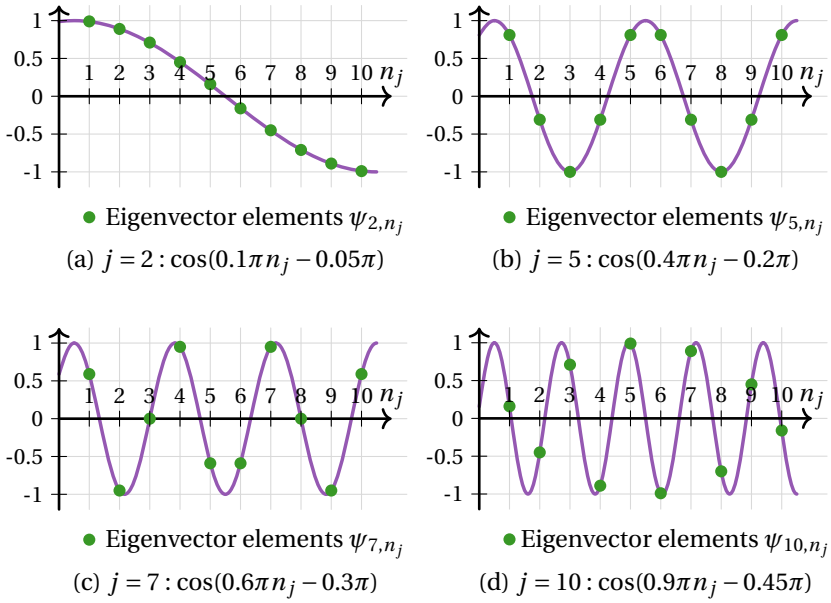
Figure 4.3: Continuous version of the eigenvalue distribution as in Eq. (4.9) for $N = 10$. The discrete eigenvalues of diffusion matrix D_1 are μ_n for $j \in \{1,\ldots,10\}$, which noted as red dots.

Lemma 4.1

The values μ_j in Eq. (4.11) and the vectors $\psi_j = (\psi_{j,1},\ldots,\psi_{j,N_j})^\top$ with ψ_{j,n_j} in Eq. (4.10) solve the eigenvalue problem

$$A_1\psi_j = \mu_j\psi_j \tag{4.12}$$

with matrix A_1 from Eq. (3.36) for $j \in \{1,\ldots,N_j\}$.



(a) $j = 2 : \cos(0.1\pi n_j - 0.05\pi)$ (b) $j = 5 : \cos(0.4\pi n_j - 0.2\pi)$

(c) $j = 7 : \cos(0.6\pi n_j - 0.3\pi)$ (d) $j = 10 : \cos(0.9\pi n_j - 0.45\pi)$

Figure 4.4: Eigenvector elements ψ_{j,n_j} (green dots) and underlying cosine oscillation as in Eq.(4.10) with $N_j = 10$ for $j \in \{2,5,7,10\}$.

Proof. We consider a matrix

$$A_1 = p_1 \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -1 \end{bmatrix}$$

with $p_1 = \frac{\alpha_1}{\Delta x_1^2}$. In this proof, we check that the right-hand side of Eq. (4.12) coincides with its left-hand side. For this evaluation, we transform expression $\mu_j \psi_j$ via angle sum identities to the left-hand side expression $A_1 \psi_j$. In the beginning of this proof, we collect several identities, which help us to evaluate the term $\mu_j \psi_j$. We use these identities in the next proof again.

We introduce function

$$f(z,n) := \cos([2n-1]z)\,, \tag{4.13}$$

which is used to express the eigenvector elements

$$\psi_{j,n_j} = f(v,n_j) = \cos\left([2n_j - 1]v\right)$$

and the eigenvalues

$$\mu_j = -2p_1[1 - f(2v,1)] = -2p_1[1 - \cos(2v)]$$

with $v = (j-1)\frac{\pi}{2N_j}$ and $j \in \{1,\dots,N_j\}$. We note the cosine angle sum identities

$$2\cos(v)\cos(w) = \cos(v+w) + \cos(v-w), \tag{4.14}$$

$$2\sin(v)\sin(w) = -\cos(v+w) + \cos(v-w) \tag{4.15}$$

and $\cos(-v) = \cos(v)$ for $v, w \in \mathbb{R}$. We apply Eq. (4.14) on $f(z,n)$ and we find the identity

$$\begin{aligned} 2\, f(2z,1)\, f(z,n) &= 2\cos(2z)\, \cos([2n-1]z) \\ &= \cos([2n+1]z) + \cos([2n-3]z) \\ &= f(z,n+1) + f(z,n-1). \end{aligned} \tag{4.16}$$

We evaluate the term $f(z,n-1)$ for $n=1$ as

$$f(z,0) = \cos(-z) = \cos(z) = f(z,1). \tag{4.17}$$

We continue with $n = N_j$: we identify z in $f(z,n)$ by $v = (j-1)\frac{\pi}{2N_j}$ and we calculate

$$\begin{aligned} -f(v,N_j) + f(v,N_j+1) &= -\cos([2N_j-1]v) + \cos([2N_j+1]v) \\ &\overset{\text{Eq. (4.15)}}{=} 2\sin(2N_j v)\sin(v) \\ &= 2\sin\big([j-1]\pi\big)\, \sin(v) \equiv 0 \end{aligned}$$

for all $j \in \{1,\dots,N_j\}$. Hence, we have the terminal condition

$$f(z,N_j+1) = f(z,N_j) \tag{4.18}$$

Now we have all necessary identities at hand and we check the eigenvalue equations.

In the first row of the eigenvalue equation (4.12), we calculate

$$\begin{aligned} \mu_j \psi_{j,1} &= -2p_1[1-f(2v,1)]f(v,1) \\ &= p_1[-f(v,1) - f(v,1) + 2f(2v,1)f(v,1)] \\ &= p_1[-f(v,1) - f(v,1) + f(v,2) + \underbrace{f(v,0)}_{\overset{\text{Eq. (4.17)}}{=} f(v,1)}\ ] \\ &= p_1[-f(v,1) + f(v,2)] \\ &= p_1[-\psi_{j,1} + \psi_{j,2}] \end{aligned}$$

which equals the left-hand side of the first row. In the second row, we calculate

$$\begin{aligned} \mu_j \psi_{j,2} &= -2p_1[1-f(2v,1)]f(v,2) \\ &= p_1[-2f(v,2) + 2f(2v,1)f(v,2)] \\ &= p_1[-2f(v,1) + f(v,3) + f(v,1)] \\ &= p_1[\psi_{j,1} - 2\psi_{j,2} + \psi_{j,3}] \end{aligned}$$

and in the n-th row with $n \in \{2,\dots,N_j-1\}$, we note analog as above

$$\begin{aligned} \mu_j \psi_{j,n} &= -2p_1[1-f(2v,1)]f(v,n) \\ &= p_1[-2f(v,n) + 2f(2v,1)f(v,n)] \\ &= p_1[-2f(v,n) + f(v,n+1) + f(v,n-1)] \\ &= p_1[\psi_{j,n-1} - 2\psi_{j,n} + \psi_{j,n+1}]. \end{aligned}$$

Finally, we note in the last row, we find

$$\begin{aligned} \mu_j \psi_{j,N_j} &= -2p_1[1-f(2v,1)]f(v,N_j) \\ &= p_1[-f(v,N_j) - f(v,N_j) + 2f(2v,1)f(v,N_j)] \\ &= p_1[-f(v,N_j) - f(v,N_j) + f(v,N_j+1) + f(v,N_j-1)] \end{aligned}$$

and we apply here Eq. (4.18) to obtain

$$\begin{aligned}\mu_j \psi_{j,N_j} &= p_1[-f(v,N_j)+f(v,N_j-1)]\\ &= p_1[\psi_{j,N_j-1}-\psi_{j,N_j}].\end{aligned}$$

In consequence to these findings, all rows of the right-hand side of Eq. (4.12) coincide with its left-hand side. □

Two-dimensional Heat Conduction

In the one-dim. case we found the eigenvalues and eigenvectors of the tridiagonal system matrix A_1. In the next step, we transfer these ideas to the two- and three-dim. heat conduction scenarios where A_2 and A_3 are block triangular matrices as described in Section 3.4. These system matrices describe a diffusion for each spatial direction and so we take this superposition into account for our subsequent discussions. In accordance with Definition 3.2, we note the approximated two-dim. heat equation as

$$\frac{d}{dt}\Theta(t) = \underbrace{\left[p_1 \begin{pmatrix} \tilde{D}_1 & & & & \\ & \tilde{D}_1 & & & \\ & & \ddots & & \\ & & & \tilde{D}_1 & \\ & & & & \tilde{D}_1 \end{pmatrix} + p_2 \begin{pmatrix} -I & I & & & \\ I & -2I & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & -2I & I \\ & & & I & -I \end{pmatrix} \right]}_{=:A_2} \Theta(t) \tag{4.19}$$

with one-dim. diffusion matrix $\tilde{D}_1$ as in Eq. (3.31) and the coefficients $p_1 := \frac{\alpha_1}{\Delta x_1^2}$ and $p_2 := \frac{\alpha_2}{\Delta x_2^2}$. System matrix $A_2 \in \mathbb{R}^{N_j N_m \times N_j N_m}$ is noted in Definition 3.2 as block tridiagonal matrix

$$A_2 = \begin{pmatrix} \tilde{A}_{2,0} & \tilde{A}_{2,2} & & & \\ \tilde{A}_{2,2} & \tilde{A}_{2,1} & \tilde{A}_{2,2} & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{A}_{2,2} & \tilde{A}_{2,1} & \tilde{A}_{2,2} \\ & & & \tilde{A}_{2,2} & \tilde{A}_{2,0} \end{pmatrix},$$

which has *almost* a tridiagonal block Toeplitz shape as

$$T = \begin{pmatrix} T_A & T_B & & & \\ T_B & T_A & T_B & & \\ & \ddots & \ddots & \ddots & \\ & & T_B & T_A & T_B \\ & & & T_B & T_A \end{pmatrix} \tag{4.20}$$

with matrix blocks

$$T_A = \begin{pmatrix} a_0 & a_1 & & & \\ a_1 & a_0 & a_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_1 & a_0 & a_1 \\ & & & a_1 & a_0 \end{pmatrix} \quad \text{and} \quad T_B = \begin{pmatrix} b_0 & b_1 & & & \\ b_1 & b_0 & b_1 & & \\ & \ddots & \ddots & \ddots & \\ & & b_1 & b_0 & b_1 \\ & & & b_1 & b_0 \end{pmatrix}.$$

System matrix A_2 differs to T in the first and last block because $\tilde{A}_{2,0} \neq \tilde{A}_{2,1} = T_A$, and $\tilde{A}_{2,1}$ is not a Toeplitz matrix like T_A because $\tilde{D}_1$ is not a

Toeplitz matrix, see Eq. (3.31). In article [80], the eigenvalues of a tridiagonal block Toeplitz matrix T as in Eq. (4.20) are derived as

$$\mu_{j,m} = a_0 + 2a_1 \cos\left(\frac{j\pi}{N_j+1}\right) + 2b_0 \cos\left(\frac{m\pi}{N_m+1}\right) + 4b_1 \cos\left(\frac{j\pi}{N_j+1}\right)\cos\left(\frac{m\pi}{N_m+1}\right) \tag{4.21}$$

with $j \in \{1,\dots,N_j\}$ and $m \in \{1,\dots,N_m\}$. Similar to the one-dim. case we are now able to find an estimate of the eigenvalues of system matrix A_2. We set $a_0 = -2p_1 - 2p_2$, $a_1 = p_1$, $b_0 = p_2$ with $p_1 = \frac{\alpha_1}{\Delta x_1^2}$, $p_2 = \frac{\alpha_2}{\Delta x_2^2}$, and we note the eigenvalue approximation of D_2 as

$$\mu_{j,m} = -2p_1\left[1-\cos\left(\frac{j\pi}{N_j+1}\right)\right] - 2p_2\left[1-\cos\left(\frac{m\pi}{N_m+1}\right)\right]. \tag{4.22}$$

We compare the eigenvalues of the previous tridiagonal Toeplitz matrix[8] as in Eq. (4.6) with the eigenvalue estimation as in Eq. (4.22) and so we find an identical structure. Hence, we may interpret Eq. (4.22) as the two-dim. version of Eq. (4.6).

[8] The tridiagonal Toeplitz matrix in Eq. (4.5) approximates almost the one-dimensional diffusion matrix D_1.

We already know that the eigenvalues of the approximated linear one-dim. heat equation (4.4) is found as in Eq. (4.11) and thus we transfer our similarity findings from Eq. (4.6) and (4.22) to the approximated two-dim. heat equation (4.19). Thus, we claim that the eigenvalues of Eq. (4.19) are noted as

$$\mu_{j,m} = -2p_1\left[1-\cos\left([j-1]\frac{\pi}{N_j}\right)\right] - 2p_2\left[1-\cos\left([m-1]\frac{\pi}{N_m}\right)\right] \tag{4.23}$$

with $p_1 = \frac{\alpha_1}{\Delta x_1^2}$ and $p_2 = \frac{\alpha_2}{\Delta x_2^2}$ and for $j \in \{1,\dots,N_j\}$ and $m \in \{1,\dots,N_m\}$. The eigenvalues are sorted with the global index $i(j,m) = j + (m-1)N_j$ as $\mu_{j,m} = \mu_i$, see also Eq. (3.8). In accordance with the eigenvector computation of the one-dim. case, see Eq. (4.10), we formulate the i-th eigenvector as

$$\psi_i := (\psi_{i,1},\dots,\psi_{i,N_c})^\top$$

with $N_c = N_j \cdot N_m$ and the vector elements

$$\psi_{(j,m),(n_j,n_m)} = \cos\left(\frac{(j-1)(2n_j-1)\pi}{2N_j}\right)\cos\left(\frac{(m-1)(2n_m-1)\pi}{2N_m}\right) \tag{4.24}$$

for $n_j \in \{1,\dots,N_j\}$ and $n_m \in \{1,\dots,N_m\}$. We find the superposition of the diffusion as an addition in Eq. (4.23) and as a multiplication in Eq. (4.24). As in the one-dim. case, we state our ideas in a lemma and we prove the correctness of the eigenvalue problem (4.2) with the claimed eigenvalues and eigenvectors. These findings are presented without a proof in article [37] to derive a time-discrete heat conduction model.

Lemma 4.2

The values μ_i in Eq. (4.23), the vectors $\psi_i = (\psi_{i,1},\dots,\psi_{i,N_jN_m})^\top$ with the elements $\psi_{(j,m),(n_j,n_m)} = \psi_{i,n}$ in Eq. (4.24) and index $i(j,m) = j+(m-1)N_j$ as in Eq. (3.8) solve the eigenvalue problem

$$A_2\psi_i = \mu_i\psi_i \tag{4.25}$$

with system matrix A_2 in Eq. (3.37) for $(j,m) \in \{1,\dots,N_j\} \times \{1,\dots,N_m\}$ and $(n_j,n_m) \in \{1,\dots,N_j\} \times \{1,\dots,N_m\}$.

Proof. We carry out this proof analog to the one of Lemma 4.1. We prove that the right-hand side of Eq. (4.25) is identical with its left-hand side expression. However, we need to take the block structure into account here. Similar to the previous proof, we firstly describe the supportive identities and evaluate secondly the eigenvalue expressions. We introduce the same function as in the previous proof

$$f(z,n) := \cos([2n-1]z)\ ,$$

which fulfills the identity

$$2\ f(2z,1)\ f(z,n) = f(z,n+1) + f(z,n-1)$$

as shown in Eq. (4.16). We express the eigenvector elements as

$$\begin{aligned}\psi_{(j,m),(n_j,n_m)} &= f(v,n_j)\ f(w,n_m) \\ &= \cos\left([2n_j-1]v\right)\ \cos\left([2n_m-1]w\right)\end{aligned}$$

with $(n_j,n_m) \in \{1,\dots,N_j\} \times \{1,\dots,N_m\}$ and the eigenvalues as

$$\begin{aligned}\mu_j &= -2p_1[1-f(2v,1)] - 2p_2[1-f(2w,1)] \\ &= -2p_1[1-\cos(2v)] - 2p_2[1-\cos(2w)]\end{aligned}$$

with $v = (j-1)\frac{\pi}{2N_j}$, $w = (m-1)\frac{\pi}{2N_m}$ and $j \in \{1,\dots,N_j\}$, $m \in \{1,\dots,N_m\}$. We shorten the notation of the eigenvector elements as $\psi_{(j,m),(n_j,n_m)} = \psi_{i,n}$ with the global indices

$$\begin{aligned}i(j,m) &= j + (m-1)N_j\ , \\ n(n_j,n_m) &= n_j + (n_m-1)N_j.\end{aligned}$$

Multiplying the i-th eigenvalue with the n-th element of the corresponding eigenvector, we yield

$$\begin{aligned}\mu_i\psi_{i,n} &= \left(-2p_1[1-f(2v,1)] - 2p_2[1-f(2w,1)]\right) f(v,n_j)\ f(w,n_m) \\ &= -2[p_1+p_2]f(v,n_j)\ f(w,n_m) + 2p_1 f(2v,1)f(v,n_j)\ f(w,n_m) \\ &\quad + 2p_2 f(v,n_j)\ f(2w,1)f(w,n_m). \qquad (4.26)\end{aligned}$$

We further specify the products in Eq. (4.26) as

$$\begin{aligned}2f(2v,1)f(v,n_j)\ f(w,n_m) &= [f(v,n_j-1) + f(v,n_j+1)]\ f(w,n_m) \\ &= f(v,n_j-1)\ f(w,n_m) + f(v,n_j+1)f(w,n_m)\end{aligned}$$

and

$$\begin{aligned}2f(v,n_j)\ f(2w,1)f(w,n_m) &= f(v,n_j)[f(w,n_m-1) + f(w,n_m+1)] \\ &= f(v,n_j)f(w,n_m-1) + f(v,n_j)f(w,n_m+1).\end{aligned}$$

For $(n_j,n_m) \in \{2,\dots,N_j-1\} \times \{2,\dots,N_m-1\}$, we note

$$f(v,n_j-1)\ f(w,n_m) = \psi_{i,[n_j-1+(n_m-1)N_j]}, \qquad (4.27a)$$

$$f(v,n_j+1)\ f(w,n_m) = \psi_{i,[n_j+1+(n_m-1)N_j]}, \qquad (4.27b)$$

$$f(v,n_j)\ f(w,n_m-1) = \psi_{i,[n_j+(n_m-2)N_j]}, \qquad (4.27c)$$

$$f(v,n_j)\ f(w,n_m+1) = \psi_{i,[n_j+n_mN_j]}. \qquad (4.27d)$$

We find the remaining expressions for the *initial* values $n_j = 1$, $n_m = 1$ with identity (4.17) as

$$f(v,0)\, f(w,n_m) = f(v,1)\, f(w,n_m) = \psi_{i,[1+(n_m-1)N_j]}, \tag{4.28}$$

$$f(v,n_j)\, f(w,0) = f(v,n_j)\, f(w,1) = \psi_{i,[n_j]} \tag{4.29}$$

and we note with Eq. (4.18) the *terminal* values for $n_j = N_j$, $n_m = N_m$ as

$$f(v,N_j+1)\, f(w,n_m) = f(v,N_j)\, f(w,n_m) = \psi_{i,[n_m N_j]}, \tag{4.30}$$

$$f(v,n_j)\, f(w,N_m+1) = f(v,n_j)\, f(w,N_m) = \psi_{i,[n_j+(N_m-1)N_j]}. \tag{4.31}$$

System matrix A_2 is a block tridiagonal matrix, see Eq. (3.37), and so we compute of the eigenvalue problem with two nested iterations: we iterate over each row block $n_m \in \{1,\dots,N_m\}$ and each row inside the block $n_j \in \{1,\dots,N_j\}$. We remind that each index (j,m) and (n_j,n_m) corresponds to a cell in the finite volume grid, see Fig. 3.5 and Fig. 3.7.

1. Block Row: In the first block row, $n_m = 1$, we multiply i-th or (j,m)-th eigenvector with the blocks $\tilde{A}_{2,0}$ and $\tilde{A}_{2,2}$ as

$$[\tilde{A}_{2,0}, \tilde{A}_{2,2}] \begin{pmatrix} \psi_{i,1} \\ \vdots \\ \psi_{i,N_j} \\ \psi_{i,N_j+1} \\ \vdots \\ \psi_{i,2N_j} \end{pmatrix} = \tilde{A}_{2,0} \begin{pmatrix} \psi_{i,1} \\ \vdots \\ \psi_{i,N_j} \end{pmatrix} + \tilde{A}_{2,2} \begin{pmatrix} \psi_{i,N_j+1} \\ \vdots \\ \psi_{i,2N_j} \end{pmatrix} = \mu_i \begin{pmatrix} \psi_{i,1} \\ \vdots \\ \psi_{i,N_j} \end{pmatrix}$$

with the tridiagonal matrix $\tilde{A}_{2,0} = p_1 \tilde{D}_1 - p_2 I$ and the diagonal matrix $\tilde{A}_{2,2} = p_2 I$. The position of the finite volume cells corresponding to the indices of the eigenvalues and eigenvectors is portrayed in Fig. 4.5. In the first line, $n_j = 1$, the first eigenvector element has the local indices $(n_j,n_m) = (1,1)$ and so we note eigenvalue equation

$$-[p_1+p_2]\psi_{i,1} + p_1\psi_{i,2} + p_2\psi_{i,N_j+1} = \mu_i \psi_{i,1}. \tag{4.32}$$

We evaluate $\mu_i\psi_{i,1}$ as in Eq. (4.26) with the identities (4.27d, 4.28,4.29) and we yield

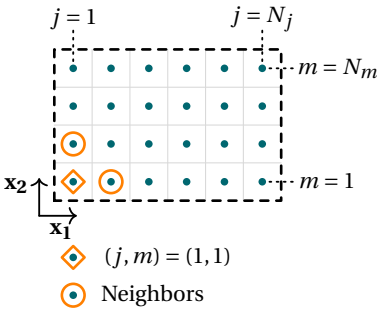
$$\begin{aligned} \mu_i\psi_{i,1} &= -2[p_1+p_2]\psi_{i,1} + p_1[\psi_{i,1}+\psi_{i,2}] + p_2[\psi_{i,1}+\psi_{i,N_j+1}] \\ &= -[p_1+p_2]\psi_{i,1} + p_1\psi_{i,2} + p_2\psi_{i,N_j+1}. \quad \checkmark \end{aligned}$$

We mark the evaluation of $\mu_i\psi_{i,1}$ with ✓ to express the correctness of the eigenvalue equation. In the next rows, $n_j \in \{2,\dots,N_j-1\}$ and $n_m = 1$, we formulate the eigenvalue equation

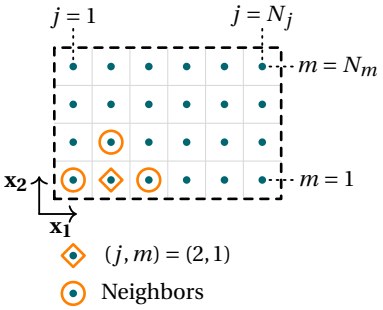
$$p_1\psi_{i,n_j-1} - [2p_1+p_2]\psi_{i,n_j} + p_1\psi_{i,n_j+1} + p_2\psi_{i,N_j+n_j} = \mu_i\psi_{i,n_j}. \tag{4.33}$$

We see that Eq. (4.33) is fulfilled because we calculate its right-hand side with Eq. (4.26) and identities (4.27a, 4.27b, 4.27d,4.29) as

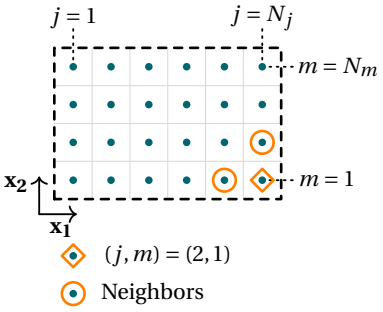
$$\begin{aligned} \mu_i\psi_{i,n_j} &= -2[p_1+p_2]\psi_{i,n_j} + p_1[\psi_{i,n_j-1}+\psi_{i,n_j+1}] + p_2[\psi_{i,n_j}+\psi_{i,N_j+n_j}] \\ &= p_1\psi_{i,n_j-1} - [2p_1+p_2]\psi_{i,n_j} + p_1\psi_{i,n_j+1} + p_2\psi_{i,N_j+n_j}. \quad \checkmark \end{aligned}$$



(a) First line: $n_j = 1$



(b) Central lines: $n_j = 2$



(c) Last line: $n_j = N_j$

Figure 4.5: The position of the finite volume cells corresponds to the eigenvalue equations:
Fig. (a) and Eq.(4.32),
Fig. (b) and Eq. (4.33) for $n_j = 2$,
Fig. (c) and (4.34).
The central point (j,m) correspond to the index of the right-hand side expression $\mu_i\psi_{i,(n_j,n_m)}$.

The eigenvalue problem in the last row of the first block, $n_j = N_j$ and $n_m = 1$, is given as

$$p_1\psi_{i,N_j-1} - (p_1+p_2)\psi_{i,N_j} + p_2\psi_{i,2N_j} = \mu_i\psi_{i,N_j}. \tag{4.34}$$

We state the right-hand side of the eigenvalue equation as

$$\begin{aligned}
\mu_i\psi_{i,N_j} =& -2[p_1+p_2]f(v,N_j)\ f(w,1) + 2p_1 f(2v,1)f(v,N_j)\ f(w,1) \\
&+ 2p_2 f(v,N_j)\ f(2w,1)f(w,1) \\
=& -2[p_1+p_2]f(v,N_j)\ f(w,1) \\
&+ p_1\left[f(v,N_j-1)\ f(w,1) + f(v,N_j+1)f(w,1)\right] \\
&+ p_2\left[f(v,N_j)f(w,1) + f(v,N_j)f(w,2)\right] \\
=& \ p_1 f(v,N_j-1)\ f(w,1) - [p_1+p_2]f(v,N_j)\ f(w,1) + f(v,N_j)f(w,2) \\
&+ p_1[-f(v,N_j) + f(v,N_j+1)]\ f(w,1)
\end{aligned}$$

and we apply the identity (4.30) to yield

$$\begin{aligned}
\mu_i\psi_{i,N_j} &= p_1 f(v,N_j-1)\ f(w,1) - [p_1+p_2]f(v,N_j)\ f(w,1) + f(v,N_j)f(w,2) \\
&= p_1\psi_{i,N_j-1} - (p_1+p_2)\psi_{i,N_j} + p_2\psi_{i,2N_j}. \quad \checkmark
\end{aligned}$$

2. Block Row: In the second block row, $n_m = 2$, we multiply the i-th or (j,m)-th eigenvector with the blocks $\tilde{A}_{2,1}$ and $\tilde{A}_{2,2}$ as

$$[\tilde{A}_{2,2},\tilde{A}_{2,1},\tilde{A}_{2,2}]\begin{pmatrix}\psi_{i,1}\\ \vdots \\ \psi_{i,N_j}\\ \psi_{i,N_j+1}\\ \vdots \\ \psi_{i,2N_j}\\ \psi_{i,2N_j+1}\\ \vdots \\ \psi_{i,3N_j}\end{pmatrix} = \tilde{A}_{2,2}\begin{pmatrix}\psi_{i,1}\\ \vdots \\ \psi_{i,N_j}\end{pmatrix} + \tilde{A}_{2,1}\begin{pmatrix}\psi_{i,N_j+1}\\ \vdots \\ \psi_{i,2N_j}\end{pmatrix} + \tilde{A}_{2,2}\begin{pmatrix}\psi_{i,2N_j+1}\\ \vdots \\ \psi_{i,3N_j}\end{pmatrix} = \mu_i\begin{pmatrix}\psi_{i,N_j+1}\\ \vdots \\ \psi_{i,2N_j}\end{pmatrix}$$

where we have the tridiagonal matrix $\tilde{A}_{2,1} = p_1\tilde{D}_1 - 2p_2 I$ and the diagonal matrix $\tilde{A}_{2,2} = p_2 I$. This procedure is analog to the previous one but here we apply Eq. (4.28) only in the first row of the block. In the first row of the second block row, $n_j = 1$ and $n_m = 2$, the eigenvalue equation is stated as

$$p_2\psi_{i,1} - [p_1+2p_2]\psi_{i,N_j+1} + p_1\psi_{i,N_j+2} + p_2\psi_{i,2N_j+1} = \mu_i\psi_{i,N_j+1}. \tag{4.35}$$

We evaluate $\mu_i\psi_{i,N_j+1}$ analog to the previous eigenvalue computations with Eq. (4.26) and the identities (4.27b, 4.27c, 4.27d) and the initial value identity (4.28). Thus, we yield

$$\begin{aligned}
\mu_i\psi_{i,N_j+1} &= -2[p_1+p_2]\psi_{i,N_j+1} + p_1[\psi_{i,N_j+1}+\psi_{i,N_j+2}] + p_2[\psi_{i,1}+\psi_{i,2N_j+1}] \\
&= p_2\psi_{i,1} - [p_1+2p_2]\psi_{i,N_j+1} + p_1\psi_{i,N_j+2} + p_2\psi_{i,2N_j+1}. \quad \checkmark
\end{aligned}$$

In the next rows, $n_j \in \{2,\dots,N_j-1\}$ and $n_m = 2$, the central point is completely inside the grid and all neighboring points and indices exist, see Fig. 4.6. So, we have the eigenvalue formula with all neighbors as

$$\begin{aligned}
p_2\psi_{i,n_j} + p_1\psi_{i,n_j-1+N_j} - 2[p_1+p_2]\psi_{i,n_j+N_j} + p_1\psi_{i,n_j+1+N_j} \\
+ p_2\psi_{i,n_j+2N_j} = \mu_i\psi_{i,n_j+N_j}.
\end{aligned}$$

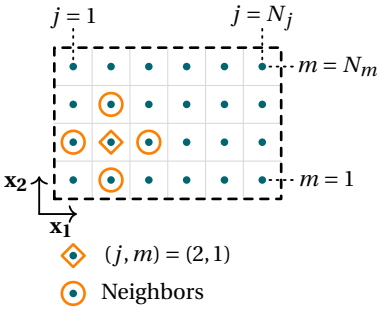


Figure 4.6: The finite volume cell at $(j,m) = (2,2)$ corresponds to the eigenvalue equation at $(n_j,n_m) = (2,2)$.

We evaluate expression $\mu_i \psi_{i,n_j}$ with Eq. (4.26) and consider all identities (4.27) to find

$$\begin{aligned}\mu_i \psi_{i,n_j+N_j} =& -2[p_1+p_2]\psi_{i,n_j+N_j} + p_1[\psi_{i,n_j-1+N_j} + \psi_{i,n_j+1+N_j}] \\ &+ p_2[\psi_{i,n_j} + \psi_{i,n_j+2N_j}] \\ =& \, p_2\psi_{i,n_j} + p_1\psi_{i,n_j-1+N_j} - 2[p_1+p_2]\psi_{i,n_j+N_j} \\ &+ p_1\psi_{i,n_j+1+N_j} + p_2\psi_{i,n_j+2N_j}. \quad \checkmark\end{aligned}$$

In the last row of the second block row, $n_j = N_j$ and $n_m = 2$, we note the eigenvalue equation

$$p_2\psi_{i,N_j} + p_1\psi_{i,2N_j-1} - [p_1+2p_2]\psi_{i,2N_j} + p_2\psi_{i,3N_j} = \mu_i\psi_{i,2N_j}.$$

We calculate the term $\mu_i\psi_{i,N_j}$ similar to Eq. (4.34): we apply the identities (4.27a, 4.27c, 4.27d) to find

$$\begin{aligned}\mu_i \psi_{i,N_j} =& -2[p_1+p_2]\psi_{i,2N_j} + p_1[\psi_{i,2N_j-1} + \psi_{i,2N_j+1}] \\ &+ p_2[\psi_{i,N_j} + \psi_{i,3N_j}]\end{aligned}$$

and we replace $\psi_{i,2N_j+1}$ by $\psi_{i,2N_j}$ with the terminal value expression

$$f(v, N_j+1) = f(v, N_j)$$

in Eq. (4.30) to obtain the desired eigenvalue equation

$$\mu_i\psi_{i,N_j} = \, p_2\psi_{i,N_j} + p_1\psi_{i,2N_j-1} - [p_1+2p_2]\psi_{i,2N_j} + p_2\psi_{i,3N_j}. \quad \checkmark$$

The solution of the eigenvalue problem for all further inner block rows $n_m \in \{3,\dots,N_m-1\}$ is analog to the described way. So, we continue with the last block row.

N_m-th Block Row: In the last block row, $n_m = N_m$, we have a similar situation as in the first block row as

$$[\tilde{A}_{2,2}, \tilde{A}_{2,0}] \begin{pmatrix} \psi_{i,(N_m-2)N_j+1} \\ \vdots \\ \psi_{i,(N_m-1)N_j} \\ \psi_{i,(N_m-1)N_j+1} \\ \vdots \\ \psi_{i,N_jN_m} \end{pmatrix} = \tilde{A}_{2,0} \begin{pmatrix} \psi_{i,(N_m-2)N_j+1} \\ \vdots \\ \psi_{i,(N_m-1)N_j} \end{pmatrix} + \tilde{A}_{2,2} \begin{pmatrix} \psi_{i,(N_m-1)N_j+1} \\ \vdots \\ \psi_{i,N_jN_m} \end{pmatrix} = \mu_i \begin{pmatrix} \psi_{i,(N_m-2)N_j+1} \\ \vdots \\ \psi_{i,(N_m-1)N_j} \end{pmatrix}.$$

In contrast to the previous block rows, we have to apply the identity of the terminal value (4.31) in each line the last block.

The first eigenvalue equation of the last block, $n_j = 1$ and $n_m = N_m$, is noted as

$$p_2\psi_{i,(N_m-2)N_j+1} - [p_1+p_2]\psi_{i,(N_m-1)N_j+1} + p_1\psi_{i,(N_m-1)N_j+2} = \mu_i\psi_{i,(N_m-1)N_j+1}.$$

We calculate the eigenvalue expression $\mu_i\psi_{i,(N_m-1)N_j+1}$ as

$$\begin{aligned}\mu_i\psi_{i,(N_m-1)N_j+1} =& \left(-2p_1[1-f(2v,1)] - 2p_2[1-f(2w,1)]\right) f(v,1)\, f(w,N_m) \\ =& -2[p_1+p_2] f(v,1)\, f(w,N_m) + 2p_1 f(2v,1) f(v,1)\, f(w,N_m) \\ &+ 2p_2 f(v,1)\, f(2w,1) f(w,N_m) \qquad (4.36)\end{aligned}$$

and we yield

$$f(v,1)\ f(2w,1) f(w,N_m) = f(v,1)\ [f(w,N_m-1) + f(w,N_m+1)].$$

Here, we apply the identity (4.31) and so we find

$$\begin{aligned}\mu_i \psi_{i,n} &= p_2 f(v,1) f(w,N_m-1) - [p_1+p_2] f(v,1)\ f(w,N_m) \\ &\quad + p_1 f(v,2)\ f(w,N_m) \\ &= p_2 \psi_{i,(N_m-2)N_j+1} - [p_1+p_2]\psi_{i,(N_m-1)N_j+1} + p_1 \psi_{i,(N_m-1)N_j+2}. \quad \checkmark\end{aligned}$$

We note the eigenvalue equation of the intermediate lines in the last block, $n_j \in \{2,\dots,N_j-1\}$ and $n_m = N_m$, as

$$\begin{aligned}p_2\psi_{i,(N_m-2)N_j+n_j} + p_1\psi_{i,(N_m-1)N_j+n_j-1} - [2p_1+p_2]\psi_{i,(N_m-1)N_j+n_j} \\ + p_1\psi_{i,(N_m-1)N_j+n_j+1} = \mu_i \psi_{i,(N_m-1)N_j+n_j}.\end{aligned}$$

and we evaluate $\mu_i\psi_{i,(N_m-1)N_j+n_j}$ analog to the previous eigenvalue computations with Eq. (4.26) and the identities (4.27a, 4.27b, 4.27c) and (4.31). Thus, we yield

$$\begin{aligned}\mu_i \psi_{i,(N_m-1)N_j+n_j} &= -2[p_1+p_2]\psi_{i,(N_m-1)N_j+n_j} \\ &\quad + p_1[\psi_{i,(N_m-1)N_j+n_j-1} + \psi_{i,(N_m-1)N_j+n_j+1}] \\ &\quad + p_2[\psi_{i,(N_m-2)N_j+n_j} + \psi_{i,(N_m-1)N_j+n_j}] \\ &= p_2\psi_{i,(N_m-2)N_j+n_j} + p_1\psi_{i,(N_m-1)N_j+n_j-1} \\ &\quad - [2p_1+p_2]\psi_{i,(N_m-1)N_j+n_j} + p_1\psi_{i,(N_m-1)N_j+n_j+1}. \quad \checkmark\end{aligned}$$

In the very last row, $n_j = N_j$ and $n_m = N_m$, we formulate the eigenvalue equation

$$p_2\psi_{i,(N_m-1)N_j} + p_1\psi_{i,N_jN_m-1} - [p_1+p_2]\psi_{i,N_jN_m} = \mu_i\psi_{i,N_jN_m}.$$

The finite volume cell at index (N_j, N_m) is adjacent to two boundary sides, see Fig. 4.7. Hence, we have to apply the identities of the terminal value expressions (4.30,4.31) and (4.27a, 4.27c) to evaluate $\mu_i\psi_{i,N_jN_m}$ as

$$\begin{aligned}\mu_i\psi_{i,N_jN_m} &= -2[p_1+p_2]\psi_{i,N_jN_m} \\ &\quad + p_1[\psi_{i,N_jN_m-1} + \psi_{i,N_jN_m+1}] \\ &\quad + p_2[\psi_{i,(N_m-2)N_j} + \psi_{i,(N_m+1)N_j}] \\ &= p_2\psi_{i,(N_m-1)N_j} + p_1\psi_{i,N_jN_m-1} - [p_1+p_2]\psi_{i,N_jN_m}. \quad \checkmark\end{aligned}$$

□

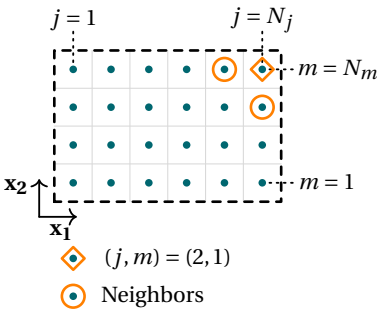


Figure 4.7: The finite volume cell at $(j,m) = (N_j, N_m)$ corresponds to the eigenvalue equation at $(n_j, n_m) = (N_j, N_m)$.

Three-dimensional Heat Conduction

The linear heat equation in three dimensions is noted as

$$\frac{d}{dt}\Theta(t) = A_3\Theta(t)$$

with the system matrix A_3 as in Eq. (3.38). Due to the fact that A_3 is a block tridiagonal matrix like A_2, we can apply the same ideas and techniques from the previous proof. We suggest to compute the eigenvalues of A_3 as

$$\begin{aligned}\mu_{j,m,k} = &-2p_1\left[1-\cos\left([j-1]\frac{\pi}{N_j}\right)\right]\\ &-2p_2\left[1-\cos\left([m-1]\frac{\pi}{N_m}\right)\right]\\ &-2p_3\left[1-\cos\left([k-1]\frac{\pi}{N_k}\right)\right]\end{aligned} \tag{4.37}$$

with coefficients $p_1 = \frac{\alpha_1}{\Delta x_1^2}$, $p_2 = \frac{\alpha_2}{\Delta x_2^2}$, $p_3 = \frac{\alpha_3}{\Delta x_3^2}$ and for

$$(j,m,k) \in \{1,\dots,N_j\} \times \{1,\dots,N_m\} \times \{1,\dots,N_k\}.$$

The eigenvalues are sorted with the global index $i(j,m,k)$, see Eq. (3.7), as $\mu_{j,m,k} = \mu_i$. We define the function

$$f(z,n) := \cos([2n-1]z)$$

and formulate the corresponding i-th eigenvector as $\psi_i := (\psi_{i,1},\dots,\psi_{i,N_c})^\top$, $N_c = N_j N_m N_k$, with its elements as

$$\psi_{(j,m,k),(n_j,n_m,n_k)} = f\left(\frac{j-1}{2N_j}\pi, n_j\right) f\left(\frac{m-1}{2N_m}\pi, n_m\right) f\left(\frac{k-1}{2N_k}\pi, n_k\right) \tag{4.38}$$

for $(n_j,n_m,n_k) \in \{1,\dots,N_j\} \times \{1,\dots,N_m\} \times \{1,\dots,N_k\}$. The correctness of eigenvalues μ_i and eigenvectors ψ_i can be checked via the evaluation of the eigenvalue problem

$$A_3\psi_i = \mu_i\psi_i$$

in an analog way to the previous proof of lemma 4.2.

Summarizing the findings of this section, we are now able to compute the eigenvalues and eigenvectors of A_{N_d}. This fact helps us in the next sections to gain a deeper understanding of the numerical behavior and to construct a closed-form solution of the approximated heat equation in Section 4.3.

4.2 Matrix Properties and Stiffness

In this section, we discuss basic properties of system matrix A_{N_d} and its related linear heat conduction problem in multiple spatial directions as formulated in Definition 3.2. First of all, we derive a matrix transformation of A_{N_d} to the diagonal matrix $\tilde{A}_{N_d}$. For this purpose, we check the symmetry $A_{N_d} = A_{N_d}^\top$ and the orthogonality of the eigenvectors of A_{N_d}. Afterward, we discuss and exemplify the numerical accuracy and stiffness of the approximated linear heat conduction problem.

Matrix Symmetry and Transformation

In the one-dim. case, we find the symmetry of matrix A_1 as

$$A_1^\top = \begin{pmatrix} -a_{1,2} & a_{1,2} & & & \\ a_{1,2} & a_{1,1} & a_{1,2} & & \\ & \ddots & \ddots & \ddots & \\ & & a_{1,2} & a_{1,1} & a_{1,2} \\ & & & a_{1,2} & -a_{1,2} \end{pmatrix} = A_1$$

because the upper and lower sub-diagonals coincide. The system matrix of the two-dim. case is a block matrix and so we need to apply the transpose on each matrix block as

$$A_2^\top = \begin{pmatrix} \tilde{A}_{2,0}^\top & \tilde{A}_{2,2}^\top & & & \\ \tilde{A}_{2,2}^\top & \tilde{A}_{2,1}^\top & \tilde{A}_{2,2}^\top & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{A}_{2,2}^\top & \tilde{A}_{2,1}^\top & \tilde{A}_{2,2}^\top \\ & & & \tilde{A}_{2,2}^\top & \tilde{A}_{2,0}^\top \end{pmatrix}$$

with the matrix blocks

$$\begin{aligned} \tilde{A}_{2,0}^\top &= \left(A_1 - p_2 I\right)^\top = \left(A_1^\top - p_2 I^\top\right) = A_1 - p_2 I = \tilde{A}_{2,0}, \\ \tilde{A}_{2,1}^\top &= \left(A_1 - 2p_2 I\right)^\top = \left(A_1^\top - 2p_2 I^\top\right) = A_1 - 2p_2 I = \tilde{A}_{2,1}, \\ \tilde{A}_{2,2}^\top &= p_2 I^\top = \tilde{A}_{2,2}. \end{aligned}$$

Accordingly, we yield $A_2^\top = A_2$. We find the symmetry of A_3 analog to the two-dim. case as

$$A_3^\top = \begin{pmatrix} \tilde{A}_{3,0}^\top & \tilde{A}_{3,2}^\top & & & \\ \tilde{A}_{3,2}^\top & \tilde{A}_{3,1}^\top & \tilde{A}_{3,2}^\top & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{A}_{3,2}^\top & \tilde{A}_{3,1}^\top & \tilde{A}_{3,2}^\top \\ & & & \tilde{A}_{3,2}^\top & \tilde{A}_{3,0}^\top \end{pmatrix} = A_3$$

because all block matrices are symmetric as

$$\begin{aligned} \tilde{A}_{3,0}^\top &= \left(A_2 - p_3 I\right)^\top = \left(A_2^\top - p_3 I^\top\right) = A_2 - p_3 I = \tilde{A}_{3,0}, \\ \tilde{A}_{3,1}^\top &= \left(A_2 - 2p_3 I\right)^\top = \left(A_2^\top - 2p_3 I^\top\right) = A_2 - 2p_3 I = \tilde{A}_{3,1}, \\ \tilde{A}_{3,2}^\top &= p_2 I^\top = \tilde{A}_{3,2}. \end{aligned}$$

Now, we have the matrix symmetry at hand and so we show the orthogonality of the eigenvectors in the subsequent paragraphs. We define the finite dimensional scalar product for real-valued vectors $\langle \cdot, \cdot \rangle : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ as

$$\langle v, w \rangle := \sum_{i=1}^{N} v_i w_i = v^\top w$$

for $v, w \in \mathbb{R}^N$. If the vectors v and w are not zero vectors and their scalar product is zero as $\langle v, w \rangle = 0$ then v and w are orthogonal.

A matrix $M \in \mathbb{R}^{N \times N}$ is called self-adjoint if for all $v, w \in \mathbb{R}^N$ the identity

$$\langle Mv, w \rangle = \langle v, Mw \rangle$$

holds. We know that a real-valued symmetric matrix $M \in \mathbb{R}^{N\times N}$ is self-adjoint because we find

$$\langle Mv, w\rangle = \langle v, Mw\rangle = (Mv)^\top w = v^\top M^\top w = v^\top Mw = \langle v, Mw\rangle$$

and this concept holds in particular for A_{N_d} as $\langle A_{N_d} v, w\rangle = \langle v, A_{N_d} w\rangle$ for any $v, w \in \mathbb{R}^{N_c}$ with the number of cells N_c. In the next step, we assume two different eigenvalues $\mu_{i_1} \neq \mu_{i_2}$ and eigenvectors $\psi_{i_1} \neq \psi_{i_2}$ for indices $i_1 \neq i_2 \in \{1,\ldots,N_c\}$, which fulfill the eigenvalue equations

$$A_{N_d}\psi_{i_1} = \mu_{i_1}\psi_{i_1} \text{ and } A_{N_d}\psi_{i_2} = \mu_{i_2}\psi_{i_2}\ .$$

In consequence, we find with the identities

$$\langle A_{N_d}\psi_{i_1}, \psi_{i_2}\rangle = \langle \psi_{i_1}, A_{N_d}\psi_{i_2}\rangle$$

and

$$\begin{aligned}\langle A_{N_d}\psi_{i_1}, \psi_{i_2}\rangle &= \langle \mu_{i_1}\psi_{i_1}, \psi_{i_2}\rangle = \mu_{i_1}\langle \psi_{i_1}, \psi_{i_2}\rangle,\\ \langle \psi_{i_1}, A_{N_d}\psi_{i_2}\rangle &= \langle \psi_{i_1}, \mu_{i_2}\psi_{i_2}\rangle = \mu_{i_2}\langle \psi_{i_1}, \psi_{i_2}\rangle\end{aligned}$$

that the scalar product $\langle \psi_{i_1}, \psi_{i_2}\rangle = 0$ because $\mu_{i_1} \neq \mu_{i_2}$. This fact means that all eigenvectors ψ_i are orthogonal and they are a complete basis of $\mathbb{R}^{N_c}$. If the eigenvectors are orthogonal and their vector norm $\|v\| := \sqrt{\langle v, v\rangle}$ is one, then we call them orthonormal. They are computed as

$$\overline{\psi}_i := \frac{\psi_i}{\|\psi_i\|}$$

for $i \in \{1,\ldots,N_c\}$. Moreover, the orthonormal eigenvectors form an *orthogonal* matrix[9] $\overline{V} := [\overline{\psi}_1,\ldots,\overline{\psi}_{N_c}]$ because

$$\overline{V}^\top \overline{V} = I \Leftrightarrow \overline{V}^\top = \overline{V}^{-1}.$$

We apply the identity $\langle \psi_{i_1}, A_{N_d}\psi_{i_2}\rangle = \mu_{i_2}\langle \overline{\psi}_{i_1}, \psi_{i_2}\rangle$ for each possible $i_1, i_2 \in \{1,\ldots,N_c\}$ to yield the transformation

$$\overline{V}^\top A_{N_d}\overline{V} = \tilde{A}_{N_d}\overline{V}^\top\overline{V} = \tilde{A}_{N_d}\ , \tag{4.39}$$

which is equivalent to the identity

$$A_{N_d} = \overline{V}\ \tilde{A}_{N_d}\ \overline{V}^\top \tag{4.40}$$

with matrix $\tilde{A}_{N_d}$ consisting of the eigenvalues as

$$\tilde{A}_{N_d} := \begin{pmatrix}\mu_1 & & & \\ & \mu_2 & & \\ & & \ddots & \\ & & & \mu_{N_c}\end{pmatrix}.$$

We remark that the symmetry of A_{N_d} and the resulting orthogonality of $\overline{V}$ are key factors to compute the transformation (4.40) because otherwise we had to use the inverse $\overline{V}^{-1}$ and its computation might be costly.

[9] As the definitions of orthogonal vectors and matrices differ, we remark that an orthogonal matrix must have unit vectors as rows and columns. In the context of complex-valued computations, we find the term *unitary* matrix.

One of the main goals of transformation (4.39) is the evaluation of $\exp(A_{N_d} t)$ as part of the solution of the linear heat equation

$$\Theta(t) = \exp(A_{N_d} t)\Theta(0).$$

We calculate this matrix exponential as

$$\begin{aligned}
\exp(A_{N_d} t) &= \sum_{n=0}^{\infty} \frac{1}{n} (A_{N_d} t)^n = \sum_{n=0}^{\infty} \frac{1}{n} \left(\overline{V}\, A_{N_d} t\, \overline{V}^{-1} \right)^n \\
&= \sum_{n=0}^{\infty} \overline{V}\, \frac{1}{n} \left(A_{N_d} t \right)^n \overline{V}^{-1} \\
&= \overline{V}\, \exp(\tilde{A}_{N_d} t)\, \overline{V}^{-1} = \overline{V}\, \exp(\tilde{A}_{N_d} t)\, \overline{V}^{\top} \\
&= \overline{V}\, \operatorname{diag}(\exp(\mu_1 t), \ldots, \exp(\mu_{N_c} t))\, \overline{V}^{\top}. \qquad (4.41)
\end{aligned}$$

In Section 4.1, we computed the eigenvalues of A_{N_d} and we find the first eigenvalue for each geometry to be zero: $\mu_1 = 0$, see Eq. (4.11,4.23,4.37). This affects the finding of the inverse of A_{N_d}, the numerical accuracy and the stiffness property as we discuss next. We calculate the determinant of A_{N_d} as

$$\det\left(A_{N_d}\right) = \det\left(\overline{V} \tilde{A}_{N_d} \overline{V}^{\top}\right) = \det\left(\overline{V}\right) \det\left(\tilde{A}_{N_d}\right) \det\left(\overline{V}^{\top}\right) = \det\left(\tilde{A}_{N_d}\right)$$

because $\overline{V}$ is unitary with $\det(\overline{V}^{\top}\overline{V}) = \det(I) = 1$. Thus, we find the determinant as the product of eigenvalues

$$\det\left(\tilde{A}_{N_d}\right) = \prod_{i=1}^{N_c} \mu_i = 0$$

because $\mu_1 = 0$ for all considered geometries.[10] This issue implies that we cannot compute the inverse of A_{N_d} and $\tilde{A}_{N_d}$ because the inverse of a square matrix $M \in \mathbb{R}^{N \times N}$ is found with the adjugate $\operatorname{adj}(\cdot)$ as

$$M^{-1} = \frac{\operatorname{adj}(M)}{\det(M)}.$$

[10] A matrix with a zero eigenvalue is also called singular matrix.

The fact that we are not able to compute the inverse of A_{N_d} leads to further implications. We consider a linear heat conduction problem with constant non-zero Neumann boundary conditions as

$$\frac{d}{dt}\Theta(t) = A_{N_d}\Theta(t) + \phi$$

with a constant heat flux vector $\phi \in \mathbb{R}^{N_c}$. If we wish to find a steady-state temperature distribution Θ_{st} with

$$\frac{d}{dt}\Theta(t) = 0 = A_{N_d}\Theta_{st} + \phi$$

for $t \to \infty$, then we are not able compute Θ_{st} via

$$\Theta_{st} = -A_{N_d}^{-1}\phi$$

because the inverse matrix $A_{N_d}^{-1}$ does not exist. Instead, we have to solve an optimization problem

$$\Theta_{st}^{*} = \arg\min_{\Theta_{st}} \| A_{N_d}\Theta_{st} + \phi \|$$

to yield an approximation of the steady-state temperature Θ_{st}^{*}.

Numerical Accuracy and Stiffness

The right-hand side of the linear heat equation (3.34) is computed several times during a simulation run and each execution causes small numerical errors. Thus, we are interested how precisely our approach works. A well established statement is that a computation is called *well-posed* if small variations of the input data lead to small variations of the resulting data, see [81, page 37]. In the context of matrix vector multiplications as

$$A\,w = b$$

with $A \in \mathbb{R}^{N\times N}$, $w,\ b \in \mathbb{R}^N$ and $N \in \mathbb{N}$, we call the matrix A *well-conditioned* if its related matrix vector multiplication is well-posed. Otherwise, we call A *ill-conditioned*. We begin with a general overview and apply the findings of this analysis afterwards on the linear heat equation.

We consider a mapping $f : \mathbb{R}^N \to \mathbb{R}^N$, $N \in \mathbb{N}$, the true input data $w \in \mathbb{R}^N$ and the disturbed input data $\tilde{w} := w + \Delta w$ with small variations $\Delta w \in \mathbb{R}^N$. The relative error of the function evaluation f has to be smaller than the error of the input data as

$$e_{rel} = \frac{\|f(\tilde{w}) - f(w)\|}{\|f(w)\|} \leq \kappa\, \frac{\|\tilde{w} - w\|}{\|w\|}$$

with coefficient $\kappa > 0$. In case of a linear mapping $f(w) := A\,w$ with $A \in \mathbb{R}^{N\times N}$ we yield

$$e_{rel} = \frac{\|A(\tilde{w} - w)\|}{\|Aw\|} = \frac{\|A\Delta w\|}{\|Aw\|} \leq \kappa \frac{\|\Delta w\|}{\|w\|}. \tag{4.42}$$

If we consider a disturbance $\delta > 0$ only at the i-th position as $\Delta w = \delta e_i$ with a standard unit vector $e_i = (0,\dots,0,1,0,\dots,0)^\top$, $i \in \{1,\dots,N\}$, then we find

$$A\Delta w = Ae_i\delta = \delta \begin{pmatrix} a_{1,i} \\ \vdots \\ a_{N,i} \end{pmatrix}$$

and we note Eq. (4.42) as

$$e_{rel} = \frac{\|A\Delta w\|}{\|Aw\|} = \frac{|\delta|\|Ae_i\|}{\|Aw\|} = \frac{|\delta|\sqrt{\sum_{n=1}^{N} a_{n,i}^2}}{\|Aw\|} \leq \kappa\, \frac{|\delta|}{\|w\|}. \tag{4.43}$$

We see in Eq. (4.43) that the error depends on the position $i \in \{1,\dots,N\}$ and if we consider all possible positions then the computation of the error would be computationally costly. Therefore, we may approximate it via the right-hand side of Eq. (4.43). The coefficient κ is called condition number and in case of a symmetric matrix A with full rank, we obtain it as

$$\kappa := \|A\|\|A^{-1}\| = \frac{\max\limits_{\|w\|=1} \|Aw\|}{\min\limits_{\|w\|=1} \|Aw\|} = \frac{|\mu_{max}(A)|}{|\mu_{min}(A)|} \tag{4.44}$$

in which $|\mu_{max}(A)| = \sqrt{\mu_{max}(A^\top A)}$ and $|\mu_{min}(A)| = \sqrt{\mu_{min}(A^\top A)}$ denote the absolute maximum and minimum eigenvalue of A.

When we apply these concepts on the right-hand side of the linear heat equation $A_{N_d}\Theta(t)$ for any time $t \in [0, T_{final})$, then we see the main issue that the inverse $A_{N_d}^{-1}$ and consequently κ do not exist because

$$\mu_1 = 0 = |\mu_{min}(A_{N_d})|.$$

This means that we cannot estimate the relative error because the upper bound, the right-hand side of Eqs. (4.42, 4.43) may grow up to infinity. Hence, we call matrix A_{N_d} *ill-conditioned*.

We find this issue in another property: the *stiffness* of the differential equation. A system of differential equations is called stiff if the fastest component of its solution is significantly faster than the slowest component. In case of linear systems[11], e.g. $\frac{d}{dt}z(t) = Az(t)$ with $A \in \mathbb{R}^{N\times N}$ the differential equation is called stiff, if

$$|\mu_{max}(A)| \gg |\mu_{min}(A)|$$

and the stiffness ratio is noted as $\frac{|\mu_{max}(A)|}{|\mu_{min}(A)|}$ as in Eq. (4.44). We see that the linear heat equation is stiff because $|\mu_{max}(A)| = |\mu_{N_c}| \gg |\mu_{min}(A)| = 0$. This stiffness property affects the application of the numerical integration methods, because common (non-stiff) numerical solvers, like the explicit Euler method and the explicit Runge-Kutta method may work poorly and we need stiffness-aware, implicit, numerical solvers to handle this issue. We discuss this situation in Chapter 5.

[11] This stiffness concept also applies to nonlinear differential equations and may be studied via linearization of the nonlinear equations.

Example: Numerical Error of One-dimensional Heat Conduction

We exemplify the relative error e_{rel} as in Eq. (4.43) for the one-dim. heat equation

$$\frac{d}{dt}\Theta(t) = A_1\,\Theta(t)$$

with $A_1 = p_1\,\tilde{D}_1$ and $p_1 = 1$, see Eq. (3.31). We demonstrate the error for an increasing number of cells $N_c \in \{3,4,\ldots,100\}$ and we assume the normalized temperature vector

$$\overline{\Theta}_{N_c} = \frac{1}{\|\Theta_{N_c}\|}\Theta_{N_c} \quad \text{with} \quad \Theta_{N_c} = \begin{pmatrix} 1 \\ 1+\frac{1}{N_c-1} \\ 1+\frac{2}{N_c-1} \\ \vdots \\ 1+\frac{N_c-2}{N_c-1} \\ 2 \end{pmatrix}.$$

We only calculate the relative errors, which relate to the first and second column

$$e_{rel,1} = |\delta|\frac{\|A_1\,e_1\|}{\|A_1\,\overline{\Theta}_{N_c}\|} \quad \text{and} \quad e_{rel,2} = |\delta|\frac{\|A_1\,e_2\|}{\|A_1\,\overline{\Theta}_{N_c}\|}$$

with standard unit vectors

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{and} \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

and have the norms

$$\|A_1\,e_1\| = = \sqrt{(-1)^2+1^2} = \sqrt{2},$$
$$\|A_1\,e_2\| = = \sqrt{1^2+(-2)^2+1^2} = \sqrt{6}.$$

N_c	3	5	10	20	50	100
$e_{rel,1}$	5.4	13.8	43.6	130.0	529.6	1512.8
$e_{rel,2}$	9.3	24.9	75.6	225.2	917.4	2620.2

Table 4.1: Relative numerical error of the linear heat equation.

All remaining relative errors are equal to the previous ones because the error of the last column is analog to the first one and the error of the second column is the same for all central columns as

$$\|A_1 e_i\| = \begin{cases} \|A_1 e_1\| & \text{if } i = N_c, \\ \|A_1 e_2\| & \text{if } i \in \{3, \dots, N_c - 1\}. \end{cases}$$

As $\|A_1\ e_2\| = \sqrt{3}\|A_1\ e_1\|$, we see that $e_{rel,2} = \sqrt{3}\ e_{rel,1}$. We set the disturbance $\delta = 1$ and evaluate the relative error for $N_c = \{3, \dots, 100\}$. The computed relative errors are noted in Table 4.1 and depicted Fig. 4.8. We find a nonlinear rise of the relative error, which means that finer approximations are stronger affected by the ill-conditioned matrix A_{N_d} than coarse ones. As we are interested in a precise approximation to simulate the thermal dynamics exactly, we face the challenge to compute exact temperatures with a fine spatial sampling while avoiding such numerical inaccuracies.

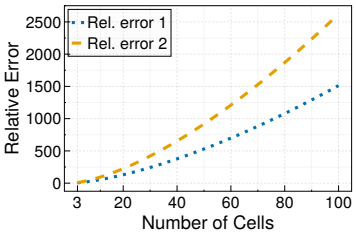


Figure 4.8: The relative error $e_{rel,1}$ and $e_{rel,2}$ in the simulation of the linear heat conduction is increasing by the number of cells $N_c \in \{3, 4, \dots, 100\}$.

4.3 Analytical Solution of the Linear Problem

In this section, we derive the analytical solution of the linear heat equation with non-zero boundary conditions (3.30). We know that an inhomogeneous differential equation

$$\frac{d}{dt} z(t) = Az(t) + f(t) \tag{4.45}$$

with system matrix $A \in \mathbb{R}^{N \times N}$, states $z : [0, T_{final}) \to \mathbb{R}^N$ and additional force $f : [0, T_{final}) \to \mathbb{R}^N$ is solved via "variation of constants" as

$$z(t) = \exp(At)\ z(0) + \int_0^t \exp(A[t-\tau])\ f(\tau) d\tau. \tag{4.46}$$

We transfer this concept to our linear heat conduction problem with temperatures $z(t) = \Theta(t)$, system matrix $A = A_{N_d}$, see Eq. (3.35) and heat flux

$$f(t) = \sum_{l=1}^{N_d} E_l \frac{\phi_l(t)}{\Delta x_l}.$$

In consequence, we evaluate Eq. (4.41) and we obtain the solution

$$\Theta(t) = \overline{V} \exp(\tilde{A}_{N_d} t) \overline{V}^\top \Theta(0) + \overline{V} \int_0^t \exp(\tilde{A}_{N_d}[t-\tau]) \overline{V}^\top \left[\sum_{l=1}^{N_d} E_l \frac{\phi_l(\tau)}{\Delta x_l} \right] d\tau \tag{4.47}$$

with diagonal matrix $\exp(\tilde{A}_{N_d} t) = \mathrm{diag}(\exp(\mu_1 t), \dots, \exp(\mu_{N_c} t))$. Hence, the computation of the eigenvalues and eigenvectors in Section 4.1 enables us here to find an analytical solution (4.47) of the spatially approximated heat equation (3.30). If we start at any time $t_1 \in [0, T_f)$ and we integrate until $t_2 \in (0, T_f]$, $t_1 < t_2$, then we solve the linear heat equation iteratively as

$$\Theta(t_2) = \overline{V}\exp(\tilde{A}_{N_d}[t_2-t_1])\overline{V}^\top\Theta(t_1) + \overline{V}\int_{t_1}^{t_2}\exp(\tilde{A}_{N_d}[t_2-\tau])\overline{V}^\top\left[\sum_{l=1}^{N_d}E_l\frac{\phi_l(\tau)}{\Delta x_l}\right]d\tau \tag{4.48}$$

for arbitrary time steps in $[0,T_f]$. If these time steps are sampled equidistantly with sampling time $\Delta T = t_2 - t_1$ and we assume a constant heat flux between the samplings then we yield at time $t = n\Delta T$ with iteration $n \in \{0,1,\dots,\lfloor\frac{T_f}{\Delta T}\rfloor - 1\}$ the solution

$$\Theta([n+1]\Delta T) = \overline{V}\,\mathrm{e}^{\tilde{A}_{N_d}\Delta T}\overline{V}^\top\Theta(n\Delta T) + \overline{V}\int_0^{\Delta T}\mathrm{e}^{\tilde{A}_{N_d}[\Delta T-\tau]}\,d\tau\overline{V}^\top\left[\sum_{l=1}^{N_d}E_l\frac{\phi_l(n\Delta T)}{\Delta x_l}\right].$$

In some scenarios, it is useful to consider the transformed temperatures

$$\tilde{\Theta}(t) := \overline{V}^{-1}\Theta(t) = \overline{V}^\top\Theta(t),$$

which lead to the differential equation

$$\begin{aligned}\frac{d}{dt}\tilde{\Theta}(t) &= \overline{V}^\top\frac{d}{dt}\Theta(t)\\ &= \overline{V}^\top A_{N_d}\overline{V}\tilde{\Theta}(t) + \overline{V}^\top\left[\sum_{l=1}^{N_d}E_l\frac{\phi_l(t)}{\Delta x_l}\right]\\ &= \tilde{A}_{N_d}\tilde{\Theta}(t) + \overline{V}^\top\left[\sum_{l=1}^{N_d}E_l\frac{\phi_l(t)}{\Delta x_l}\right].\end{aligned} \tag{4.49}$$

We find the solution of differential equation (4.49) via "variation of constants" like above as

$$\tilde{\Theta}(t) = \exp(\tilde{A}_{N_d}t)\tilde{\Theta}(0) + \left[\int_0^t\exp(\tilde{A}_{N_d}[t-\tau])\overline{V}^\top\sum_{l=1}^{N_d}E_l\frac{\phi_l(\tau)}{\Delta x_l}d\tau\right]. \tag{4.50}$$

We discuss the relation between the solution of the original and transformed states in Eq. (4.47) and (4.50) in an example in the end of this section.

Constant Heat Flux

The total heat flux of supplied power and thermal emissions usually varies in time. Though, this variation impedes the task to analyze the impact of boundary conditions on the thermal dynamics inside the object. Hence, we may assume a constant heat flux , e.g. $\phi_l(t) \equiv \phi_l = const.$ In this case, we are able to calculate the integral in Eq. (4.47) and (4.50) in a closed form as

$$\begin{aligned}&\int_0^t\exp(\tilde{A}_{N_d}[t-\tau])\overline{V}^\top\sum_{l=1}^{N_d}E_l\frac{\phi_l(\tau)}{\Delta x_l}d\tau\\ &= \int_0^t\exp(\tilde{A}_{N_d}[t-\tau])d\tau\;\overline{V}^\top\sum_{l=1}^{N_d}E_l\frac{\phi_l}{\Delta x_l}\\ &= M(t)\;\overline{V}^\top\sum_{l=1}^{N_d}E_l\frac{\phi_l}{\Delta x_l}\end{aligned} \tag{4.51}$$

with diagonal matrix

$$M(t) := \mathrm{diag}\left(t, \frac{1}{|\mu_2|}\left[1-\mathrm{e}^{\mu_2 t}\right],\dots,\frac{1}{|\mu_{N_c}|}\left[1-\mathrm{e}^{\mu_{N_c}t}\right]\right).$$

We find the elements of $M(t)$ via simple integration for $\mu_1 = 0$ as

$$\int_0^t \mathrm{e}^{0[t-\tau]}\, d\tau = \int_0^t d\tau = t$$

and

$$\int_0^t \mathrm{e}^{\mu_i[t-\tau]}\, d\tau = \frac{1}{\mu_i}\left[\mathrm{e}^{\mu_i t} - 1\right] = \frac{1}{|\mu_i|}\left[1 - \mathrm{e}^{\mu_i t}\right]$$

for $i \in \{2, \ldots, N_c\}$ because all $\mu_i < 0$. Therefore, we note the solution of Eq. (4.47) and

$$\Theta(t) = \overline{V} \exp(\tilde{A}_{N_d} t) \overline{V}^\top \Theta(0) + \overline{V} M(t) \overline{V}^\top \sum_{l=1}^{N_d} E_l \frac{\phi_l}{\Delta x_l} \tag{4.52}$$

and we find the solution of Eq. (4.50) with the transformed temperatures $\tilde{\Theta}$ as

$$\tilde{\Theta}(t) = \exp(\tilde{A}_{N_d} t)\, \tilde{\Theta}(0) + M(t)\, \overline{V}^\top \sum_{l=1}^{N_d} E_l \frac{\phi_l}{\Delta x_l}. \tag{4.53}$$

Heat Transfer along Boundary Sides

If we assume the thermal emissions ϕ_{em}, as in Definition 2.3, in the integral of Eq. (4.47) and (4.50), then we obtain a (nonlinear) state feedback because the temperature values along the boundary sides determine the emitted heat flux ϕ_{em}. In case of pure linear heat transfer

$$\phi_{l,i}(t) = -h_{l,i}\, \left[\Theta_i(t) - \Theta_{amb,l,i}\right]$$

in each boundary cell $i \in \mathscr{S} \setminus \mathring{\mathscr{S}}$, see Eq. (3.8) we summarize all $\phi_{l,i}$ for each $l \in \{1, \ldots, N_d\}$ and we note the state feedback via thermal emissions as

$$\phi_l(t) = -H_l\, E_l^\top\, \Theta(t) + H_l\, \Theta_{amb,l}. \tag{4.54}$$

Here, the expression $E_l^\top\, \Theta(t)$ filters for boundary cells and the heat transfer coefficients are stored as $H_l = \mathrm{diag}(h_{l,1}, \ldots, h_{l,N_j N_m N_k})$ with

$$\begin{aligned} H_1 \in \mathbb{R}^{2N_m N_k \times 2N_m N_k} \quad &, \quad \Theta_{amb,1} \in \mathbb{R}^{2N_m N_k}, \\ H_2 \in \mathbb{R}^{2N_j N_k \times 2N_j N_k} \quad &, \quad \Theta_{amb,2} \in \mathbb{R}^{2N_j N_k}, \\ H_3 \in \mathbb{R}^{2N_j N_m \times 2N_j N_m} \quad &, \quad \Theta_{amb,3} \in \mathbb{R}^{2N_j N_m}. \end{aligned}$$

We identify ϕ_l in right-hand side of Eq. (4.49) with the thermal emission in Eq. (4.54) and so we obtain the differential equation

$$\begin{aligned} \frac{d}{dt}\tilde{\Theta}(t) &= \tilde{A}_{N_d}\tilde{\Theta}(t) + \overline{V}^\top \left[\sum_{l=1}^{N_d} E_l \frac{\phi_l(\tau)}{\Delta x_l}\right] \\ &= \underbrace{\left[\tilde{A}_{N_d} - \overline{V}^\top \sum_{l=1}^{N_d} \frac{1}{\Delta x_l} E_l\, H_l\, E_l^\top\right]}_{=:A_{TE}} \tilde{\Theta}(t) + \overline{V}^\top \left[\sum_{l=1}^{N_d} \frac{1}{\Delta x_l} E_l\, H_l\, \Theta_{amb,l}\right] \end{aligned}$$

in which the new matrix[12] $A_{TE} = \tilde{A}_{N_d} - \overline{V}^\top \sum_{l=1}^{N_d} \frac{1}{\Delta x_l} E_l H_l E_l^\top$ is in general not a diagonal (or triangular) matrix. Hence, we lose the approach to formulate the eigenvalues and eigenvectors of A_{TE} in a closed form and so we may lose the numerical precision of the solution. However, the eigenvalues and eigenvectors of A_{TE} might be computed numerically. These ideas are limited to the linear heat transfer because in case of nonlinear heat radiation, we have a nonlinear differential equation.

[12] It describes a *natural feedback* through thermal emissions.

Gauss-Legendre Quadrature

The integrals in the Eq. (4.47) and (4.50) are in general difficult to solve manually for an arbitrary (smooth and integrable) heat flux $\phi_l(t)$. Thus, we need to compute the integral numerically. One of the most prominent approaches is the Gauss-Legendre quadrature, which is stated as

$$\int_{-1}^{1} f(s)ds = \sum_{i=1}^{n} w_i f(s_i) + R_n \tag{4.55}$$

for an integrable function $f : \mathbb{R} \to \mathbb{R}$ with weights $w_i > 0$, quadrature nodes $s_i \in [-1,1]$, remainder $R_n \geq 0$, and the order of quadrature $n \in \mathbb{N}_{>0}$, see also the literature [82, p. 40] and [83, p. 887]. The task is to find suitable weights and quadrature nodes such that the remainder $R_n \approx 0$ and

$$\int_{-1}^{1} f(s)ds \approx \sum_{i=1}^{n} w_i f(s_i).$$

The quadrature node s_i is the i-th root of the Legendre polynomial $P_n(x)$, which is calculated either with the recursion formula

$$(n+1)P_{n+1}(s) = (2n+1)s\,P_n(s) - nP_{n-1}(s)$$

or with Rodrigues' formula [13]

$$P_n(s) = \frac{1}{2^n n!}\frac{d^n}{ds^n}\left(s^2-1\right)^n.$$

[13] Named after Benjamin Olinde Rodrigues (*1795, † 1851) [84].

We calculate the weights as

$$w_i = \frac{2}{(1-s_i^2)\left[\frac{d}{ds}P_n(s)|_{s=s_i}\right]^2}$$

and we find the remainder as

$$R_n = \frac{2^{2n+1}\,[n!]^4}{(2n+1)\,[(2n)!]^3} f^{2n}(s)$$

for $s \in (-1,1)$. We are interested in the interval $[t_1, t_2]$ with $0 \leq t_1 < t_2 \leq T_f$ instead of $[-1,1]$ and so we need to change the interval in Eq. (4.55) with $\xi \mapsto \frac{t_2-t_1}{2}s + \frac{t_1+t_2}{2}$ and $\frac{d\xi}{ds} = \frac{t_2-t_1}{2}$ as

$$\begin{aligned}\int_{t_1}^{t_2} f(\xi)d\xi &= \int_{-1}^{1} f\left(\frac{t_2-t_1}{2}s + \frac{t_1+t_2}{2}\right)\frac{t_2-t_1}{2}ds \\ &\approx \frac{t_2-t_1}{2}\sum_{i=1}^{n} w_i f\left(\frac{t_2-t_1}{2}s_i + \frac{t_1+t_2}{2}\right)\end{aligned} \tag{4.56}$$

The numerical integration in the subsequent example is implemented with JULIA library *FastGaussQuadrature.jl*. This library provides methods to compute the weights w_i and quadrature nodes s_i in case of a high order $n \geq 60$ according to the approach in article [85].

Example: Simulation of One-dimensional Heat Conduction

We exemplify our findings with a simplified one-dim. heat conduction example with $N_c = 3$, $\Delta x = 1$, and the coefficients $\lambda = \rho = c = 1$. We only have two boundary sides B_W and B_E which influence only one cell per side and so we have the heat flux $\phi_1(t) = \begin{pmatrix} \phi_{1,1}(t) \\ \phi_{1,2}(t) \end{pmatrix}$. We compute the solution of the transformed problem (4.50) for two scenarios. Firstly, we assume a constant heat flux and solve the integral manually; and secondly, the heat flux varies in time and we solve the integral numerically with Gauss-Legendre quadrature. We note the differential equation of the spatially approximated heat conduction as

$$\begin{aligned} \frac{d}{dt}\Theta(t) &= A_1\Theta(t) + E_1\phi_1(t) \\ &= \begin{pmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{pmatrix}\Theta(t) + \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \phi_{1,1}(t) \\ \phi_{1,2}(t) \end{pmatrix}. \end{aligned} \tag{4.57}$$

We find the eigenvalues $\mu \in \{0, -1, -3\}$ and the original and normalized eigenvectors as

$$V = \begin{pmatrix} 1 & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ 1 & 0 & -1 \\ 1 & -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \quad \text{and} \quad \overline{V} = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{pmatrix}$$

and we transform the original differential equation (4.57) with the eigenvectors and the new states $\tilde{\Theta}$ into

$$\begin{aligned} \frac{d}{dt}\tilde{\Theta}(t) &= \tilde{A}_1\tilde{\Theta}(t) + V^\top E_1\phi_1 \\ &= \begin{pmatrix} 0 & & \\ & -1 & \\ & & -3 \end{pmatrix}\tilde{\Theta}(t) + \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{pmatrix}\begin{pmatrix} \phi_{1,1}(t) \\ \phi_{1,2}(t) \end{pmatrix}. \end{aligned} \tag{4.58}$$

Scenario 1: Constant Heat Flux

In the first scenario, we consider a constant heat flux $\phi_1(t) = \phi_1$. We yield the integral as in Eq. (4.51) with diagonal matrix

$$M(t) = \operatorname{diag}\left(t, 1 - \mathrm{e}^{-t}, \frac{1}{3}\left[1 - \mathrm{e}^{-3t}\right]\right).$$

and we solve differential equation (4.58) with Eq. (4.53) as

$$\tilde{\Theta}_1(t) = \tilde{\Theta}_1(0) + t\,\frac{1}{\sqrt{3}}\left[\phi_1 + \phi_2\right], \tag{4.59a}$$

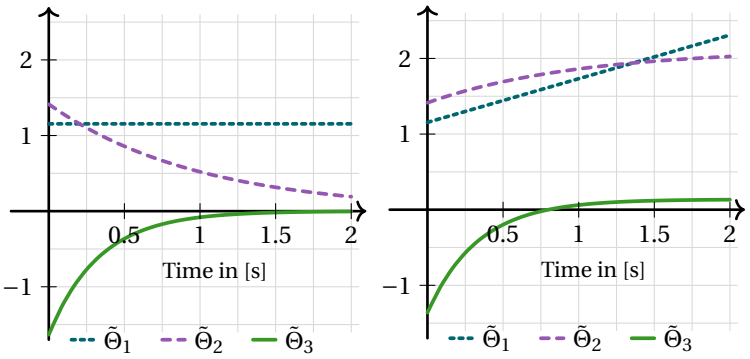
$$\tilde{\Theta}_2(t) = \mathrm{e}^{-t}\tilde{\Theta}_2(0) + \left[1 - \mathrm{e}^{-t}\right]\,\frac{1}{\sqrt{2}}\left[\phi_1 - \phi_2\right], \tag{4.59b}$$

$$\tilde{\Theta}_3(t) = \mathrm{e}^{-3t}\tilde{\Theta}_3(0) + \frac{1}{3\sqrt{6}}\left[1 - \mathrm{e}^{-3t}\right]\left[\phi_1 + \phi_2\right]. \tag{4.59c}$$

We consider the initial temperature values

$$\Theta_0 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} \quad \Rightarrow \quad \tilde{\Theta}_0 = \overline{V}^\top\Theta_0 \begin{pmatrix} \frac{2}{\sqrt{3}} \\ \sqrt{2} \\ -\frac{4}{\sqrt{6}} \end{pmatrix}$$

and we visualize the solution in Fig. 4.9 for two cases:



(a) Insulation (b) Heating and Cooling

Figure 4.9: Simulation of transformed solution (4.59) with insulated boundaries in (a) $\phi_1 = (0,0)^\top$ and constant heat flux in (b) $\phi_1 = (2,-1)^\top$.

1. insulated boundary sides, $\phi_1 = (0,0)^\top$, in Fig. 4.9 (a) and
2. heating on B_W and cooling on B_E, $\phi_1 = (2,-1)^\top$, in Fig. 4.9 (b).

In Fig. 4.9 (a) we find that the states $\tilde{\Theta}_2$ and $\tilde{\Theta}_3$ converge towards zeros and $\tilde{\Theta}_1$ is constant because they are only affected by the unforced term $e^{\mu_i t}\tilde{\Theta}_i$. The states in Fig. 4.9 (b) increase because of the positive heat flux sum $\phi_{1,1} + \phi_{1,2} = 1$. In particular, the first state rises linearly because of linear time in Eq. (4.59a) and the states $\tilde{\Theta}_2$ and $\tilde{\Theta}_3$ look like a charging curve of a capacitor in a RC circuit because of the terms

$$\left[1 - e^{-t}\right] \quad \text{and} \quad \left[1 - e^{-3t}\right]$$

in Eq. (4.59b, 4.59c).

Scenario 2: Time-varying Heat Flux

Now, we wish to solve the original and transformed differential equations (4.57, 4.58) with the time-varying heat flux

$$\phi_{1,1}(t) = 1.2\exp\left(-\left[0.7\left(t - \frac{T_f}{3}\right)\right]^2\right) \quad , \quad \phi_{1,2}(t) = 0. \tag{4.60}$$

Figure 4.10: Time-varying heat flux on boundary B_W as in Eq. (4.60).

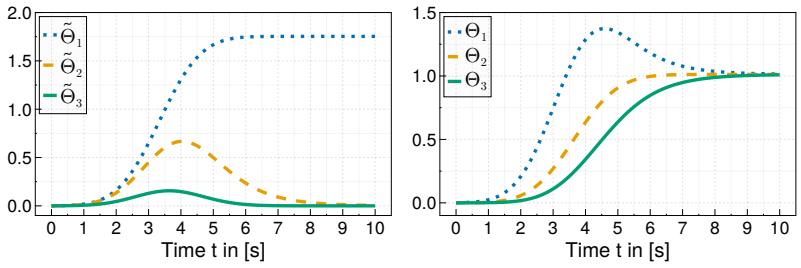
This heat flux means that we supply a power density on B_W while B_E is insulated. We consider this heat flux $\phi_{1,1}(t)$ here because we discuss input signals with such a shape in Chapter 7 for the open-loop control design. We subdivide the time interval $(0, T_f)$ into parts (t_n, t_{n+1}) such that

$$0 = t_0 < t_1 < t_2 < \ldots < t_{N_T} = T_f.$$

and we solve the differential equation of the transformed states iteratively in accordance with Eq. (4.48) as

$$\begin{aligned}\tilde{\Theta}(t_{n+1}) = \exp(\tilde{A}_{N_d}[t_{n+1} - t_n])\,\tilde{\Theta}(t_n) \\ + \int_{t_n}^{t_{n+1}} \exp(\tilde{A}_{N_d}[t_{n+1} - \tau])\overline{V}^\top E_1 \frac{\phi_1(\tau)}{\Delta x_1} d\tau\end{aligned}$$

for $n \in \{0,\ldots,N_T - 1\}$. In particular, we compute the integral numerically with Gauss-Legendre quadrature as in Eq. (4.56). For the simulation, we assume a final time $T_f = 10$ seconds, equidistant time steps $t_n = 0.1n$ and initial values $\Theta(0) = \tilde{\Theta}(0) = (0,0,0)^\top$. Finally, we compute the original temperatures as $\Theta(t_n) = \overline{V}\tilde{\Theta}(t_n)$.



(a) Transformed States

(b) Original States

Figure 4.11: Simulation of the one-dim. heat conduction with time-varying heat flux on B_W and insulation on B_E. Temperature evolution is visualized of the transformed states $\tilde{\Theta}$ in (a) and of the original states Θ in (b).

In Fig. 4.11 (a), the state $\tilde{\Theta}_1$ purely integrates heat flux $\phi_{1,1}$ while $\tilde{\Theta}_2$ and $\tilde{\Theta}_3$ approach a similar shape like $\phi_{1,1}$. In contrast to that, the original states in Fig. 4.11 (b) only show an integrating behavior and all states reach the same final temperature. This is an important fact to steer a temperature distribution from an initial value to a desired final value, see Chapter 7.

In this chapter, we derived the eigenvalues and eigenvectors of the linear heat conduction and we constructed the analytical solution with them. In Section 4.2 and 4.3, we highlight the influence of the first eigenvalue $\mu_1 = 0$, which causes the ill-conditioning of A_{N_d} and the integrating behavior of the heat flux in the solution (4.47). The analytical solution of the linear heat conduction provides a powerful tool to examine the linear thermal dynamics. However, as we face nonlinear problems in our general framework, we describe and utilize numerical solution methods in the next chapter.

5

Numerical Time Integration

In the previous chapters, we approximated the heat conduction problem in space and we discussed the special case of linear thermal dynamics, where we can compute eigenvalues and eigenvectors of the linear system and this enables us to find an analytical solution. However, we had to consider thermally insulated boundaries to yield such a closed-form solution. As we are able to note such analytical solutions only for certain scenarios, in particular for constant material properties and no heat radiation, it is necessary to present a numerical solution in time of the approximated nonlinear heat conduction, see Definition 3.1. For this purpose, we introduce the Euler integration[1] in Section 5.1 and the Runge-Kutta methods[2] in Section 5.2. We apply these methods on the approximated linear heat equation and we discuss the influence of eigenvalues on the quality of the numerical results. In Section 5.3, we evaluate and compare the backward Euler, trapezoidal rule and an implicit Runge-Kutta method for a simple linear heat equation.

[1] According to Leonhard Euler (*1707,†1783) [86].

[2] According to Carl David Tolmé Runge (*1856, †1927) [87] and Martin Wilhelm Kutta (*1867,†1944) [88].

5.1 Euler Integration Methods

First of all, we introduce the Euler methods, which provide the simplest numerical integration approaches. We do not consider the spatially approximated quasilinear heat equation (3.27) explicitly because the presented method may be applied to any linear or nonlinear system and we can transfer the concept directly to the heat equation. We wish to solve numerically the differential equation

$$\frac{d}{dt}z(t) = f(z(t), t) \tag{5.1}$$

with states $z : [0, T_f) \to \mathbb{R}^N$, right-hand side $f : \mathbb{R}^N \times [0, T_f)$ and initial value $z(0) = z_0$. We approximate the differential operator by a first order finite difference approach

$$\frac{d}{dt}z(t) \approx \frac{1}{\Delta T}\left[z(t + \Delta T) - z(t)\right]$$

with sampling time $\Delta T > 0$ and we set the right-hand side as

$$f(z, t) := \omega f(z, t) + (1 - \omega)\, f\left(z(t + \Delta T), t + \Delta T\right)$$

with decision variable $\omega \in [0, 1]$. We obtain the time-discrete equation

$$z(t + \Delta T) = z(t) + \Delta T\, \left[\omega f(z, t) + (1 - \omega) f(z(t + \Delta T), t + \Delta T)\right]$$

and we note the one-step iteration scheme

$$z(t_{n+1}) = z(t_n) + \Delta T \left[\omega f(z_n, t_n) + (1-\omega) f(z_{n+1}, t_{n+1})\right] \tag{5.2}$$

with time steps $t = t_n = n\Delta T$ for $n \in \{0, 1, \ldots, N_T\}$ and states $z_n = z(t_n)$. We visualize the sampling of the right-hand side of Eq. (5.1) in Fig. 5.1. Here, we see that we lose probably necessary information of the continuous differential equation with a coarse sampling. We call iteration (5.2) either forward Euler ($\omega = 1$), backward Euler ($\omega = 0$) or trapezoidal rule ($\omega = \frac{1}{2}$) and we note them as

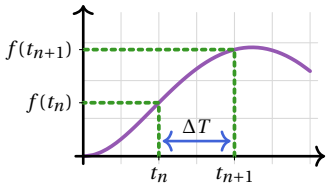


Figure 5.1: Sampling of nonlinear differential equation (5.1) at time steps t_n and $t_{n+1} = t_n + \Delta T$.

$$\omega = 1: \quad z(t_{n+1}) = z(t_n) + \Delta T\, f(z_n, t_n), \tag{5.3a}$$

$$\omega = 0: \quad z(t_{n+1}) = z(t_n) + \Delta T\, f(z_n, t_{n+1}), \tag{5.3b}$$

$$\omega = \frac{1}{2}: \quad z(t_{n+1}) = z(t_n) + \frac{\Delta T}{2}\left[f(z_n, t_n) + f(z_n, t_{n+1})\right]. \tag{5.3c}$$

The forward Euler method ($\omega = 1$) may be applied directly on any differential equation of the form (5.1). Nevertheless for $\omega < 1$ we need to solve the nonlinear equations

$$z(t_{n+1}) - (1-\omega)\,\frac{\Delta T}{2} f(z, n+1) = z(t_n) + \omega\,\frac{\Delta T}{2} f(z_{n+1}, t_{n+1})$$

which might be computationally expensive for a large number of states. The trapezoidal rule is also known in the context of partial differential equations, in particular the heat equation, as Crank-Nicolson method.[3]

[3] The method was developed by John Crank (*1916, †2006) [89], and Phyllis Nicolson (*1917, †1968) [90].

The one-step methods in Eq. (5.3) provide two parameters: sampling time $\Delta T > 0$ and decision variable $\omega \in [0, 1]$. The quality of the numerical results depends strongly on their choice and so we need to check these algorithms. For this purpose, we consider the differential equation

$$f(z, t) = \alpha\, z(t)$$

with $\alpha < 0$ and initial value $z(0) \neq 0$ as test problem. The analytical solution of this system is known as

$$z(t) = \exp(\alpha t)\, z(0) \tag{5.4}$$

and the numerical algorithm is derived from the iteration scheme (5.2) as

$$\begin{aligned} z(t_{n+1}) &= \frac{1 + \omega\, \alpha\Delta T}{1 - (1-\omega)\, a\Delta T}\, z(t_n) \\ &= \left[\frac{1 + \omega\, \alpha\Delta T}{1 - (1-\omega)\, a\Delta T}\right]^n z(0). \end{aligned} \tag{5.5}$$

The transition from an initial state $z(0)$ to a future state $z(t_{n+1})$ is described by the term $\frac{1+\omega\ \alpha\Delta T}{1-(1-\omega)\ \alpha\Delta T}$. We reformulate this term with the new variable $\zeta := -\alpha\Delta T$ as the Euler iterator of the test problem

$$g(\zeta, \omega) := \frac{1 - \omega\zeta}{1 + (1-\omega)\zeta} \tag{5.6}$$

and we note the iteration

$$z(t_{n+1}) = g(\zeta, \omega)\, z(t_n) = g(\zeta, \omega)^n\, z(0). \tag{5.7}$$

The Euler iterator (5.6) is depicted in Fig. 5.2 for $\omega \in \{0, \frac{1}{2}, 1\}$. We know that the analytical solution (5.4) converges towards zero because $a < 0$. Thus,

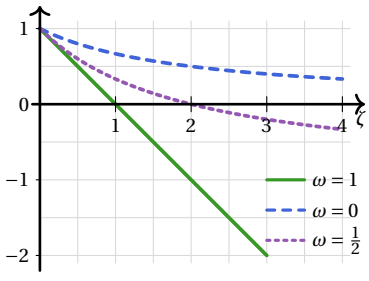


Figure 5.2: Euler iterator $g(\zeta, \omega)$ for forward Euler: $\omega = 1$ (green), backward Euler: $\omega = 0$ (blue) and trapezoidal rule: $\omega = \frac{1}{2}$ (purple).

the numerical solution has to approach zero in the same way. In fact, the iteration algorithm (5.7) converges towards zero if we choose (ζ,ω) such that $g(\zeta,\omega) \in (-1,1)$ because $g(\zeta,\omega)^n \to 0$ for $n \to \infty$. Otherwise, we yield an pure oscillating result for $g(\zeta,\omega) = -1$ because $g(\zeta,\omega)^n = (-1)^n$ or a diverging result for $g(\zeta,\omega) < -1$ because $g(\zeta,\omega)^n \to \pm\infty$. When we search for the limit of g as

$$\lim_{\zeta\to\infty} g(\zeta,\omega) = \lim_{\zeta\to\infty} \frac{1-\omega\zeta}{1+(1-\omega)\zeta} = -\frac{\omega}{1-\omega}$$

then we find that $g(\zeta,\omega) \leq -1$ for $\omega \leq \frac{1}{2}$. This finding means that the backward Euler and trapezoidal rule provide numerical converging algorithms, which do not depend on the choice of sampling time ΔT. In contrast, the forward Euler method only converges towards zero if $\zeta = -\alpha\Delta T < 2$ or equally $\Delta T < -\frac{2}{\alpha}$. Additionally, we see in Eq. (5.6) that $g(\zeta,\omega) = 0$ for $\zeta = \frac{1}{\omega}$ and $\omega > 0$. So, we can reach the final state $z(N_T) = 0$ in one step in Eq. (5.7) as

$$z(t_1) = g\left(\frac{1}{\omega},\omega\right) z(0) = 0 \quad \text{and} \quad z(t_1) = z(t_2) = \ldots = z(t_{N_T}) = 0$$

if we choose either the forward Euler or trapezoidal rule.

We compare the numerical results of the one-step iteration algorithms in Eq. (5.5) with the analytical solution (5.4). We fix parameter $\alpha = -1$ and initial value $z(0) = 1$ and we choose sampling time $\Delta T = 0.5$, which guarantees a converging numerical solution. In Fig. 5.3, we visualize the analytical solution and the resulting numerical iterations

$$\begin{aligned}
z(t) &= exp(-t) && \text{(analytical solution)},\\
z(t_{n+1}) &= \left(\frac{1}{2}\right)^n && \text{(forward Euler)},\\
z(t_{n+1}) &= \left(\frac{3}{5}\right)^n && \text{(backward Euler)},\\
z(t_{n+1}) &= \left(\frac{2}{3}\right)^n && \text{(trapezoidal rule)}.
\end{aligned}$$

We see that the trapezoidal rule approximates the analytical solution better than the forward and backward Euler method in this example. This finding is only a specific result for the mentioned example and can not be stated in general.

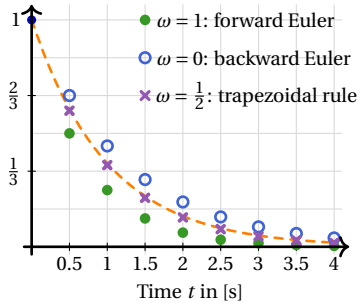


Figure 5.3: Comparison of the iteration algorithms in Eq. (5.5) with the analytical solution (5.4) (orange line).

Linear Heat Equation with Insulated Boundaries

We transfer the general concepts of the integration methods to the linear heat equation with transformed states as in Eq. (4.49) to demonstrate the applicability. In particular, we consider a system with insulated boundary conditions as

$$\frac{d}{dt}\tilde{\Theta}(t) = \tilde{A}_{N_d}\tilde{\Theta}(t) \tag{5.8}$$

in which $N_d \in \{1,2,3\}$ denotes the number of dimensions and

$$\tilde{A}_{N_d} = \text{diag}(\mu_1,\ldots,\mu_{N_c})$$

with $N_c = N_j \cdot N_m \cdot N_k$ as the number of temperature cells. We emphasize that Eq. (5.8) deals as a test system to analyze the numerical integration methods and the findings of this analysis shall be finally applied on the original quasilinear heat equation (3.29) with boundary conditions.

We apply the one-step iteration scheme (5.2) on the transformed heat equation (5.8) and we obtain

$$\tilde{\Theta}(t_{n+1}) = M\,\tilde{\Theta}(t_n) = M^n\,\tilde{\Theta}(0)$$

with diagonal matrix

$$\begin{aligned} M &:= (I-(1-\omega)\Delta T\tilde{A}_{N_d})^{-1}(I+\omega\Delta T\tilde{A}_{N_d}) \\ &= \mathrm{diag}(m_1,\ldots,m_{N_c}) \quad \text{and} \\ M^n &= \mathrm{diag}(m_1^n,\ldots,m_{N_c^n}) \end{aligned}$$

in which the elements are noted as[4]

$$m_i = \frac{1+\omega\Delta T\mu_i}{1-(1-\omega)\Delta T\mu_i}$$

for $i \in \{1,\ldots,N_c\}$. We note $m_i \equiv g(-\Delta T\mu_i,\omega)$ and we compare this finding with the ideas from the previous paragraph, see Fig. 5.2. So, we formulate

$$m_i \begin{cases} >0 & \text{if } \omega = 0, \\ \in(-1,1] & \text{if } \omega \in (0,\frac{1}{2}], \\ \in(-\infty,1] & \text{if } \omega \in (\frac{1}{2},1]. \end{cases}$$

Increasing the iteration $n \to \infty$, we distinguish the four scenarios

$$m_i^n \begin{cases} \to 0 & \text{if } m_i \in (-1,0)\cup(0,1), \\ =0 & \text{if } m_i = 0, \\ =\pm 1 & \text{if } m_i = -1, \\ \to \pm\infty & \text{if } m_i < -1. \end{cases} \tag{5.9}$$

If $(1+\omega\Delta T\mu_i) < 0$, then we yield numerical oscillations[5] as

$$m_i^n \begin{cases} <0 & \text{if } n \text{ is odd}, \\ >0 & \text{if } n \text{ is even}. \end{cases}$$

Hence, we desire all diagonal elements as $m_i \in (-1,1]$, $i \in \{1,\ldots,N_c\}$ such that all temperature values converge towards zero as $\tilde{\Theta}(t_n) \to 0$. We denote the iteration algorithm as *numerically stable* if the states show this convergence. Moreover, if we have $(1+\omega\Delta T\mu_i) > 0$, then $m_i \in (0,1]$ and we avoid numerical oscillations.

All eigenvalues μ_i are sorted in matrix $\tilde{A}_{N_d}$ as

$$0 = \mu_1 > \mu_2 > \mu_3 > \ldots > \mu_{N_c} \approx -4[p_1+p_2+p_3]$$

with $p_1 = \frac{\alpha_1}{\Delta x_1^2}$, $p_2 = \frac{\alpha_2}{\Delta x_2^2}$ and $p_3 = \frac{\alpha_3}{\Delta x_3^2}$ in the three-dim. case, see Eq. (4.37). This sorting also applies to m_i and we find $m_1 = 1$ as the largest entry and

$$m_{N_c} = \frac{1+\omega\Delta T\mu_{N_c}}{1-(1-\omega)\Delta T\mu_{N_c}}$$

as the smallest entry. If $m_{N_c} \in (-1,1)$ then all other m_i are inside this interval, too. We summarize these concepts in the following definition.

[4] We find these matrix elements because of the diagonal structure of $\tilde{A}_{N_d}$, see also Section 4.2.

[5] This does not occur for the backward Euler method, $\omega = 0$.

Definition 5.1 (Numerical Stability of the One-Step Iteration)
The one-step iteration of the transformed linear heat equation (5.8) converges towards zero, $\tilde{\Theta}_i(t_n) \to 0$ for $i \in \{2,\dots,N_c\}$, if

$$m_i = \frac{1+\omega\Delta T\mu_i}{1-(1-\omega)\Delta T\mu_i} \in (-1,1) \tag{5.10}$$

as in Eq. (5.9), and we denote the iteration as **numerically stable**.
Otherwise, if $m_i < -1$ then the iteration tends to $\pm\infty$ and we call it **numerically unstable**. ◯

We conclude from the previous ideas and Definition 5.1 the following statements.

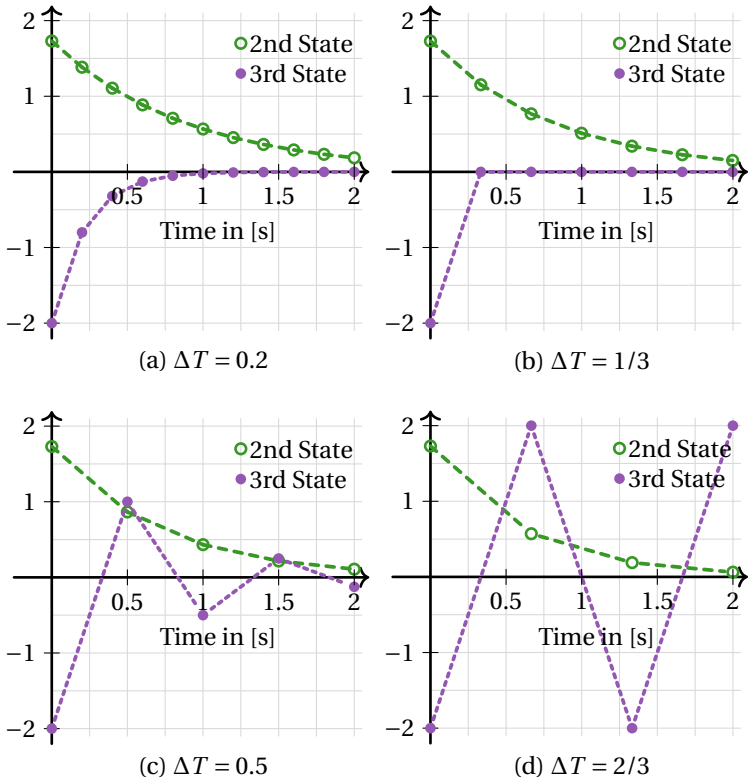
- If condition (5.10) holds for $i = N_c$, then it holds also for $i \in \{2,\dots,N_c-1\}$ because $m_1 > \dots > m_{N_c}$.
- If we choose $\omega \in \left[0, \frac{1}{2}\right]$, e.g. backward Euler or trapezoidal rule, then condition (5.10) holds for all $\Delta T > 0$.
- If we set $\omega \in \left(\frac{1}{2}, 1\right]$, e.g. forward Euler, then we need to choose the sampling time as
$$\Delta T \in \left(0, \frac{-2}{\mu_{N_c}[2\omega-1]}\right) \tag{5.11}$$
to guarantee a numerically stable one-step iteration.

In the literature, we find that a numerical integration method is called *A-stable* if the iteration converges for any sampling time $\Delta T > 0$, this is the case for the backward Euler method and the trapezoidal rule (or Crank-Nicolson method). The forward Euler method is not *A-stable* because the sampling time has to be $\Delta T \in \left(0, \frac{-2}{\mu_{N_c}}\right)$ to yield a converging iteration. If a numerical integration is *A-stable* and the iterator term

$$m_i \equiv g(\Delta T\mu_i, \omega) > 0$$

for any sampling time $\Delta T > 0$ and eigenvalue $\mu_i < 0$, then the method is called *L-stable*. The backward Euler method is *L-stable*, but not the trapezoidal rule. In the literature, we find *A-stability* in [91] and [92, p. 42], and *L-stability* in [92, p. 45] and [93, p. 7].

The forward Euler method is a standard approach to solve differential equations numerically, because it is very simple to implement. Though, its performance and the quality of results depend strongly on the choice of the sampling time ΔT. As heat conduction problems usually operate slowly, we wish to choose a large ΔT, but this may lead to numerical instabilities. On the other hand the backward Euler method and the trapezoidal rule provide numerically stable approaches to solve our heat conduction problems. A drawback of these methods is the task to solve an implicit equation, e.g. via the computation of a matrix inverse, which could be computationally costly. We finish this section with a numerical evaluation of the forward Euler method applied on a small heat conduction problem.



(a) $\Delta T = 0.2$ (b) $\Delta T = 1/3$

(c) $\Delta T = 0.5$ (d) $\Delta T = 2/3$

Figure 5.4: Application of the forward Euler method on linear heat equation (5.12). The second state converges smoothly to zero for all sampling times. The third state operates smoothly only for $\Delta T \in (0, \frac{1}{2})$ and oscillates for $\Delta T \in (\frac{1}{2}, \frac{2}{3})$.

Example: Forward Euler Method for One-dimensional Problem

We apply the forward Euler method on a small-scale one-dim. linear equation $\frac{d}{dt}\Theta(t) = A_1\Theta(t)$ with

$$A_1 = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

to demonstrate the numerical stability, see also the example in Section 4.3. We transform the linear differential equation with $\tilde{\Theta}(t) = \overline{V}\Theta(t)$ to

$$\frac{d}{dt}\begin{pmatrix} \tilde{\Theta}_1(t) \\ \tilde{\Theta}_2(t) \\ \tilde{\Theta}_3(t) \end{pmatrix} = \begin{pmatrix} 0 & & \\ & -1 & \\ & & -3 \end{pmatrix}\begin{pmatrix} \tilde{\Theta}_1 \\ \tilde{\Theta}_2 \\ \tilde{\Theta}_3 \end{pmatrix}$$

in which $\overline{V} = [\overline{\psi}_1, \overline{\psi}_2, \overline{\psi}_3]$ denotes the orthonormal eigenvectors, see Section 4.2, and we derive with Eq. (5.2) and $\omega = 1$ the forward Euler iteration formula

$$\tilde{\Theta}_1(n+1) = m_1\,\tilde{\Theta}_1(n) = \tilde{\Theta}_1(n), \tag{5.12a}$$

$$\tilde{\Theta}_2(n+1) = m_2\,\tilde{\Theta}_2(n) = [1-\Delta T]\,\tilde{\Theta}_2(n), \tag{5.12b}$$

$$\tilde{\Theta}_3(n+1) = m_3\,\tilde{\Theta}_3(n) = [1-3\Delta T]\,\tilde{\Theta}_3(n). \tag{5.12c}$$

In accordance with Definition 5.1, we seek for the maximum sampling time ΔT such that $m_i \in (-1,1)$ for $i \in \{1,2,3\}$. The smallest eigenvalue $\mu_3 = -3$ corresponds to $m_3 = 1-3\Delta T$ in Eq. (5.12c). So, we find that forward Euler method is numerically stable, if the sampling time is inside the interval $\Delta T \in (0, \frac{2}{3})$. In Table 5.1, we note m_2 and m_3 for five sampling

ΔT	$\frac{1}{5}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{2}{3}$	1
$m_2 = 1-\Delta T$	$\frac{4}{5}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{3}$	0
$m_3 = 1-3\Delta T$	$\frac{3}{5}$	0	$-\frac{1}{2}$	-1	-2

Table 5.1: Analysis of the sampling time for the example system (5.12) of the Euler method.

times and we see for these cases that $m_2 \geq 0$. Hence, only the last state $\tilde{\Theta}_3$ does not converge to zero while $\tilde{\Theta}_1$ and $\tilde{\Theta}_2$ do so. We portray the states $\tilde{\Theta}_2$ and $\tilde{\Theta}_3$ in Fig. 5.4 for the sampling times $\Delta T \in \{0.2, \frac{1}{3}, 0.5, \frac{2}{3}\}$ to emphasize the numerical stability for $\Delta T < \frac{2}{3}$.

5.2 Runge-Kutta Integration Methods

We return to the initial ideas of the one-step methods where we assume the function $z : [0, T_f] \to \mathbb{R}$ as the solution of a differential equation $\frac{d}{dt}z(t) = f(z,t)$. We find the solution at time $t + \Delta T$ with time step $\Delta T > 0$ as

$$z(t+\Delta T) = z(t) + \Delta T \frac{d}{dt}z(t) + \mathcal{O}(\Delta T^2). \tag{5.13}$$

The term $\mathcal{O}(\Delta T^2)$ summarizes all remaining higher-order terms of the approximation. Reshaping Eq. (5.13) and considering $\frac{\mathcal{O}(\Delta T^2)}{\Delta T} = \mathcal{O}(\Delta T) \to 0$ for $\Delta T \to 0$ leads us to the first-order finite difference approximation

$$\frac{d}{dt}z(t) \approx \frac{1}{\Delta T}\left[z(t_{n+1}) - z(t_n)\right]$$

with $t = t_n := n\Delta T$ and we note Eq. (5.13) as the forward Euler method

$$z(t_{n+1}) = z(t_n) + \Delta T f(z(t_n), t_n)$$

as in Eq. (5.2). If we take higher-order derivatives into account as

$$z(t+\Delta T) = z(t) + \Delta T \frac{d}{dt}z(t) + \frac{\Delta T^2}{2}\frac{d^2}{dt^2}z(t) + \mathcal{O}(\Delta T^3), \tag{5.14}$$

then we need to approximate the second-order derivative $\frac{d^2}{dt^2}z(t) = \frac{d}{dt}f(z,t)$ to find the one-step iteration. We approach this second-order derivative as

$$\begin{aligned}\frac{d^2}{dt^2}z(t) = \frac{d}{dt}f(z,t) &\approx \frac{1}{\Delta T}\left[f(z+\Delta z, t+\Delta T) - f(z,t)\right] \\ &\approx \frac{1}{\Delta T}\left[f(z+\Delta T f(z,t), t+\Delta T) - f(z,t)\right]\end{aligned}$$

with $\Delta z \approx z(t+\Delta T) - z(t) \approx \Delta T f(z,t)$ and we reformulate Eq. (5.14) as

$$\begin{aligned}z(t+\Delta T) &\approx z(t) + \Delta T f(z,t) \\ &\quad + \frac{\Delta T}{2}\left[f(z+\Delta T f(z,t), t+\Delta T) - f(z,t)\right] \\ &= z(t) + \frac{\Delta T}{2}\left[f(z,t) + f(z+\Delta T f(z,t), t+\Delta T)\right]\end{aligned}$$

to obtain the one-step algorithm

$$z_{n+1} = z_n + \frac{\Delta T}{2}\left[f(z_n, t_n) + f\left(z_n + \Delta T f(z_n, t_n),\ t_n + \Delta T\right)\right]. \tag{5.15}$$

0					
c_2	$a_{2,1}$				
c_3	$a_{3,1}$	$a_{3,2}$			
$\vdots$	$\vdots$		$\ddots$		
$c_{N_{st}}$	$a_{N_{st},1}$	$a_{N_{st},2}$	$\dots$	$a_{s,s-1}$	
	b_1	b_2	$\dots$	b_{s-1}	b_s

Table 5.2: Butcher tableau of the explicit Runge-Kutta method, see Eq. (5.16,5.17).

with the states $z_n = z(t_n)$. In fact, this one-step iteration (5.15) is a Runge-Kutta method with two stages. Runge-Kutta methods consist of nested terms of the right-hand side function f and the number of these terms is called *stages*. We may note the iteration (5.15) as the general 2-stage Runge-Kutta approach

$$z(t_{n+1}) \approx z(t_n) + \Delta T\,[b_1 k_1 + b_2 k_2]$$

with the coefficients

$$b_1 = \frac{1}{2}\,, \quad k_1 = f(z_n, t_n) \quad \text{and}$$
$$b_2 = \frac{1}{2}\,, \quad k_2 = f(z_n + \Delta T\; k_1, t_n + \Delta T).$$

We note the one-step Runge-Kutta iteration as

$$z(t_{n+1}) = z(t_n) + \Delta T \sum_{s=1}^{N_{st}} b_s\; k_s \tag{5.16}$$

with the number of stages $N_{st} > 0$ and the stages

$$k_s = f\left(z_n + \Delta T \left[\sum_{m=1}^{s-1} a_{s,m} k_m\right], t_n + c_s \Delta T\right). \tag{5.17}$$

The Runge-Kutta coefficients $a_{s,m}, b_s, c_s \in \mathbb{R}$ are usually noted in a Butcher tableau[6], see Table 5.2. The Runge-Kutta iteration (5.16) with stages (5.17) form the *explicit Runge-Kutta algorithm*. In the literature, we find many Runge-Kutta approaches with various coefficients, which need to fulfill certain conditions, for example

[6] Named after John C. Butcher (* 1933) [94].

$$c_s = \sum_{m=1} a_{s,m},$$
$$\sum_{s=1}^{N_{st}} b_s = b_1 + \ldots + b_{N_{st}} = 1.$$

The choice of coefficients depend on the desired Runge-Kutta order and these conditions guarantee a proper working algorithm, see more details in [95, p. 132, 134]. We apply the differential equation $\frac{d}{dt} z(t) = \alpha z(t)$ with $\alpha < 0$ and $z(0) \neq 0$ on the Runge-Kutta algorithm (5.16) to check the numerical stability as in Section 5.1. We evaluate the nested stages (5.17) in Eq. (5.16) to yield the one-step iteration

$$z(t_{n+1}) = \left[1 + \sum_{s=1}^{N_{st}} \beta_s\,[\alpha \Delta T]^s\right] z(t_n) \tag{5.18}$$

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

(a) Classic

0				
$\frac{1}{3}$	$\frac{1}{3}$			
$\frac{2}{3}$	$-\frac{1}{3}$	1		
1	1	-1	1	
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

(b) 3/8-Rule

Table 5.3: Coefficients of the original Runge-Kutta method.

with coefficients $\beta_s \in \mathbb{R}$. We introduce the variable $\zeta = -\alpha\Delta T$ again, define the iterator

$$g(\zeta) := 1 + \sum_{s=1}^{N_{st}} \beta_s (-\zeta)^s \tag{5.19}$$

and formulate Eq. (5.18) as

$$z(t_{n+1}) = g(\zeta)\ z(t_n) = g(\zeta)^n\ z(0). \tag{5.20}$$

If the iterator $|g(\zeta)| < 1$ then we see that the iteration (5.18) converges towards zero. Though, the iterator g is a polynomial and we know that for some $\tilde{\zeta} \in \mathbb{R}$ we find $|g(\tilde{\zeta})| > 0$. Hence, the choice of time step $\Delta T > 0$ depends on the system parameter $\alpha < 0$, and so the explicit Runge-Kutta methods are not A-stable.

We exemplify these ideas with the fourth-order Runge-Kutta approach by Martin Kutta [96], see also [95, p. 137]. The coefficients are noted in the Butcher tableaux 5.3 and both tableaux lead to the same iteration with iterator function

$$g(\zeta) = 1 - \zeta + \frac{1}{2}\zeta^2 - \frac{1}{6}\zeta^3 + \frac{1}{24}\zeta. \tag{5.21}$$

The graph of iterator g in Fig. 5.5 does not drop below zero and so the numerical solution of the test differential equation does not oscillate for any time step $\Delta t > 0$, but $g(\tilde{\zeta}) > 1$ for $\tilde{\zeta} \approx 2.8$. The exact value of stability limit $\tilde{\zeta}$ might be found by solving the quartic equation $g(\tilde{\zeta}) = 0$ algebraically. At the value $\zeta_{min} \approx 1.6$ we find the minimum of $g(\zeta)$, and so we yield the fastest convergence of the Runge-Kutta iteration for a time step $\Delta T \approx \frac{1.6}{-\alpha}$. We evaluate iteration (5.20) with g as in Eq. (5.21) for $\zeta \in \{0.6, 1.6, 2.6\}$ and we notice in Fig. 5.6 that smaller time steps, e.g. $\Delta T = 0.6$, lead to rather accurate numerical results. Summing up the recent findings, we note that the classic fourth-order explicit Runge-Kutta method has a larger area of stability, and guarantees larger time steps, then the forward Euler method, but we need for both approaches small time steps to gain a numerical exact solution.

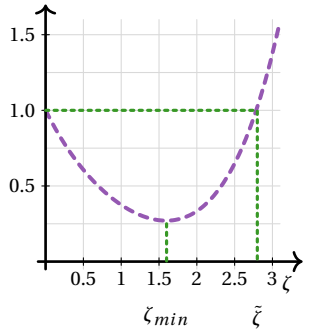


Figure 5.5: Runge-Kutta iterator $g(\zeta)$ with stability limit at $\zeta \approx 2.8$ and minimum at $\zeta \approx 1.6$.

Implicit Runge-Kutta Methods

Heat conduction phenomena consist of very fast and slow components because of their wide-ranged eigenvalue distribution, see Chapter 4. Explicit numerical solvers like the forward Euler or the explicit Runge-Kutta methods are not practical for this situation because we need to choose a (very) small time step to guarantee a stable and exact numerical solution. Hence, we need to apply implicit numerical solvers like the backward Euler method or implicit Runge-Kutta methods. The latter approach has an

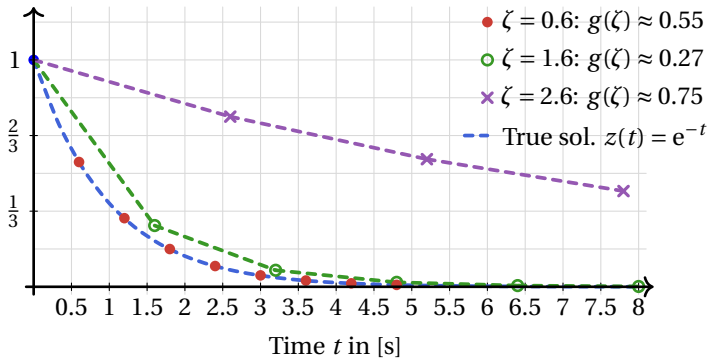


Figure 5.6: Runge-Kutta iteration (5.20) for $\zeta = 0.6$, $\zeta = 1.6$ and $\zeta = 2.6$.

c_1	$a_{1,1}$	$a_{1,2}$	…	$a_{1,N_{st}}$
c_2	$a_{2,1}$	$a_{2,2}$	…	$a_{2,N_{st}}$
⋮	⋮		⋱	
$c_{N_{st}}$	$a_{N_{st},1}$	$a_{s,2}$	…	$a_{N_{st},N_{st}}$
	b_1	b_2	…	$b_{N_{st}}$

Table 5.4: Butcher tableau of fully implicit Runge-Kutta methods.

iteration (5.16) with the stages

$$k_s = f\left(z_n + \Delta T\left[\sum_{m=1}^{N_{st}} a_{s,m} k_m\right], t_n + c_s \Delta T\right). \tag{5.22}$$

In each stage k_s of the fully implicit Runge-Kutta method we sum up all stages, in contrast to the explicit approach where we sum up only $s-1$ stages, see Eq. (5.17). So, we yield a large-scale system of implicit (nonlinear) equations. The coefficients of the fully implicit Runge-Kutta method are stored in the Butcher tableau 5.4. There exist several sub-types of implicit Runge-Kutta methods, see [98, 99]. We list three of them below.

1. Diagonally Implicit Runge–Kutta methods (DIRK): the summation in stage k_s terminates at index s as

$$k_s = f\left(z_n + \Delta T\left[\sum_{m=1}^{s} a_{s,m} k_m\right], t_n + c_s \Delta T\right). \tag{5.23}$$

So, we have in the Butcher tableau a triangular A coefficient matrix

$$\begin{pmatrix} a_{1,1} & & & & \\ a_{2,1} & a_{2,2} & & & \\ a_{3,1} & a_{3,2} & a_{3,3} & & \\ \vdots & \vdots & \ddots & \ddots & \\ a_{N_{st},1} & a_{N_{st},2} & \dots & & a_{N_{st},N_{st}} \end{pmatrix}. \tag{5.24}$$

2. *Singly Diagonally Implicit Runge-Kutta* methods (SDIRK): all diagonal elements of the triangular A matrix are equal as

$$a_{1,1} = a_{2,2} = \ldots = a_{N_{st},N_{st}} = \gamma.$$

3. *Explicit Singly Diagonally Implicit Runge-Kutta* methods (ESDIRK): the first coefficient $a_{1,1} = 0$ and all other diagonal elements are equal as

$$a_{1,1} = 0\,,\; a_{2,2} = a_{3,3} = \ldots = a_{N_{st},N_{st}} = \gamma.$$

We implement the numerical simulation with the JULIA programming language [100] using the software library *DifferentialEquations.jl* [101]. Here, we call the ESDIRK numerical solver *KenCarp5*, see [102], because it performs well for medium- and large-sized heat conduction problems. In the subsequent section, we briefly compare the numerical solvers backward Euler, trapezoidal rule and ESDIRK/KenCarp5 to motivate the further choice of the latter algorithm.

5.3 Numerical Error of Time Integration Methods

We evaluate the numerical integration methods, backward Euler, trapezoidal rule and ESDIRK/KenCarp5 with an one-dim. example. The analytical solution is derived in Appendix A.1 and provides the true temperature, which is compared with numerical results. We consider the linear heat equation (A.1) with insulated boundary conditions (A.3) and the symmetric initial temperatures (A.2). We have a length $L = 0.2$ meter, material properties $\lambda = 50$, $\rho = 8000$, $c = 400$ and so we calculate a diffusivity

$$\alpha = \frac{\lambda}{c\,\rho} = 15.625 \cdot 10^{-6}.$$

We obtain the true data[7] ϑ_{true} via an evaluation of Eq. (A.14) with maximum iteration number $k = 100$ and we compute the numerical solution $\hat{\theta}$ with sampling time $\Delta T = 10$ seconds. We evaluate the error for 10 cases with respect to the number of temperature nodes $N_j \in \{10, 20, \ldots, 100\}$, which imply the spatial sampling

$$\Delta x_1 = \frac{L}{N_j} \in \{0.02, 0.01, \ldots, 0.002\}.$$

[7] The true data is still an approximation because we need to terminate the series in Eq. (A.14).

We obtain the error as the quadratic difference of temperatures along the rod in

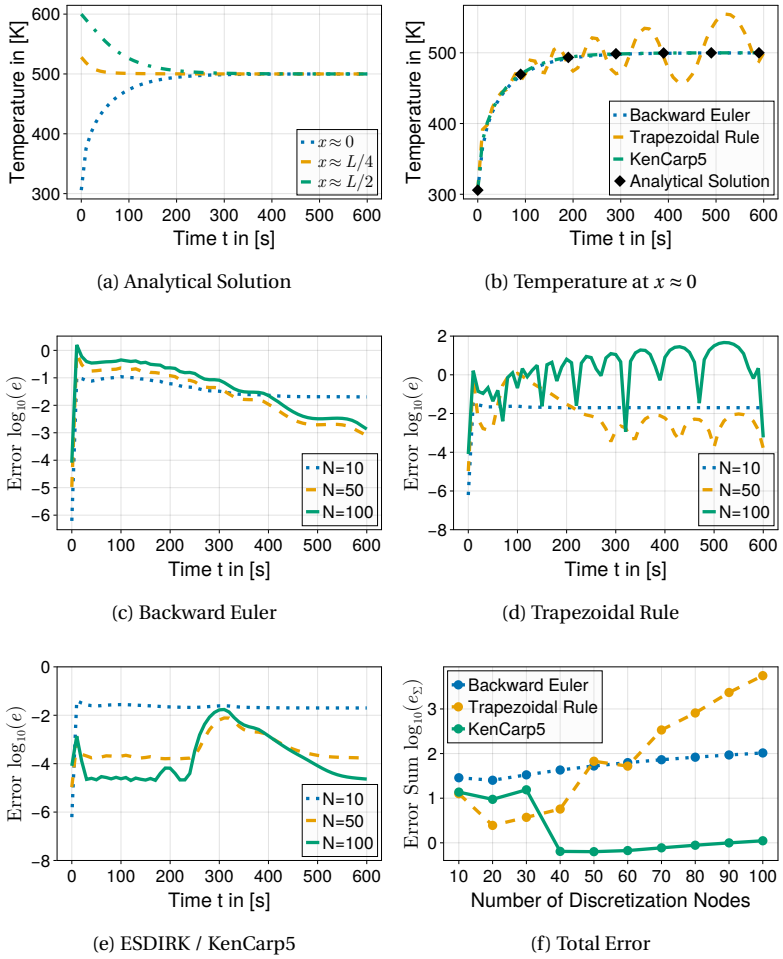
$$\begin{aligned} e(t) &= \int_0^L \left[\vartheta_{true}(t,x) - \hat{\theta}(t,x)\right]^2 dx \\ &\approx \sum_{i=1}^{N_j} \left[\vartheta_{true}(t,x_i) - \hat{\theta}(t,x_i)\right]^2 \Delta x \end{aligned} \tag{5.25}$$

and we sum up the error over the time as

$$e_\Sigma = \int_0^{T_f} e(t)\,dt \approx \sum_{n=0}^{N_T} e(t_n)\Delta T \tag{5.26}$$

with $N_T = \left\lfloor \frac{T_f}{\Delta T} \right\rfloor + 1$.

We solve the spatially approximated heat equation with the backward Euler method, trapezoidal rule and ESDIRK/KenCarp5 for $T_f = 600$ seconds and we visualize the results in Fig. 5.8. The analytical solution is symmetric, see Fig. A.2, and so we depict the true temperature evolution only at three positions, $x \in \{0.001, 0.051, 0.101\}$ meter in Fig. 5.8 (a). We compare the numerical solutions for $N_j = 100$ at the first node in Fig. 5.8 (b) and we find that the backward Euler method and KenCarp5 are very close to the true data, but the trapezoidal rule drives into numerical oscillations. We portray the numerical error (5.25) of the integration methods



(a) Analytical Solution

(b) Temperature at $x \approx 0$

(c) Backward Euler

(d) Trapezoidal Rule

(e) ESDIRK / KenCarp5

(f) Total Error

Figure 5.8: Evaluation of the numerical error of the backward Euler method, trapezoidal rule and ESDIRK/KenCarp5. The analytical solution in (a) approaches the steady-state temperature for $t > 300$ seconds. The backward Euler method and KenCarp5 coincide with the true temperature at $x \approx 0$ in (b), but the trapezoidal rule tends to numerical oscillations. The backward Euler method shows in (c) almost the same numerical error for all cases, where the error is larger in the beginning until the temperatures are settled at 500 Kelvin, for $t \in [0,300]$. The numerical error of the trapezoidal rule in (d) exhibits even oscillations for $N_j = 50$ and remarkable ones for $N_j = 100$. KenCarp5 has a small numerical error in (e) for $N_j \in \{50,100\}$, but we notice a peak at $t = 300$, where the thermal behavior transits from diffusion to steady-state. We note in (f) that the total error e_Σ of the trapezoidal rule increases for an increasing number of nodes, e.g. $N_j > 20$, while it drops in case of KenCarp5 for $N_j > 30$.

in Fig. 5.8 (c) to (e) for $N_j \in \{10,50,100\}$ in logarithmic scale $\log_{10}(e(t))$. Here, we notice that all solvers unveil an almost constant and equal error for $N_j = 10$. The backward Euler method in Fig. 5.8 (c) has a similar numerical error for all $N_j \in \{10,50,100\}$, while the error of KenCarp5 in Fig. 5.8 (e) decreases significantly for finer approximations. The trapezoidal rule shows an oscillatory numerical error in Fig. 5.8 (d) for $N_j = \{50,100\}$.

We yield a deeper insight of this numerical error in Fig. 5.8 (f), where the summed up numerical error (5.26) of the trapezoidal rule has a minimum at $N_j = 20$ and increases for a higher number of nodes. This poor performance is caused by the fact that matrix entry m_{N_c} approaches the stability limit for a high number of nodes, e.g. $m_{N_c} \approx -1$ for $N_j = 100$, see also Definition 5.1. In Fig. 5.7, we portray the value of m_{N_c} and we find that increasing numerical error in Fig. 5.8 (f) corresponds to value of m_{N_c} close to the stability limit.

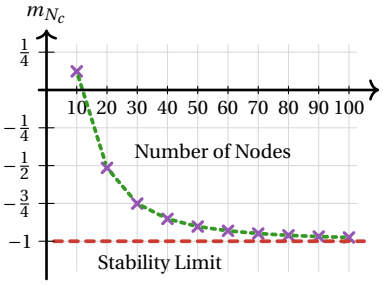


Figure 5.7: The trapezoidal rule is approaching the stability limit because the smallest matrix entry $m_{N_c} \approx -1$ for an increasing number of nodes N_j.

Control System Design

6

Control System Framework

The thermal dynamics is affected on the boundaries of the considered geometry as explained in Section 2.4. We recapitulate that cooling is a purely natural process, which is caused by thermal emissions $\phi_{em}(t,x)$ as linear heat transfer and nonlinear heat radiation, see Section 2.5. Thus, it is not manipulated by a technical operator. In contrast to that, the heating is driven by heat supply $\phi_{in}(t,x)$ via actuators and their input signals are computed with control units. The control system has two tasks: it has to increase the measured temperatures from an initial towards a final operating point with

$$P_{in}(t) = \int_{B_{in}} \phi_{in}(t,x)dx > \int_{\partial\Omega} \phi_{em}(t,x)dx = P_{em}(t) \tag{6.1}$$

and it has to avoid a temperature drop after reaching the desired operating temperature with

$$P_{in}(t) = \int_{B_{in}} \phi_{in}(t,x)dx = \int_{\partial\Omega} \phi_{em}(t,x)dx = P_{em}(t). \tag{6.2}$$

If the initial temperature is close to the ambient temperature, then the emissions are quite small and we simply find a proper heat supply to guarantee condition (6.1). When the temperature difference between object and surrounding increases, the computation of a suitable large heat supply is more difficult. We face two main issues: firstly, the area of actuation is in many scenarios smaller than the area of thermal emissions; and secondly, we usually do not measure the temperature on all boundary sides to determine the complete emitted heat flux.[1] As a consequence, the exact amount of emitted power P_{em} in Ineq. (6.1) and Eq. (6.2) is unknown and we need to find a good estimate of this quantity to yield a proper working control system. We exemplify a scenario with one actuator and and one sensor on the opposite sides in Fig. 6.1, where only a part of the full thermal emissions can be recorded. Moreover, a slow temperature propagation from the actuators towards the sensors impedes an exact stabilization of the closed-loop control system at the reached operating temperature. Therefore, we need an intelligent control architecture, which includes the extensive heat conduction model and computes proper input signals to steer and stabilize the heating process.

In this chapter, we state an overview about the control of the thermal dynamics. In Section 6.1, we specify the actuator and sensor models, and

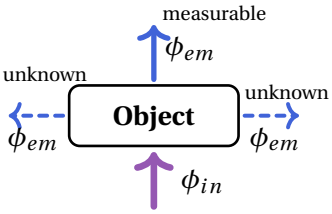


Figure 6.1: Heat conduction example with heat supply ϕ_{in} on one boundary side and thermal emissions ϕ_{em} on the other sides. The emissions on the left and right side are unknown, it can be "measured" only on the top.

[1] We also need the *true* coefficients of the heat transfer and heat radiation to compute the emitted heat flux with the measured temperatures.

we formulate the spatially approximated nonlinear thermal dynamics with supplied heat and temperature measurements. Afterwards, in Section 6.2, we introduce the principle of two-degrees-of-freedom control and we discuss the concepts of open-loop and closed-loop design in the context of our heat conduction framework.

6.1 Actuation and Temperature Measurement

In our control framework, we assume multiple, spatially distributed, actuators and sensors, which operate on the boundary sides. In Section 2.4, we defined the actuator boundary $B_{in} \subseteq \partial\Omega$, see Definition 2.2, and in an analogous way we define the sensor setup $B_{out} \subseteq \partial\Omega$. In the next paragraphs, we describe the location and the model of actuators and sensors, and we explain how these models are embedded in the heat conduction framework. These modeling approaches are presented in our articles [34, 35, 37, 40].

Actuator Setup

We explained in Section 2.4 that the heat is supplied on boundary $B_{in} \subseteq \partial\Omega$, see Definition 2.2. This heat supply is realized by multiple, spatially distributed, actuators operating on boundary B_{in} and we need to specify the location of them. We assume in total $N_u > 0$ actuators on B_{in} and we say that each actuator has its own segment, $\beta_n \subseteq B_{in}$ with $B_{in} = \bigcup_{n=1}^{N_u} \beta_n$. Segments do not overlap, $\bigcap_{n=1}^{N_u} \beta_n = \{\}$, and the size of all segments is equal

$$|\beta_1| = |\beta_2| = \ldots = |\beta_{N_u}|.$$

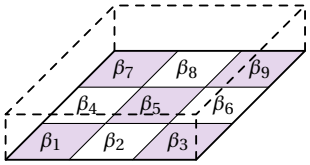


Figure 6.2: Example of a partition with nine segments on the underside of a cuboid.

A segment is only part of one boundary side, e.g. B_U. If more than one boundary side is actuated, then each boundary side has its own partition, for example the boundary sides B_U and B_E are partitioned as

$$B_U = \bigcup_{n=1}^{N_{u,1}} \beta_n \quad \text{and} \quad B_E = \bigcup_{n=N_{u,1}+1}^{N_{u,2}} \beta_n.$$

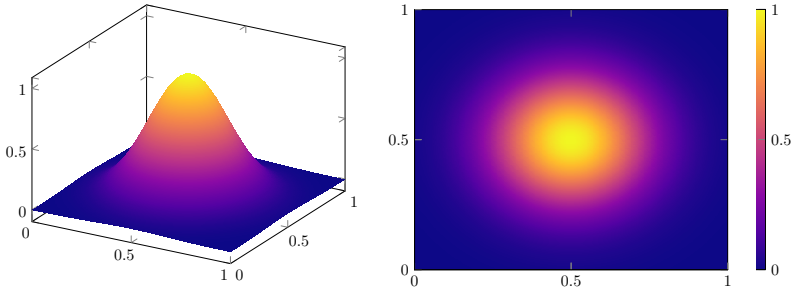
An example partition for a cuboid is portrayed in Fig. 6.1. An actuator has a spatial characteristics $b_n : B_{in} \to [0,1]$ and it determines how much power can be supplied in each position $x \in \beta_n$. We model this spatial characteristics via the exponential function

$$b_n(x) := \begin{cases} m_n \exp\left(-\|M_n(x - x_{c,n})\|^{2\nu_n}\right) & \text{for } x \in \beta_n, \\ 0 & \text{for } x \in B_{in} \setminus \beta_n \end{cases} \tag{6.3}$$

with scaling $m \in [0,1]$, (diagonal) curvature matrix $M \in \mathbb{R}^{N_d \times N_d}$, power $\nu \in \mathbb{N}_{>0}$, central point $x_{c,n} \in \beta_n$ of the n-th segment, and number of dimensions $N_d \in \{1,2,3\}$. In Eq. (6.3) we consider the Euclidean vector norm

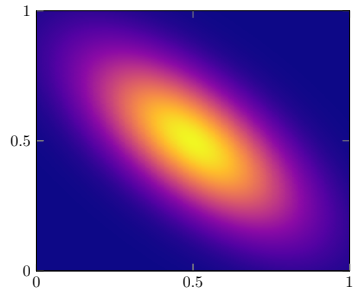
$$\|v\|_2 := \sqrt{\sum_{n=1}^{N} v_n^2}$$

with vector $v \in \mathbb{R}^N$ and so we have b_n as a Gaussian-shaped function. In Fig. 6.3, we visualize three shapes of b_n depending on curvature matrix

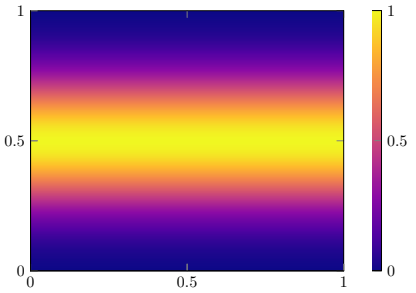


(a) Circular Gaussian

(b) Gaussian, top view



(c) Rotated Ellipse



(d) Stripe

Figure 6.3: Example shapes of spatial characteristics b_n as in Eq. (6.3). We have a standard radial Gaussian with a diagonal matrix M in (a) and (b). If M has elements on its sub-diagonal, then we yield a rotated elliptic shape in (c). If M has only one non-zero element, then we have striped shape in (d). We consider $\nu = 1$ and the curvature matrices

- $M = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ in (a) and (b),
- $M = \begin{pmatrix} 4 & 1.5 & 0 \\ 1.5 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ in (c) and
- $M = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ in (d)

for an actuation on boundary B_U.

M. We remark that one may also choose a different vector norm, e.g. the maximum norm

$$\|v\|_\infty := \max(v_1, \ldots, v_N)$$

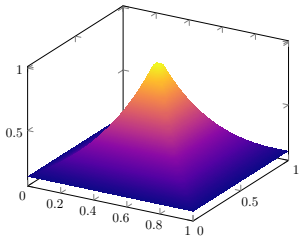
to model a sharp peak or box-shaped characteristics as depicted in Fig. 6.4. However, we only discuss the spatial characteristics with the Euclidean vector norm $\|v\|_2$ in the following examples of this thesis.

If we assume one- and two-dimensional geometries, then we can simplify the formulation of the spatial characteristics in Eq. (6.3). In the one-dim. case we only need to set a scale $m > 0$ and in case of a two-dim. geometry we obtain the simplified exponential function

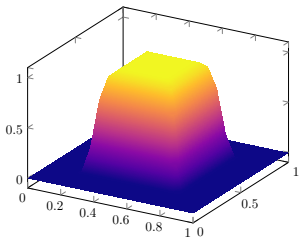
$$b_n(x) = m_n \exp\left(-|M_n(x - x_{c,n})|^{2\nu_n}\right)$$

with scalar $M > 0$ for $x \in \beta_n$.

In practice, the design of spatial characteristics b_n is not trivial and requires a good knowledge of the actuator's physical behavior. The modeling and system identification of thermal actuators is an active field of research and we find examples in the literature regarding resistive heating [104], inductive heating in [105] and micro-hotplates [106]. In Chapter 1, we stated two examples of controlled thermal processes: laser beam welding and post-exposure bake as part of lithography. Here, we find one important difference of both processes where we have small point-shaped sources in laser beam welding and wide uniform source like electrical heating elements in the post-exposure bake, see also the commercial product [107]. Hence, we need to care about suitable values of matrix M, norm p and power ν to specify the heat source properly.



(a) Sharp peak, $\nu = 1$



(b) Box shape, $\nu = 10$

Figure 6.4: Spatial characteristics with maximum norm
$b(x) = \exp\left(-\|M(x - x_c)\|_\infty^\nu\right)$ with

- $\nu = 1$ for a sharp peak in (a) and
- $\nu = 10$ for a box shape in (b).

Once we have the actuator model, we need to compute an input signal

$$u_n : [0, T_{final}) \rightarrow \mathbb{R}_{\geq 0}$$

for $n \in \{1, \ldots, N_u\}$ to influence the thermal dynamics properly. We sum up the spatial characteristics and the input signal and we formulate the supplied heat flux as

$$\phi_{in}(t,x) := \sum_{n=1}^{N_u} b_n(x)\, u_n(t). \tag{6.4}$$

Sensor Setup

We construct an analog concept for the temperature measurement with $N_y > 0$ sensors and boundary $B_{out} \subseteq \partial\Omega$. B_{out} has a partition with segments $\gamma_n \subseteq B_{out}$ such that $B_{out} = \bigcup_{n=1}^{N_y} \gamma_n$ and $\bigcap_{n=1}^{N_u} \gamma_n = \{\}$. The sensor segments γ_n belong to only one boundary side like the actuator segments β_n. We note the spatial characteristics of the sensors as

$$g_n(x) := \begin{cases} \tilde{m}_n \exp\left(-\|\tilde{M}_n(x - x_{c,n})\|_{\tilde{p}_n}^{\tilde{v}_n}\right) & \text{for } x \in \gamma_n, \\ 0 & \text{for } x \in B_{out} \setminus \gamma_n \end{cases} \tag{6.5}$$

for $n \in \{1, \ldots, N_y\}$, c.f. Eq. (6.3), and define the n-th measurement as

$$y_n(t) := \left[\int_{\gamma_n} g_n(x) dx\right]^{-1} \int_{\gamma_n} g_n(x)\, \vartheta(t,x) dx. \tag{6.6}$$

So, the temperature measurement at the n-th sensor is a weighted mean with spatial characteristics g_n as the weight.

Spatial Approximation of the Actuator and Sensor Setup

We need to approximate the actuator and sensor boundaries to evaluate the heat supply ϕ_{in} and temperature measurement y at the discrete nodes $x^i = x^{j,m,k}$. Thus, we store the indices of cells, which have boundaries being part of an actuator or sensor segment as

$$S_{\beta,n} := \{i = i(j,m,k) | \beta_n \subset \partial\Omega_{j,m,k}\}$$

for $n \in \{1, \ldots, N_u\}$ and

$$S_{\gamma,n} := \{i = i(j,m,k) | \gamma_n \subset \partial\Omega_{j,m,k}\}$$

for $n \in \{1, \ldots, N_y\}$. We find the issue that the finite volume approximation leads to nodes x^i inside the geometry and one grid node may have two or three adjacent boundary sides, e.g. at corners or along edges, see Section 3.3. So, we distinguish the supplied heat flux for each direction $l \in \{1,2,3\}$ as

$$\phi_l(t, x^i) = \phi_{in,l}(t, x^i) = \sum_{n=1}^{N_u} b_{l,n}(x^i)\, u_n(t).$$

Furthermore, we remark that the curvature matrix M has an entry $M_{l,l} = 0$ in function $b_{l,n}(x^i)$ and the same idea applies for $\tilde{M}$ of the sensor characteristics in Eq. (6.5).

In the following steps, we reformulate the spatially approximated quasi-linear heat equation (3.29) with the actuator and sensor setup to obtain a state space formulation of the complete control system. We note the nonlinear diffusion terms as

$$f_{\mathscr{D}}(\Theta) = \left[f_{\mathscr{D},1}(\Theta), \ldots, f_{\mathscr{D},N_c}(\Theta) \right]^\top \tag{6.7a}$$

with

$$f_{\mathscr{D},i}(\Theta) = \sum_{l=1}^{3} \mathscr{D}_l(\Theta_i, \Theta_{i-\mu}, \Theta_{i+\mu}) / s(\Theta_i) \tag{6.7b}$$

and we unify the specific heat capacity and the density as

$$s(\Theta_i) := \rho(\Theta_i)\ c(\Theta_i).$$

In accordance with Eq. (3.28), we note the approximated supplied heat flux as $\phi_{in,l}/\Delta x_l$ and so we find the approximated actuation as

$$\tilde{b}_{n,i} := \begin{cases} b_{l,n}(x^i)/\Delta x_l & \text{if } i \in \mathscr{S}_{\beta,n}, \\ 0 & \text{else} \end{cases}$$

where

$$l = \begin{cases} 1 & \text{if } \beta_n \subset B_W \cup B_E, \\ 2 & \text{if } \beta_n \subset B_S \cup B_N \text{ and} \\ 3 & \text{if } \beta_n \subset B_U \cup B_T. \end{cases}$$

The influence of the n-th actuator is noted as

$$\tilde{b}_n(\Theta) = \left[\frac{\tilde{b}_{n,1}}{s(\Theta_1)}, \ldots, \frac{\tilde{b}_{n,N_c}}{s(\Theta_{N_c})} \right]^\top \tag{6.8a}$$

and we aggregate all actuators in the temperature-dependent matrix

$$B(\Theta) = \left[\tilde{b}_1(\Theta), \ldots, \tilde{b}_{N_u}(\Theta) \right]. \tag{6.8b}$$

Additionally, we need to formulate the spatially approximated thermal emissions, see also Eq. (3.26). We introduce the set

$$\tilde{\mathscr{S}}_i := \{ l \in \{1,2,3\} \,|\, \mathrm{pos}(l,i) \neq 0 \}$$

to define the correspondence between direction $l \in \{1,2,3\}$ and global index $i = i(j,m,k)$. In a similar way, we denote the thermal emission of the i-th cell as

$$\tilde{w}_i(t,\Theta_i) := \begin{cases} \frac{1}{s(\Theta_i)} \sum\limits_{l \in \tilde{\mathscr{S}}_i} \phi_{em,l}(t,x^i)/\Delta x_l & \text{if } i \in S \setminus \mathring{\mathscr{S}}, \\ 0 & \text{else.} \end{cases}$$

We aggregate the thermal emissions for all cells as

$$w(t,\Theta) = \left[\tilde{w}_1(t,\Theta_1), \ldots, \tilde{w}_{N_c}(t,\Theta_{N_c}) \right]^\top \tag{6.9}$$

and we formulate the spatially approximated system dynamics as

$$\frac{d}{dt}\Theta(t) = f_{\mathscr{D}}(\Theta) + B(\Theta)\ u(t) + w(t,\Theta).$$

The spatial characteristics of the temperature measurement is approximated by

$$\tilde{c}_{n,i} = \begin{cases} g_{l,n}(x^i) & \text{if } i \in \mathscr{S}_{\gamma,n}, \\ 0 & \text{else} \end{cases}$$

where

$$l = \begin{cases} 1 & \text{if } \gamma_n \subset B_W \cup B_E, \\ 2 & \text{if } \gamma_n \subset B_S \cup B_N \text{ and} \\ 3 & \text{if } \gamma_n \subset B_U \cup B_T. \end{cases}$$

We collect all entries $\tilde{c}_{n,i}$ as vectors

$$\tilde{c}_n = \left[\tilde{c}_{n,1}, \ldots, \tilde{c}_{n,N_c}\right] \tag{6.10a}$$

of the n-th sensor with $n \in \{1, \ldots, N_y\}$ and we note the output matrix as

$$C = \left[\tilde{c}_1, \ldots, \tilde{c}_{N_y}\right]^\top. \tag{6.10b}$$

Consequently, we yield the output signal as

$$y(t) = C\ \Theta(t).$$

Definition 6.1 (Nonlinear and Linear State Space Formulation)
The approximated heat conduction phenomena with actuation and temperature measurement is described by the nonlinear state-space system

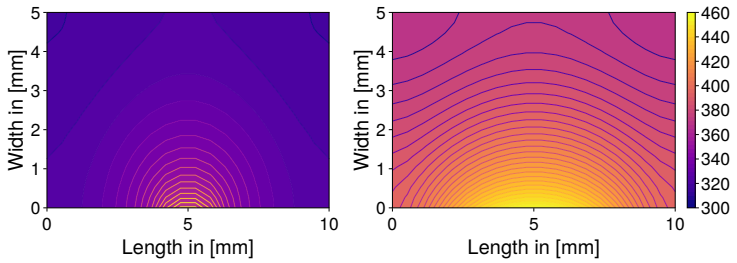
$$\frac{d}{dt}\Theta(t) = f_{\mathscr{D}}(\Theta) + B(\Theta)\ u(t) + w(t,\Theta), \tag{6.11a}$$

$$y(t) = C\ \Theta(t). \tag{6.11b}$$

The nonlinear system dynamics $f_{\mathscr{D}} : \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$ is formulated in Eq. (6.7) with the diffusion term $\mathscr{D}$ as in Eq. (3.27). Mapping $B : \mathbb{R}^{N_c} \to \mathbb{R}^{N_c \times N_u}$ is specified in Eq. (6.8). It connects the n-th input signal $u_n : [0,T) \to \mathbb{R}_{\geq 0}$ for $n \in \{1, \ldots, N_u\}$ with the spatial characteristics b_n of the n-th partition β_n. The input signals are restricted as $u_n(t) \geq 0$. The thermal emissions are stored in function $w : [0,T) \times \mathbb{R}^{N_c} \to \mathbb{R}^{N_c}$, see Eq. (6.9). The temperature data of all nodes is mapped to the output signals with matrix $C \in \mathbb{R}^{N_y \times N_c}$ as noted in Eq. (6.10).
If we consider constant material properties and no thermal emissions, as $w(t,\Theta) \equiv 0$, then we yield the **linear system dynamics**

$$\frac{d}{dt}\Theta(t) = A\ \Theta(t) + B\ u(t) \tag{6.12}$$

with matrices $A \in \mathbb{R}^{N_c \times N_c}$ and $B \in \mathbb{R}^{N_c \times N_u}$, see Definition 3.2. ◯



(a) Narrow Shape (b) Wide Shape

Figure 6.6: Temperature distribution of an actuation with a narrow and a wide spatial characteristics. The temperatures in (a) reach a maximum of ca. 370 Kelvin in a region close to the actuator. The maximum values in (b) reach up to ca. 450 Kelvin and a large region has temperatures above 400 Kelvin.

Example: Actuation of a Rectangular Object

We exemplify the design of the spatial characteristics for a two-dim. model with length $L = 0.1$ meter, width $W = 0.05$ meter and material properties $\lambda = 50$, $c = 400$, $\rho = 8000$. We assume one actuator on boundary B_S and thermal insulation on all remaining other sides, $\phi_{em} \equiv 0$. Regarding the spatial characteristics, we fix the scaling, power and central point as $m = 1$, $\nu = 2$ and $x_c = (0.05, 0)^\top$ and we distinguish a curvature with $M = 100$ for a narrow and $M = 30$ for a wide shape, see also Fig. 6.5.

We apply a constant input signal $u(t) = 2 \cdot 10^5$ and simulate the heat conduction for $T_{final} = 120$ seconds. The temperature distribution at the final time $t = T_{final}$ is visualized in Fig. 6.6. We find that the narrow actuator shape leads to low temperatures in Fig. 6.6 (a) and we only have a small hot region close to the actuator. In contrast, the wide shape results in higher temperatures in Fig. 6.6 (b), where a large region has temperatures above 400 Kelvin.

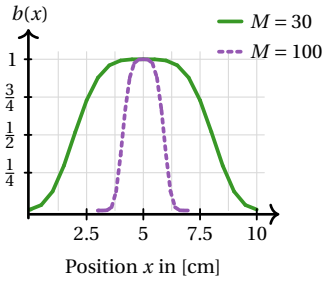


Figure 6.5: Spatial characteristics for actuation of a rectangular geometry. We distinguish a wide ($M = 30$) and narrow shape ($M = 100$).

6.2 Two-Degrees-of-Freedom Control Design

One of the main goals of this contribution is to find suitable input signals to heat up the object from an initial to a target temperature and keep it on this level. In the design of control systems, we deal with the feed-forward control to steer a system from one operating point to the next one, and we stabilize the system dynamics at the reached operating point with a feedback control. Therefore, we construct a feed-forward controller u_{ff} for the heating-up procedure in the time $t \in [0, T_{ff})$ and a feedback controller u_{mpc}[2] to prevent a cooling-down during $t \in [T_{ff}, T_{final})$ with $0 < T_{ff} < T_{final}$. We distinguish the controller type by the operation time as

$$u(t) = \begin{cases} u_{ff}(t) & \text{for } t \in [0, T_{ff}), \\ u_{mpc}(t) & \text{for } t \in [T_{ff}, T_{final}). \end{cases} \tag{6.13}$$

The feed-forward control u_{ff} is computed offline, meaning before the operation; and the feedback control u_{mpc} is computed online, during the operation. Hence, we may spend more computational time on finding u_{ff} than u_{mpc} because we wish to achieve an accurate heating-up. In contrast to that the feedback signals need to be computed quickly[3] to guarantee a stable closed-loop behavior. We portray the two-degrees-of-freedom control approach in Fig. 6.7.

[2] The naming indices of u_{ff} and u_{mpc} refer to "feed-forward" and "model predictive control".

[3] As the heat conducts slowly from the actuators to the sensors we may allow a "long" time to compute the feedback control. We discuss this in Chapter 8.

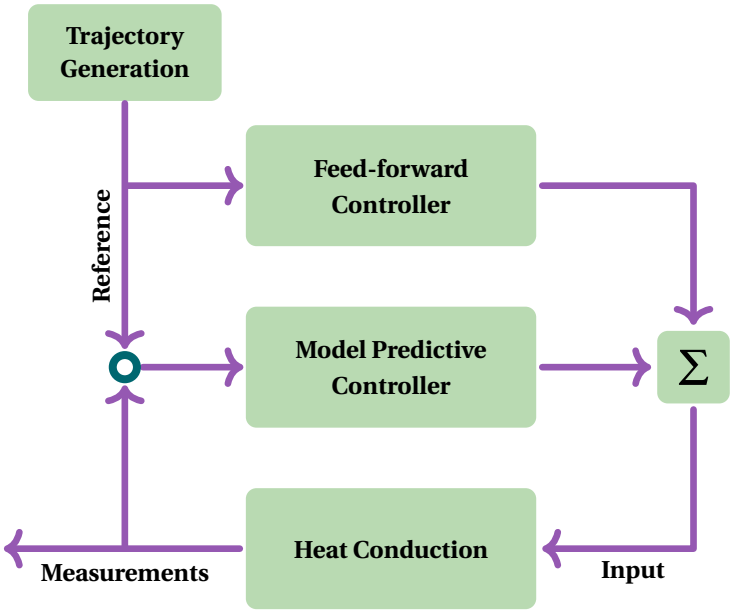


Figure 6.7: Scheme of a two degrees-of-control approach. The trajectory generator computes a reference signal, which is has to be followed by temperature measurements. The input signal of the feed-forward control is computed offline, before the operation of the heating-up procedure. The input signal from the model predictive control is computed online, during the thermal treatment of the object.

Feed-forward Control

The output signals have to follow a predefined transition during the heating-up phase. This transition is part of a reference function

$$r : [0, T_{final}] \rightarrow \mathbb{R}_{\geq},$$

which need to be tracked by the output signals as $y(t) \approx r(t)$ during the complete operation time. We allow only positive, non-decreasing, reference signals because we only discuss heating-up procedures. In the beginning of the transition, the reference and the output signals have to coincide approximately as

$$r(0) \approx \ y(0) = C\Theta(0).$$

The reference function has to approach a desired temperature $\Theta_d \in \mathbb{R}^{N_y}$ with $\Theta_{d,i} > y_i(0)$ for all $i \in \mathscr{S}_y$ in the end of the transition as

$$\lim_{t \rightarrow T_{ff}} r(t) \approx \Theta_d.$$

During the stabilization time $t \in [T_{ff}, T_{final})$, we claim that $r(t) \approx \Theta_d$. The transition from the initial output measurements towards the desired temperature is visualized in Fig. 6.8. We wish to find an input signal $u_{ff}(t)$, which drives the thermal dynamics properly in order to minimize error

$$e_{ff}(t) = r(t) - y(t) \tag{6.14}$$

during the heating-up time $t \in [0, T_{ff})$. If we assume any continuous and bounded input signal $u(t)$, then we find for each initial value $\Theta(0)$ a future temperature $\Theta(t)$ via integration of differential eq. (6.11a) and we yield the output measurements $y(t)$ with Eq. (6.11b). In the opposite way, we specify the output values with reference signal $y(t) \equiv r(t)$ and we search for an initial temperatures distribution $\Theta(0)$ and input signals $u(t)$, which lead to these output values. Though, in general we are not able find temperatures $\Theta(t)$ from output measurements $y(t)$ in an analytical way because

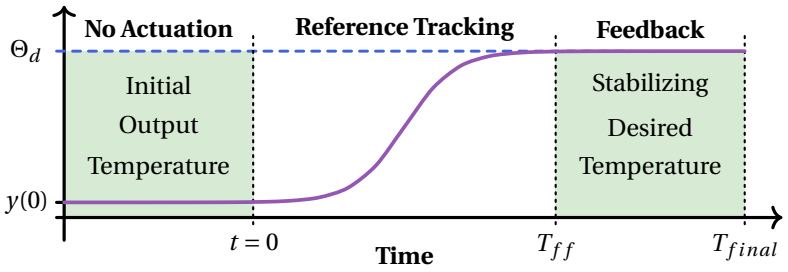


Figure 6.8: Transition from an initial operating temperature towards the desired temperature. A feed-forward control is applied to reach the reference tracking and a feedback control stabilizes the reached temperatures afterwards.

the mapping in Eq. (6.11b) is not unique. The number of output signals is less than the number of temperature states and the averaging process in Eq. (6.6) can not be reversed. Even if we know the temperatures $\Theta(t)$, then we might still not be able to find $(\Theta(0), u(t))$ analytically because the diffusion operation is ill-posed, as discussed for the linear case in Section 4.2. We see that solving the forward problem (known input, find output) is significantly easier than solving the inverse problem (known output, find input). Solving this inverse problem is not impossible: we firstly introduce an analytical approach and discuss secondly a numerical technique.

For some specific scenarios we can approximate an analytical input signal if the system is *differentially flat*. A finite dimensional nonlinear system

$$\dot{z}(t) = f(z(t), u(t)) \quad , \quad y(t) = g(z(t))$$

with states $z(t)$, input $u(t)$ and output $y(t)$ is called differentially flat if the number of input and output signals coincide $N_u = N_y$ and we have a flat output signal[4] $y(t)$ such that we find the mappings Ψ_z and Ψ_u to obtain the states and input signals via the derivatives of $y(t)$ as

$$z(t) = \Psi_z\left(y(t), \dot{y}(t), \ldots, y^{(n-1)}(t)\right), \tag{6.15a}$$
$$u(t) = \Psi_u\left(y(t), \dot{y}(t), \ldots, y^{(n)}(t)\right). \tag{6.15b}$$

[4] The flat output signal does not have to be a real measurement.

The differentiation order n in Eq. (6.15) corresponds to the system dimension, $z : [0, T_{ff}] \to \mathbb{R}^n$. This control approach was initially proposed in [108] and later extended for the heat equation and other partial differential equations in [46]. In case of PDE, we need (theoretically) an infinite number of derivatives $y^{(n)}(t)$ to yield the states and input signals. As it is difficult to apply the flatness-based control directly on nonlinear PDE, a common solution is to approximate the nonlinear PDE and apply the flatness-based control on the large-scale system of nonlinear ordinary differential equations. This approach is discussed in the articles [109, 110, 116] and an detailed analysis is described in the doctoral thesis [111].

A comprehensive study of flatness-based control techniques is presented in the works [112, 113] and for detailed discussions on PDE flatness-based control, we refer to the books [12], [114, p. 164] and [115, Ch. 6 - 8].

The flatness-based control design is not the only way to find open-loop input signals. If an analytical approach might not be applicable, e.g. we have more an unequal number of input and output signals, then the control input can also be computed numerically with optimization-based techniques. In the optimal control design, we seek for a proper input sig-

nal u^*, which minimizes an objective function J as

$$u^*(t) = \arg\min_u J(e,u).$$

The main idea of the optimal control approach is to solve the forward and inverse problem iteratively. In the forward pass we apply the input signals and we evaluate the objective function to check if the input signals lead to useful output measurements. In the inverse pass we vary the input signals depending on the result of the objective function. We need to distinguish again between the finite (ODE) and infinite dimensional (PDE)[5] situation. In the finite dimensional case, we sample the system dynamics in time to yield a set of difference equations and we search iteratively for optimal input signals with common numerical algorithms. In the infinite dimensional case, we can either initially discretize the PDE to yield a large scale finite dimensional system, or we optimize in a first step and discretize the optimal system afterwards. The optimal control of finite dimensional systems is rather fundamental and may also be carried out by non-experts, but the PDE-constraint optimization requires strong knowledge in functional analysis, PDE theory and related fields. Furthermore, the formulation of the optimal control problem for PDE is tailored for the specific problem setup (geometry, boundary conditions, configuration of actuators and sensors, etc.) and changing the problem requires a reformulation of the optimal control problem. Beside these issues, the standard optimal control design for ODE and PDE computationally expensive because we need to discretize the geometry and sample the time domain to gain for each time step the optimal input value $u(t_n)$. In this manner, the number of parameters depends on the number of actuators N_u multiplied by the number of time steps. We refer to the book [117, p. 95] for an introduction to optimal control of the heat equation. An analysis and numerical evaluation of the optimal control design for quasilinear PDE, like the quasilinear heat equation in Definition 2.1, is described in the doctoral thesis [56]. We emphasize that this technique has also been implemented successfully in many research applications, e.g. laser welding with a quasilinear heat conduction model in [8].

[5] The optimization with PDE is also called PDE-constrained optimization.

In Chapter 7, we unify the ideas of flatness-based and optimal control to derive an optimization-based control approach. Firstly, we design input signals with the flatness-based approach for simplified (linear) models. Secondly, we approximate the flatness-based input signal by a parameterized function $u_{ff}(t;p)$ and we optimize the parameters for the original (complex) model. In other words, we create prototype input signals for the linear heat conduction model with the flatness-based control and adjust them for real, nonlinear scenarios with optimization techniques. The input signal $u_{ff}(t;p)$ has only three parameters, and so we have noticeable reduced computational costs in comparison to a fully sampled input signal. We improve this optimization-based control additionally by including thermal energy estimates. Our proposed procedure is illustrated in Fig. 6.9, and these concepts were originally introduced in the articles [39, 40].

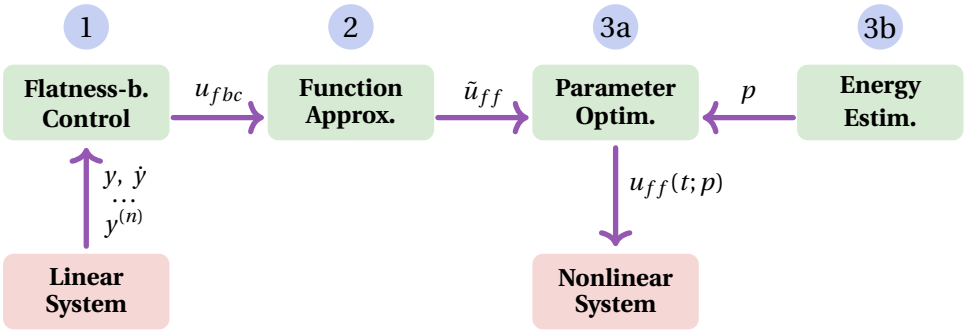


Figure 6.9: Scheme of derivation of the feed-forward control signal. In step 1 we compute the flatness-based control u_{fbc} for the linear system. We approximate u_{fbc} in the 2. step to construct the prototype $\tilde{u}_{ff}$. We optimize the parameters p in step 3 and include additional information about the estimate of the thermal energy. Finally, we apply the computed feed-forward control signal $u_{ff}(t;p)$.

Feedback Control

After the heating-up phase $t > T_{ff}$, we activate the feedback control to keep the measurements $y(t)$ close to the desired temperatures Θ_d. We need to design a closed-loop controller to minimize the error

$$e_{fb}(t) = \Theta_d - y(t)$$

for $t \in [T_{ff}, T_{final})$. The most established controller type in the industrial automation is the PID control. It computes the input signal as

$$u_{pid}(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t)$$

with controller coefficients $(K_p,\ K_i,\ K_d) \in \mathbb{R}^3$, which amplify the proportional, integral and differential error. As we only have these three parameters, PID control might be simple to design and implement. Though, it may not perform well for thermal dynamics because the heat needs some time to propagate from the actuator to the sensor and the PID controller can not predict this behavior. Beside this issue the standard PID control requires the same number of input and output signals as $N_u = N_y$.

An alternative approach is the model predictive control (MPC) design, which uses an internal model of the system to predict the future system behavior. The MPC algorithm solves in each time step an optimal control problem to minimize the error between the prediction and the reference. For a general introduction to model predictive control we refer to the books [119, 120]. In the doctoral thesis [121], the author proposes a model predictive control design for PDE, like the heat equation and the wave equation. One remarkable issue of MPC for PDE are the high computational costs. Recent research focuses on the reduction of these costs with model order reduction, see [122]. In Section 8.2, we design a model predictive control approach, which is tuned with the knowledge of thermal emissions. Due to a proper choice of control parameters, we are able to stabilize the measurement values close to the desired temperatures.

7

Open-loop Control Design

The main task of the control system is to heat up the object to reach a desired temperature. In particular, we wish to steer the thermal dynamics from an initial towards the final uniform temperature distribution along a specified reference trajectory. We approach this goal in two major steps: prototyping the input signal with flatness-based control for a simplified model and transferring the prototype signal to a parametrized function, which is optimized to fit the specific needs of the full nonlinear system.

First of all, we introduce in Sections 7.1 and 7.2 and the flatness-based control design, which offers a well established set of mathematical methods to design open-loop control algorithms for ordinary and partial differential equations. Here, we assume only constant material properties and neglect thermal emissions to yield a linear system. We restrict the discussion in Section 7.1 to the one-dim. heat equation because the flatness-based control design for higher-dimensional geometrical domains leads to complex formulations, and is out of scope of this thesis, see the book [115, p. 127, 143] and article [118]. We transfer these ideas to the spatially approximated heat conduction problems in Section 7.2 and we find similar results for the one-dim. case as in Section 7.1. Furthermore, we evaluate the flatness-based control for a two-dim. geometry with multiple actuators and sensors, but we face the issue that this control scenario has only limited relevance for our proposes. In Section 7.1, we explain the choice and parametrization of the reference function and how it influences the input signal and the resulting thermal dynamics. The introduced reference signal is very smooth and fulfills certain criteria, which are necessary for an analytically correct reference tracking of PDE in general. The disadvantage of this reference function is its complexity and finding the derivatives is computationally costly. Hence, we propose in Section 7.3 further approaches to find simple, suitable and computationally cheap reference functions.

In the second part of this chapter, in Section 7.4, we introduce a parameterized bell-shaped function to approximate the flatness-based signal. The parameters of this input signal are optimized in subsequent steps to control the original nonlinear problem properly. Accordingly, we describe the influence of our input signal parameters and we explain stepwise their optimization. We extend the optimization-based design with energy considerations in Section 7.5 because the temperatures are only determined

by the ratio of supplied and emitted thermal energy. This concept of energy balance simplifies the parameter optimization significantly as we neglect the temporal evolution of the heat dynamics and tune the integral of our input signal. Finally, in Section 7.6, we wrap up all the presented ideas of feed-forward control and we exemplify them on a two-dim. heat conduction model with anisotropic and temperature-dependent thermal conductivity and nonlinear boundary conditions.

7.1 Flatness-based Control of the Linear Heat Equation

The flatness-based control design for partial differential equations, in particular the heat equation, was initially described in the article [46] and gradually extended for further PDE, see [12], and complex scenarios as in [115, p. 133, 143]. Here, we only assume the simplest applicable version of the heat conduction phenomena: the linear one-dim. heat equation without thermal emissions. In accordance with [38,46], we derive an input signal as a power series with analytical tools. Furthermore, we design the reference signal and discuss how its derivatives are used to compute the input signal. For this purpose, we apply the ideas of article [36] to compute the reference derivatives. In the end of this section, we simulate the one-dim. heat equation with found input signal and we discuss how the final transition time T_{ff} affects the input signal and in consequence the heat dynamics. This issue is evaluated for various scenarios in [38].

We return to the continuous formulation of the linear heat equation

$$\frac{d}{dt}\vartheta(t,x) = \alpha\frac{\partial^2}{\partial x^2}\vartheta(t,x) \tag{7.1}$$

with diffusivity[1] $\alpha = \frac{\lambda_1}{c\,\rho}$ and $(t,x) \in (0,T_{ff}) \times (0,L)$ as mentioned in Eq. (2.21). We specify an actuation on the left side (B_W) and a thermal insulation on the right side (B_E) as

$$u(t) = -\lambda\frac{\partial}{\partial x}\vartheta(t,x) \quad \text{for } x \in B_W\,, \tag{7.2a}$$

$$0 = \lambda\frac{\partial}{\partial x}\vartheta(t,x) \quad \text{for } x \in B_E \tag{7.2b}$$

and we measure the temperature on boundary B_E as

$$y(t) = \vartheta(t,L). \tag{7.3}$$

This heat conduction model is strongly simplified, but it is the prototype system for the flatness-based control of PDE, see [46]. According to the literature [3, p. 232] and [123, p. 111], the solution of ordinary and partial differential equations may be formulated in terms of a power series.[2] In case of the linear heat equation, we define the power series

$$\tilde{\vartheta}(t,x) := \sum_{i=0}^{\infty}\hat{\vartheta}_i(t)\frac{(L-x)^i}{i!}. \tag{7.4}$$

We find the derivatives of $\tilde{\vartheta}$ with respect to position x as

$$\frac{\partial}{\partial x}\tilde{\vartheta}(t,x) = -\sum_{i=0}^{\infty}\hat{\vartheta}_{i+1}(t)\frac{(L-x)^i}{i!} \quad \text{and} \tag{7.5}$$

$$\frac{\partial^2}{\partial x^2}\tilde{\vartheta}(t,x) = \sum_{i=0}^{\infty}\hat{\vartheta}_{i+2}(t)\frac{(L-x)^i}{i!} \tag{7.6}$$

[1] As we only consider the one-dim. case, we drop the index, $\alpha = \alpha_1$.

[2] This solution technique is also known as Frobenius method, see [124]. Ferdinand Georg Frobenius (∗1849, †1917) [125] extended previous ideas by Lazarus Immanuel Fuchs (∗1833, †1902) [126] and Karl Weierstraß(∗1815, †1897) [127].

and we note the derivative in time as

$$\frac{\partial}{\partial t}\tilde{\vartheta}(t,x) = \sum_{i=0}^{\infty} \frac{\partial}{\partial t}\hat{\vartheta}_i(t)\frac{(L-x)^i}{i!}. \tag{7.7}$$

In the heat equation (7.1), we set $\vartheta(t,x) \equiv \tilde{\vartheta}(t,x)$ and so we yield with Eq. (7.6,7.7) the identity

$$\sum_{i=0}^{\infty} \frac{d}{dt}\hat{\vartheta}_i(t)\frac{(L-x)^i}{i!} \;=\; \alpha \sum_{i=0}^{\infty} \hat{\vartheta}_{i+2}(t)\frac{(L-x)^i}{i!}.$$

We compare the left and right-hand side in the previous equation and we take out the identity

$$\frac{d}{dt}\hat{\vartheta}_i(t) = \alpha\, \hat{\vartheta}_{i+2}(t). \tag{7.8}$$

In the subsequent steps, we formulate $\hat{\vartheta}_i$ in terms of the output signal $y(t)$ and its derivatives to find the mappings Ψ_x and Ψ_u, see Eq. (6.15). We note the output signal in Eq. (7.3) as

$$y(t) = \vartheta(t,L) = \sum_{i=0}^{\infty} \hat{\vartheta}_i(t)\frac{0^i}{i!} = \hat{\vartheta}_0(t)$$

and we deduce from Eq. (7.8) the identity

$$\frac{d^i}{dt^i}y(t) = \frac{d^i}{dt^i}\hat{\vartheta}_0(t) = \alpha^i \hat{\vartheta}_{2i}(t) \quad \text{for } i > 0.$$

The boundary condition on B_E, Eq. (7.2b), leads us to expression

$$\lambda\frac{\partial}{\partial x}\vartheta(t,L) = -\lambda\sum_{i=0}^{\infty} \hat{\vartheta}_{i+1}(t)\frac{0^i}{i!} = -\lambda\hat{\vartheta}_1(t) = 0$$

and we continue this fact with Eq. (7.8) to yield

$$\frac{d^i}{dt^i}\hat{\vartheta}_1(t) = \alpha^i\; \hat{\vartheta}_{2i+1} \equiv 0.$$

We summarize the previous findings and we split the identity (7.8) into both sequences

$$\hat{\vartheta}_{2i}(t) = \;\alpha^{-i} y^{(i)}(t) \quad \text{and} \quad \hat{\vartheta}_{2i+1}(t) = \;0. \tag{7.9}$$

We reformulate Eq. (7.4) as

$$\tilde{\vartheta}(t,x) = \sum_{i=0}^{\infty} \begin{cases} \frac{y^{\left(\frac{i}{2}\right)}(t)}{\alpha^{\frac{i}{2}}}\frac{[L-x]^i}{i!} & \text{if } i \text{ is even,} \\ 0 & \text{if } i \text{ is odd} \end{cases}$$

and we set $i \to 2i$ to derive the output to states mapping in Eq. (6.15a) as

$$\tilde{\vartheta}(t,x) = \sum_{i=0}^{\infty} \frac{y^{(i)}(t)}{\alpha^i}\frac{[L-x]^{2i}}{2i!}. \tag{7.10}$$

We formulate the actuation on boundary B_W, Eq (7.2a), in terms of $\hat{\vartheta}_{2i}(t)$ with Eq. (7.5) as

$$u(t) = \;-\lambda\frac{\partial}{\partial x}\vartheta(t,0) = \lambda\sum_{i=0}^{\infty} \hat{\vartheta}_{i+1}(t)\frac{L^i}{i!}$$

and we yield in an analog way to Eq. (7.10) with $i \rightarrow 2i+1$ the output to input mapping in Eq. (6.15b) as

$$u(t) = \lambda \sum_{i=0}^{\infty} \frac{y^{(i+1)}(t)}{\alpha^{i+1}} \frac{L^{2i+1}}{(2i+1)!}. \tag{7.11}$$

The mappings in Eq. (7.10) and (7.11) enable us to compute the temperature at any position $x \in \Omega$ and the input signal if we consider sufficiently enough derivatives of $y(t)$. Hence, we can find the inverse system of the heat equation, but we need to estimate the number of necessary series sequences and this situation is discussed in article [38].

Reference function of Gevrey class

We wish to steer the output of heat equation (7.3) along a predefined reference trajectory

$$r(t) = r_0 + \Delta r \, \psi(t,p)$$

with $r_0 = y(0)$, $\Delta r = \Theta_d - r_0 > 0$ and transition ψ. Hence, we identify the output y and all of its derivatives $\frac{d^i}{dt^i} y(t)$ in Eq. (7.11) by reference function $r(t)$ and $\frac{d^i}{dt^i} r(t)$. The transition function[3] has to be zero at $t = 0$ and one at $t = T_{ff}$ and need to be infinite-times continuously differentiable in the time because we have theoretically an infinite number of derivatives $\frac{d^i}{dt^i} r(t)$. Additionally, all derivatives of the transition need to be close to zero, e.g.

$$\lim_{t \rightarrow \{0, T_{ff}\}} \frac{d^i}{dt^i} \psi(t, \cdot) \approx 0$$

for $i \in \mathbb{N}_{>0}$. In the initial article on PDE flatness-based control [46], the authors propose the transitions function

$$\psi(t,p) := \begin{cases} 0 & \text{if } t \leq 0, \\ 1 & \text{if } t \geq T_{ff}, \\ \frac{\int_0^t \omega\left(\frac{\tau}{T_{ff}}, p\right) d\tau}{\int_0^{T_{ff}} \omega\left(\frac{\tau}{T_{ff}}, p\right) d\tau} & \text{if } t \in (0, T_{ff}), \end{cases} \tag{7.12}$$

which contains the integral of the bump function

$$\omega(t,p) := \begin{cases} 0 & t \notin [0,1], \\ \exp\left(-[t-t^2]^{-p}\right) & t \in (0,1). \end{cases} \tag{7.13}$$

The steepness of the transition is specified with parameter p, which need to be set such that condition $1 + \frac{1}{p} < 2$ or equally $p > 1$ holds. The smooth transition and bump function goes over to a sharp behavior if we increase parameter p as visualized in Fig. 7.1. Transition ψ and bump function ω are functions of the Gevrey class,[4] we refer for a detailed analysis to article [46]. We need to differentiate the reference signal $r(t)$ and likewise also transition $\psi(t,p)$ to compute the input $u(t)$. We find the derivatives of $\psi(t,p)$ with the scaled bump function as

$$\frac{d}{dt}\psi(t,p) = \frac{\omega(t,p)}{\hat{\omega}(p)} \quad \text{and} \quad \frac{d^i}{dt^i}\psi(t,p) = \frac{\frac{d^i}{dt^i}\omega(t,p)}{\hat{\omega}(p)}$$

[3] In the subsequent sections, we consider weaker conditions, e.g. $\psi(0,p) \approx 0$ instead of $\psi(0,p) = 0$.

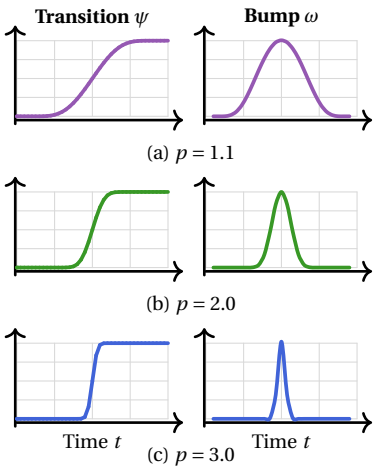


Figure 7.1: Transition ψ and bump function ω for parameter $p \in \{1.1, 2, 3\}$. An increasing p leads to a steep transition and sharp bump function.

[4] The Gevrey class is introduced by Maurice Gevrey in [128], and further information is provided online [129].

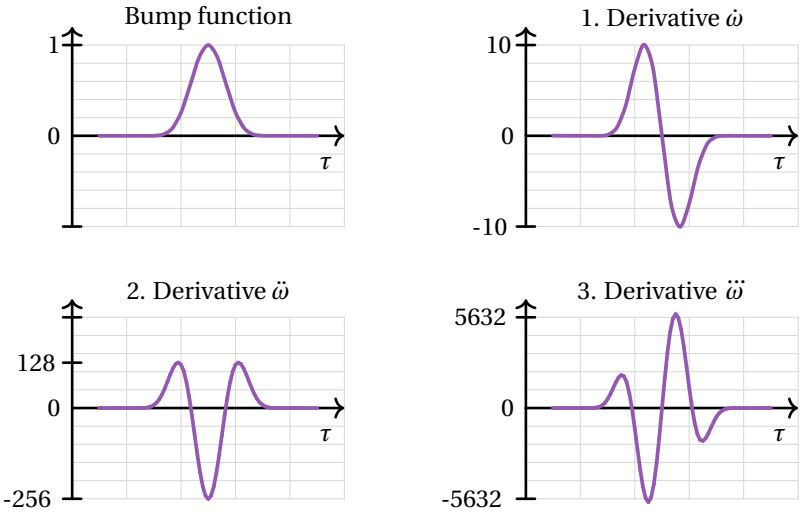


Figure 7.2: Bump function $\omega(t,p)$ and its first derivatives for $p = 2$. The maximum value of the derivatives increase dramatically by the order of differentiation.

with integral $\hat{\omega}(p) = \int_0^{T_{ff}} \omega\left(\frac{\tau}{T_{ff}}, p\right) d\tau$. So, the i-th derivative of transition $\psi(t,p)$ corresponds to the $(i-1)$-th derivative of $\omega(t,p)$. Bump function ω in Eq. (7.13) is a function composition as

$$\omega(t,p) = \exp\big(f(g(t),p)\big) \qquad \text{with} \qquad (7.14a)$$

$$f(z,p) = -z^{-p} \qquad \text{and} \qquad (7.14b)$$

$$g(t) = t - t^2. \qquad (7.14c)$$

We evaluate the derivatives of Eq. (7.14a) via the chain rule and we obtain an expression

$$\frac{d^i}{dt^i}\omega(t,p) = \frac{\sum_{n=0}^{N_{num}} b_n t^n}{\sum_{n=0}^{N_{den}} a_n t^n} \exp(f(t,p))$$

with coefficients $a_n, b_n \in \mathbb{R}$ and the order of the numerator polynomial is smaller than the denominator, $N_{num} < N_{den}$. The order of both polynomials increase by the order of differentiation. We evaluate the derivative $\frac{d^i}{dt^i}\omega(t,p)$ for $t \in (0,1)$ and so we yield large values because of the rational function term. In Fig. 7.2, we portray the first derivatives of the bump function for $p = 2$ and we see that the maximum value increases significantly by the order of differentiation. This fact is a problem for the computation of the input signal because the sequence elements shall not increase to infinity and we need to terminate the power series in Eq. (7.11) to yield a suitable approximation

$$u_{N_{iter}}(t) := \lambda \sum_{i=0}^{N_{iter}} \frac{r^{(i+1)}(t)}{\alpha^{i+1}} \frac{L^{2i+1}}{(2i+1)!} \qquad (7.15)$$

with $\|u_{N_{iter}}(t) - u(t)\| < \varepsilon$ for a small $\varepsilon > 0$. We see in Eq. (7.15) that the series elements contain the diffusivity α and the choice of the control parameters, time T_{ff} and steepness factor p. In article [38], the authors discuss the impact of the material parameters and control parameters on the resulting input signal.

Computation of the Derivatives

We need to compute several, possibly a large number of derivatives of bump function $\omega(t,p)$ to find an exact approximation of input $u(t)$ as in Eq. (7.15). However, the computation of derivatives of $\omega(t,p)$ is not trivial because it is very smooth and it is a function composition, see Eq. (7.14). A numerical evaluation is not applicable because high-order derivatives tend to strongly oscillating behavior, see also Fig. 7.2, and a manual calculation is too error prone. Hence, a symbolic computation, e.g. as in [130] with MATLAB, or an algorithmic approach with recursion formulas as in [131] may solve this task. However, these approaches are tailored for the bump function $\omega(t,p)$ as noted in Eq. (7.13). As an alternative, the authors present in article [36] a method to compute derivatives for generic function compositions as $f \circ g(t)$ with $g : \mathbb{R} \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$. For this purpose, the function composition is evaluated with Faà di Bruno's formula and Bell polynomials. In this thesis, we compute the derivatives with Faà di Bruno's formula, which is implemented in the JULIA library *BellBruno.jl*, see [45].

Example: PDE Flatness-based Control

We demonstrate the flatness-based control for the linear heat equation without thermal emissions. We consider a one-dim. model of a rod with length $L = 0.1$ and material properties $\lambda = 50$, $\rho = 8000$ and $c = 400$. The measured temperature $y(t) = \vartheta(t,L)$ shall follow the reference signal

$$r(t) = 300 + 100\,\psi(t,p) \tag{7.16}$$

with steepness parameter $p = 2$, see Fig. 7.1 (b). The input signal in Eq. (7.15) contains coefficients related to the geometrical object and the reference derivatives as

$$u_{N_{iter}}(t) = \sum_{i=0}^{N_{iter}} \eta_i \frac{d^{i+1}}{dt^{i+1}} r(t)$$

with sequence elements

$$\eta_i := \frac{\lambda\, L^{2i+1}}{\alpha^{i+1}\,(2i+1)!}.$$

Here, we compute the input signal for $N_{iter} = 20$ iterations, we evaluate the sequence η_i numerically and so we yield very high values, e.g.

$$\max_{i \in \{0,1,\ldots,N_{iter}\}} \eta_i \approx 1.95 \cdot 10^{12}.$$

The data of sequence η_i in Fig. 7.3 is displayed in semi-logarithmic scale and we see that η_i reaches its maximum for index $i = 12$ and stays on a high level afterwards. Thus, we need to choose suitable control parameters (steepness p and time T_{ff}) to yield small higher-order reference derivatives and to avoid (strong) oscillations, as in Fig. 7.2, in the computed input signal. As we already fixed the steepness $p = 2$, we vary the final time of the feed-forward control $T_{ff} \in \{400, 1200, 3000\}$ seconds and we evaluate whether the input signal and resulting temperature evolution suffice our constraints. We remark that the input shall not drop below zero because we only control the heating and not the cooling. Moreover, the temperatures shall not drop below its initial value.

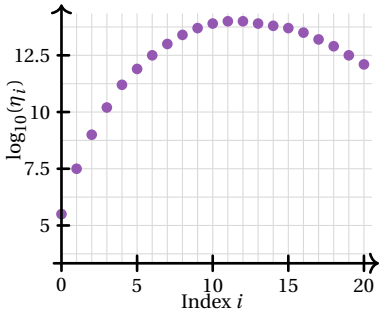
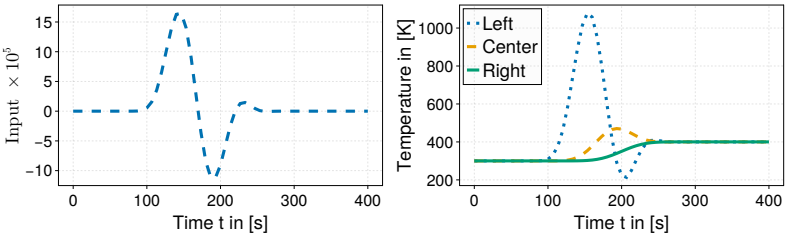
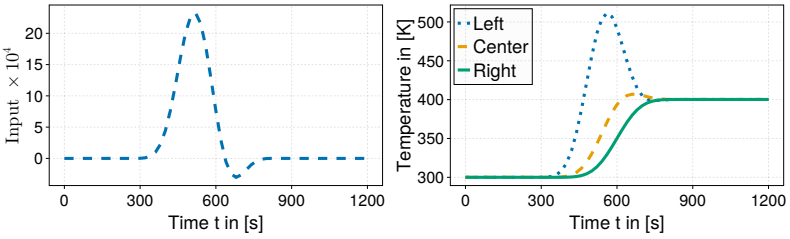


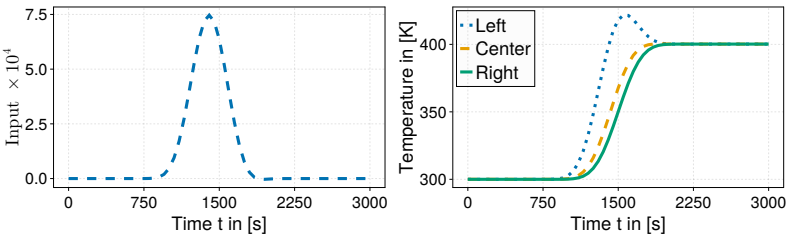
Figure 7.3: Logarithmic scaling of sequence elements η_i, which amplify the reference derivatives in the PDE flatness-based control.



(a) $T_{ff} = 400$ s



(b) $T_{ff} = 1200$ s



(c) $T_{ff} = 3000$ s

Figure 7.4: Computation of the flatness-based input signal and the resulting temperature evolution. The graphs on the right side show the temperatures of the rod on the left side: $x = 0$ m, in the center: $x = \frac{L}{2} = 0.05$ m and on the right side: $x = L = 0.1$ m. The temperature on the right side is the output and it follows the predefined reference trajectory. Though, the input signal for $T_{ff} \in \{400, 1200\}$ seconds is not admissible because it drops below zero and produces temperatures below the initial temperature in (a).

We visualize the results of our simulations in Fig. 7.4; see also article [38] for similar results. In the first case, we set $T_{ff} = 400$ seconds and we yield an input signal in Fig. 7.4 (a) with strong oscillations and a very high magnitude

$$\max_{t \in (0, T_{ff})} u(t) \approx 1.5 \cdot 10^6.$$

This intensive input signal causes very high and low temperatures, e.g. more than 1000 Kelvin and almost 200 Kelvin, on the left side of the rod, $x = 0$. This situation might be physically realizable but is not practical for ordinary industrial applications. Hence, we have to exclude this parameter setup for further applications.

In the second scenario, $T_{ff} = 1200$ seconds, the input signal reaches small negative values (after $t \approx 600$ seconds) in Fig. 7.4 (b), but all temperatures are above the initial temperature. As we only have small negative input values, we might apply a limitation of the input signal with $\tilde{u}(t) = \max(u(t), 0)$ and still yield a reasonable temperature evolution.

Finally, for $T_{ff} = 3000$ seconds, we compute an almost Gaussian-shaped input function, which produces a suitable temperature evolution because the overshoot is much smaller compared to the other scenarios. So, the first derivative $\frac{d}{dt}\psi$ has the main impact here on the shape of the input signal, see Fig. 7.2. In Section 7.4, we approximate this flatness-based input signal with a parameterized function.

7.2 Flatness-based Control of the Approximated System

The flatness-based control design was initially proposed for finite dimensional nonlinear systems in the article [108]. The main idea to find the input signal u is the differentiation of the output signal y. We firstly explain how to find the input signal in general and afterwards we distinguish between the one- and the multi-dimensional scenario. We consider the approximated linear heat conduction problem

$$\begin{aligned}\frac{d}{dt}\Theta(t) &= A\,\Theta(t) + B\,u(t)\\ y(t) &= C\,\Theta(t)\end{aligned}$$

with matrices $A \in \mathbb{R}^{N_c \times N_c}$, $B \in \mathbb{R}^{N_c \times N_u}$ and $C \in \mathbb{R}^{N_y \times N_c}$, see Definition 6.1. We differentiate the output y for N_c times to find the mappings in Eq. (6.15) and so we obtain

$$\frac{d^i}{dt^i} y(t) = CA^i\, x(t) + CA^{i-1} B\, u(t) \tag{7.17}$$

for $i \in \{1,\dots,N_c\}$. If the term $CA^{i-1}B$ vanishes for $i \in \{1,\dots,N_c-1\}$, then we can note the state mapping ψ_x. In this case, we summarize y and its $N_c - 1$ derivatives as

$$z(t) := \begin{pmatrix} y(t) \\ \dot{y}(t) \\ \vdots \\ y^{(N_c-1)}(t) \end{pmatrix} = \begin{pmatrix} C \\ C\,A \\ \vdots \\ CA^{N_c-1} \end{pmatrix} \Theta(t) = T_y\,\Theta(t)$$

with transformation matrix $T_y := \left[C, C\,A, \dots, C\,A^{N_c-1}\right]^\top$ and we find the state mapping ψ_x as

$$\Theta(t) = T_y^{-1} z(t) = \psi_x(y, \dot{y}, \dots, y^{(N_c-1)}). \tag{7.18}$$

As matrix T_y needs to be invertible the number of its rows and columns must coincide. In the second part of this section, we discuss this situation for systems with multiple input and output signals where we have more rows than columns and we need to decrease the size of T_y. We find the highest-order derivative of the output as

$$\frac{d^{N_c}}{dt^n} y(t) = CA^{N_c}\Theta(t) + CA^{N_c-1}Bu(t)$$

with $CA^{N_c-1}B \neq 0$. Accordingly, we yield the input signal as

$$u(t) = \left[CA^{N_c-1}B\right]^{-1}\left(\frac{d^n}{dt^n} y(t) - CA^{N_c}\Theta(t)\right). \tag{7.19}$$

We identify Θ in Eq. (7.19) by the right-hand side of Eq. (7.18) and we obtain the input mapping

$$\begin{aligned} u(t) &= \left[CA^{N_c-1}B\right]^{-1}\left(\frac{d^{N_c}}{dt^{N_c}} y(t) - CA^{N_c} T_y^{-1} z(t)\right)\\ &= M_u \begin{pmatrix} y(t) \\ \dot{y}(t) \\ \vdots \\ y^{(N_c)}(t) \end{pmatrix} = \psi_u(y, \dot{y}, \dots, y^{(n)}) \end{aligned} \tag{7.20}$$

with matrix

$$M_u := \left[CA^{N_c-1}B\right]^{-1}\left[-CA^{N_c}T_y^{-1}|I_{N_y}\right]. \tag{7.21}$$

I_{N_y} denotes the identity matrix of size $N_y \times N_y$. As we wish to compute an input signal depending on the reference signal, we need to identify the output $y(t)$ by the reference signal $r(t)$ in Eq. (7.20) and we note

$$u(t) = M_u \begin{pmatrix} r(t) \\ \dot{r}(t) \\ \vdots \\ r^{(N_c)}(t) \end{pmatrix}. \tag{7.22}$$

Subsequently, we discuss the peculiarities of flatness-based control for the one- and two-/three-dimensional case.

One-dimensional Scenario

If we assume the one-dimensional rod as geometric object then we have one actuator and one sensor on opposite sides. We set the actuation on boundary B_W and temperature measurement on B_E. These positions correspond to the first x_1 (actuation) and last grid node $x_{N_c} = x_{N_j}$ (measurement). Thus, we note the input and output vectors as

$$B = \left[\frac{b}{\Delta x_1\, c\rho}, 0, \ldots, 0\right]^\top \quad \text{and} \quad C = [0, \ldots, 0, \tilde{c}]$$

with $b \in [0,1]$ and $\tilde{c} \in [0,1]$. We differentiate output y and we build iteratively a relation between the output and all of its states, as depicted in Fig. 7.5. We have system matrix $A = \frac{\alpha_1}{\Delta x_1^2} D_1$ with diffusivity $\alpha_1 = \frac{\lambda_1}{c\,\rho}$ and diffusion matrix D_1, see Eq. (3.36). We calculate M_u in Eq. (7.21) with the N_c-th power of A, but this matrix contains floating point values because $A = \frac{\alpha_1}{\Delta x_1^2} D_1$. Hence, we face numerical inaccuracies in the finding of A^i and this issue may have a crucial impact on the computation of T_y^{-1}, see Eq. (7.18). We solve this problem as we split floating point and integer values as

$$A^i = \left[\frac{\alpha_1}{\Delta x_1^2}\right]^i D_1^i \quad \text{and} \quad CA^i = \left[\frac{\alpha_1}{\Delta x_1^2}\right]^i CD_1^i.$$

Accordingly, we split the transformation matrix as

$$T_y = \operatorname{diag}\left(1, \frac{\alpha_1}{\Delta x_1^2}, \ldots, \left[\frac{\alpha_1}{\Delta x_1^2}\right]^{[N_c-1]}\right) \begin{pmatrix} C \\ C\,D_1 \\ \vdots \\ CD_1^{N_c-1} \end{pmatrix} =: T_{y,1}\; T_{y,2}.$$

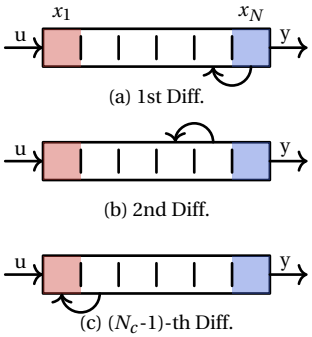


Figure 7.5: Differentiation of the output y for the one-dim. rod. The i-th derivative of $y(t)$ relates to the temperature node of index $N_c - i$.

We see that $T_{y,1}$ is a diagonal matrix and hence we simply calculate the inverse of T_y as

$$\begin{aligned} T_y^{-1} &= \left[T_{y,1}\ T_{y,2}\right]^{-1} = T_{y,2}^{-1}\ T_{y,1}^{-1} \\ &= \begin{pmatrix} C \\ C\,D_1 \\ \vdots \\ CD_1^{N_c-1} \end{pmatrix}^{-1} \operatorname{diag}\left(1, \frac{\Delta x_1^2}{\alpha_1}, \ldots, \left[\frac{\Delta x_1^2}{\alpha_1}\right]^{[N_c-1]}\right) \end{aligned} \tag{7.23}$$

In case of the computation of $CA^{N_c-1}B$, we know that vectors C and B relate to the last and first grid node, respectively. Diffusion matrix D_1 has almost a Toeplitz form and so find the i-th power of D_1 with a vector of ones on the $(i+1)$-th subdiagonal, for example

$$D^2 = \begin{pmatrix} * & * & 1 & & & \\ * & * & * & 1 & & \\ 1 & * & * & * & 1 & \\ & 1 & * & * & * & 1 \\ & & 1 & * & * & * \\ & & & 1 & * & * \end{pmatrix} \quad \text{and} \quad D^{[N_c-1]} = \begin{pmatrix} * & * & \ldots & * & 1 \\ * & * & * & & * \\ \vdots & * & \ddots & * & \vdots \\ * & & * & * & * \\ 1 & * & \ldots & * & * \end{pmatrix}.$$

Hence, we calculate

$$(0,\ldots,0,1)\ D_1^{[N_c-1]} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 1.$$

We continue this idea to obtain $CD_1^{N_c-1}B = \frac{b\,\tilde{c}}{\Delta x_1\ c\rho}$ and we compute in a next step

$$\begin{aligned} CA^{N_c-1}B &= \left[\frac{\alpha_1}{\Delta x_1^2}\right]^{[N_c-1]} CD_1^{N_c-1}B \\ &= \left[\frac{\alpha_1}{\Delta x_1^2}\right]^{[N_c-1]} \frac{b\,\tilde{c}}{\Delta x_1\ c\rho}. \end{aligned} \tag{7.24}$$

Finally, we wish to note row vector M_u as in Eq. (7.21) explicitly including the previous considerations. For the first $N_c - 1$ entries, we consider the identity

$$CA^{N_c} = \left[\frac{\alpha_1}{\Delta x_1^2}\right]^{N_c} CD_1^{N_c}$$

to derive the expression

$$\begin{aligned} \left[CA^{N_c-1}B\right]^{-1} CA^{N_c} &= \left[\frac{\Delta x_1^2}{\alpha_1}\right]^{[N_c-1]} \frac{\Delta x_1\ c\rho}{b\,\tilde{c}} \left[\frac{\alpha_1}{\Delta x_1^2}\right]^{N_c} CD_1^{N_c} \\ &= \frac{\lambda_1}{\Delta x\ b\,\tilde{c}}\ C\,D_1^{N_c}. \end{aligned}$$

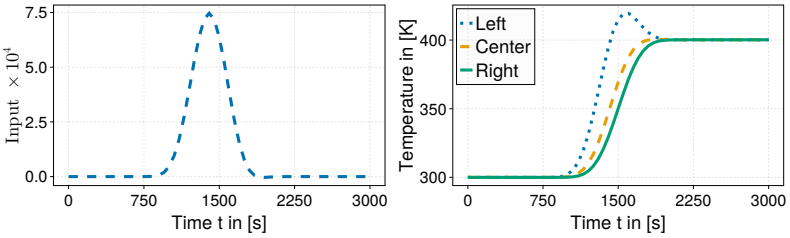


Figure 7.7: Computation of the flatness-based input signal and the resulting temperature evolution for the approximated one-dimensional heat equation. The data coincides with the results of the PDE flatness-based control in Fig. 7.4.

For the last entry of M_u, we have $[CA^{N_c-1}B]^{-1}$ by inverting the right-hand side of Eq. (7.24). We summarize all the previous findings and we note

$$\begin{aligned} M_u &= \left[-\left[CA^{N_c-1}B\right]^{-1} CA^{N_c} T_y^{-1} \middle| \left[CA^{N_c-1}B\right]^{-1} \right] \\ &= \left[-\frac{\lambda_1}{\Delta x\, b\, \tilde{c}}\, C\, D_1^{N_c}\, T_{y,2}^{-1} T_{y,1}^{-1} \middle| \left[\frac{\Delta x_1^2}{\alpha_1}\right]^{[N_c-1]} \frac{\Delta x_1\, c\rho}{b\, \tilde{c}} \right]. \end{aligned} \tag{7.25}$$

Additionally, we remark that the first entry of M_u has to be equally zero as $M_u\ (1,0,\ldots,0)^\top = 0$ because the input signal shall consist of the reference derivatives $\frac{d^i}{dt^i} r(t)$ only and not of the reference $r(t)$. We find the same situation in PDE flatness-based control in Eq. (7.11,7.15). This fact is based on the integrating behavior of the heat equation with non-insulated boundary conditions. We discussed this scenario in the example of Section 4.3, see also Eq. (4.60).

Example: Flatness-based Control in one Dimension

We consider the same heat conduction example from Section 7.1. We design an input signal for the same reference signal, see Eq. (7.16), with steepness $p = 2$ and final feed-forward time $T_{ff} = 3000$ seconds. We spatially approximate the rod with $N_c = 20$ grid nodes and we compute the input signal as in Eq. (7.22) with M_u as in Eq.(7.25). As we know that the first entry of M_u is zero as $M_u = [0, \tilde{m}_1, \tilde{m}_2, \ldots, \tilde{m}_{N_c}]$, we compute the input signal as

$$u(t) = \sum_{i=1}^{N_c} \tilde{m}_i \frac{d^i}{dt^i} r(t).$$

The values of $\tilde{m}_i$ describe the scaling of each reference derivative and they reach very large numbers, e.g. $\max_{i\in\{1,\ldots N_c\}} \tilde{m}_i \approx 2.5 \cdot 10^{13}$. We portray these values $\tilde{m}_i$ in Fig. 7.6 in semi-logarithmic scale. We compare $\tilde{m}_i$ with the sequence elements η_i from the example in Section 7.1, see Fig. 7.3, and we find similar values for the first indices, which correspond to the low order derivatives. We apply the input on our approximated heat conduction model and we visualize the input signal and the resulting temperatures in Fig. 7.7. Here, we notice that the similar values of $\tilde{m}_i$ and η_i correspond to similar input signals and thermal dynamics, see Fig. 7.4 (c).

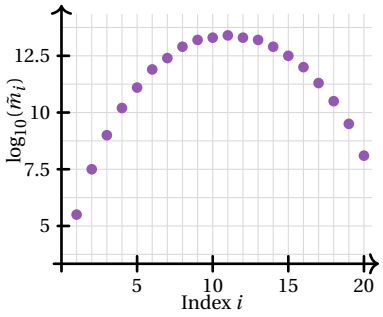


Figure 7.6: Logarithmic scaling of vector elements $\tilde{m}_i$, which amplify the reference derivatives for the flatness-based control of the one-dim. approximated heat equation.

Multiple Actuators and Sensors for Two and Three Dimensions

In the multi-dimensional scenario we consider multiple actuators and sensors, $N_u > 1$ and $N_y > 1$. The matrices B and C need to fulfill certain criteria to find the state and input mapping, ψ_x and ψ_u, with Eq. (7.17).

We know that the transformation matrix has to have full rank $T_y \in \mathbb{R}^{N_c \times N_c}$ to be invertible. However, we have N_y sensors and output matrix $C \in \mathbb{R}^{N_y \times N_c}$, and so we find the matrix

$$\begin{pmatrix} C \\ CA \\ \vdots \\ CA^{N_c-1} \end{pmatrix} \in \mathbb{R}^{N_y\ N_c \times N_c}$$

with more rows than columns. Thus, we need to reduce the number rows either by removing linear dependent ones or we consider only the first $\tilde{N}$ matrix blocks as

$$T_y = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{\tilde{N}-1} \end{pmatrix} \tag{7.26}$$

with $\tilde{N} = \frac{N_c}{N_y}$. In the latter case, we need to prove whether T_y still has full rank. This is the fact, if we have a bijective mapping between a subset of the temperature nodes and the output signals y. In an example, we assume

$$y(t) = C\,\Theta(t)$$

with matrix

$$C = \left[0_{N_y \times N_c - N_y}, \mathrm{diag}\left(\tilde{c}_1, \ldots, \tilde{c}_{N_y}\right)\right] \tag{7.27}$$

and we obtain the bijective mapping

$$\begin{aligned} y_1 &= \tilde{c}_1\ \Theta_{N_c - N_y + 1}, \\ y_2 &= \tilde{c}_2\ \Theta_{N_c - N_y + 2}, \\ &\vdots \\ y_{N_y} &= \tilde{c}_{N_y}\ \Theta_{N_c}. \end{aligned}$$

Due to the block matrix structure of A in the two- and three-dimensional case, we compute a suitable transformation matrix T_y with C in Eq. (7.27). The second term of the right-hand side in Eq. (7.17) has the dimension

$$\left[CA^{i-1}B\right] \in \mathbb{R}^{N_y \times N_u}$$

and we know that it needs to be invertible for $i = \tilde{N}$. Thus, the number of input and output signals must be equal: $N_u = N_y$. Furthermore, to guarantee $CA^{i-1}B = 0_{N_u \times N_u}$ for $i \in \{1, \ldots, \tilde{N} - 1\}$, we require that the temperature cells, which are affected by the actuation and temperature measurement must be completely separated. This means, they have to be on opposite boundary sides. In other words, the intersection of the index sets of the actuation and sensing must be empty as

$$\left(\bigcup_{n \in \{1,\ldots,N_u\}} S_{\beta,n}\right) \cap \left(\bigcup_{n \in \{1,\ldots,N_u\}} S_{\gamma,n}\right) = \{\}.$$

If we consider output matrix C as in Eq. (7.27), then we have to choose the input matrix as

$$B = \begin{pmatrix} \operatorname{diag}(b_1, \ldots, b_{N_u}) \\ 0_{N_c - N_u \times N_u} \end{pmatrix}. \tag{7.28}$$

The design constraints for matrices B and C does not match well with our concepts of input and output partitions, as introduced in Section 6.1, because one partition usually consists of several cells. In case of the actuation, we may solve this issue as we define one cell per partition and we specify several input signals with the same reference function. However, we are not able to transfer this idea to the output partitions because we have here multiple cells to find an average temperature, see Eq. (6.6). We may solve this issue by introducing a control design with two stages:

1. Low resolution approximation of object Ω where the number of input and output segments coincide with the number of input and output signals. We consider this approximation to compute the flatness-based input signal.

2. High resolution approximation of object Ω with smaller segments and accurate spatial characteristics. We apply the found flatness-based input signals on this precise model to check the open-loop dynamics.

Example: Simulation with three Actuators and Sensors

We exemplify the previous ideas on a rectangle example with length $L = 0.1$ m, width $W = 0.05$ m, and number of grid cells $N_j = 3$ along direction x_1, and $N_m = 5$ along x_2. We consider the actuation on $B_S = (0, L) \times \{0\}$ and the measurement on $B_N = (0, L) \times \{W\}$ with $N_u = N_y = 3$ input and output signals

$$u(t) = (u_1(t), u_2(t), u_3(t))^\top \quad \text{and} \quad y(t) = \big(y_1(t), y_2(t), y_3(t)\big)^\top.$$

The input signals affect the cells with index $i \in \{1, 2, 3\}$ and the output signals measure temperatures of the cells with index $i \in \{N_c - 2, N_c - 1, N_c\}$. This setup is visualized in Fig. 7.8. We consider two setups for input matrix B as in Eq. (7.28) with $b_i = \frac{\tilde{b}}{\Delta x_2 c \rho}$ and output matrix C as in Eq. (7.27). In the first case, the actuator coefficients $\tilde{b}_i$ are different while $\tilde{c}_i$ are equal and in the second case it is vice versa, see Table 7.1.

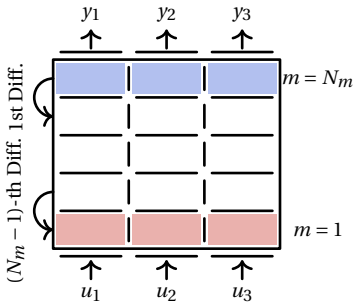


Figure 7.8: Example of flatness-based control for a rectangle with three actuators on boundary B_S and three sensors on B_N.

Table 7.1: Simulation Scenarios.

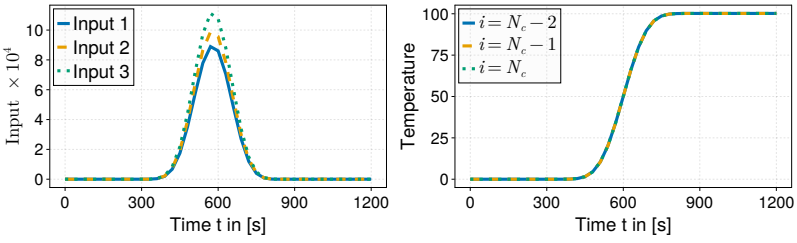
Setup	$(\tilde{b}_1, \tilde{b}_2, \tilde{b}_3)$	$(\tilde{c}_1, \tilde{c}_2, \tilde{c}_3)$
(a)	$(1, 0.9, 0.8)$	$(1, 1, 1)$
(b)	$(1, 1, 1)$	$(1, 0.9, 0.8)$

We assume an initial value of $\Theta(0) = 0_{N_c}$ (not in Kelvin) and so we have the initial output signal as $y(0) = C\Theta(0) = (0, 0, 0)^\top$. All three output signals shall be along the same reference function $r(t) = 100\, \psi(t, p)$ with a transition as in Eq. (7.12) and steepness parameter $p = 2$. We compute the input as $u(t) = M_u \tilde{r}(t)$ with matrix

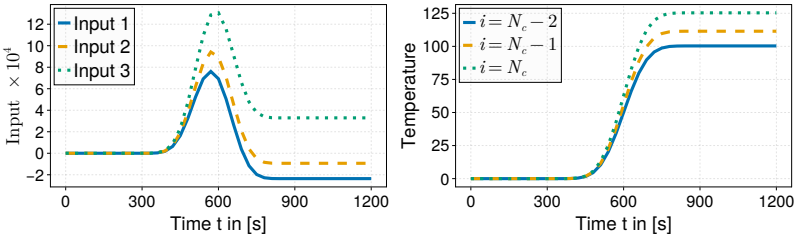
$$M_u := \left[CA^{\tilde{N}-1} B \right]^{-1} \left[-CA^{\tilde{N}} T_y^{-1} | I_{N_y} \right],$$

transformation matrix T_y as in Eq. (7.26) and

$$\tilde{N} = \frac{N_c}{N_y} = \frac{N_j\, N_m}{N_j} = 5.$$



(a) Variation of Actuator Coefficients



(b) Variation of Sensor Coefficients

Figure 7.9: Flatness-based control of the 2-dimensional heat conduction with three input and output signals. The input signal and the resulting temperature evolution is computed for two scenarios with B and C as in Table 7.1. The temperature values of the grid nodes with index $i \in \{N_c-2, N_c-1, N_c\}$, which are related to boundary B_N, are scaled in Scenario (b) because of the choice of matrix C.

We emphasize again that the computation of A^i and its further usage might be numerically sensitive for a "large" value of power i because of the floating point values in $A = \frac{\alpha_1}{\Delta x_1^2} D_1 + \frac{\alpha_2}{\Delta x_2^2} D_2$. The computed input signal and our resulting temperature evolution of $\Theta_i(t)$ for $i \in \{N_c-2, N_c-1, N_c\}$ is portrayed in Fig. 7.9. In case of setup (a), the input signals $u_2(t)$ and $u_3(t)$ are amplified to compensate the scaling of input matrix elements $\tilde{b}_2$ and $\tilde{b}_3$. The temperature on boundary B_N behaves as desired and follows the reference trajectory. In case of setup (b), the computed input signals show an almost Gaussian-like shape but their values do not approach zero after reaching its peak value. We find that u_3 stays at ca. $4 \cdot 10^4$ while u_1 and u_2 drop to negative values. These input values lead to the nonuniform final temperature distribution on boundary B_N where $\Theta_{N_c-2}(T_{ff}) \approx 100$, $\Theta_{N_c-1}(T_{ff}) \approx 110$ and $\Theta_{N_c}(T_{ff}) \approx 125$. This situation is caused by the choice of $(\tilde{c}_1, \tilde{c}_2, \tilde{c}_3)$ and so we yield for the output values

$$
\begin{aligned}
y_1(T_{ff}) &= \tilde{c}_1 \, \Theta_{N_c-2}(T_{ff}) \approx 100, \\
y_2(T_{ff}) &= \tilde{c}_2 \, \Theta_{N_c-1}(T_{ff}) \approx 100, \\
y_3(T_{ff}) &= \tilde{c}_3 \, \Theta_{N_c}(T_{ff}) \approx 100.
\end{aligned}
$$

In this example, we see the influence of matrices B and C on the computation of the input signals and the resulting thermal dynamics. The input signals in Fig. 7.9 (a) are symmetric and Gaussian-shaped like in the one-dim. examples in Fig. 7.4 (c) and Fig. 7.7. In particular, if all reference signals and the coefficients $\tilde{b}_i$ and $\tilde{c}_i$ are equal, then we yield identical input signals and we can consider a one-dim. scenario for the computation of the flatness-based input signal instead of the multi-dim. geometry.

We take up the ideas of setup (a) in our optimization-based feed-forward control to design the symmetric and Gaussian-shaped input signals, see Section 7.4. In contrast to that, we do not further discuss a scenario as in setup (b) with flatness-based control because it shows an undesired input signal.

7.3 Reference Generation

The flatness-based control approach in Section 7.1 and 7.2 depends strongly on the design of reference signal $r(t)$ and its derivatives. In Section 7.1, we constructed the reference with a very smooth transition function of the Gevrey class, which is required for the control design of infinite-dim. systems. Here, we face the problem that computing the transition and its derivatives is not trivial and could be costly, see [36]. In practice, we need to approximate the input signal and the infinite-dim. system to implement it in simulations and control algorithms for industrial controllers. Hence, we know the number of necessary reference derivatives from the number of grid nodes in the spatial approximation. If we fix the number of reference derivatives accordingly, then we can propose a transition with a finite number of smooth derivatives. Subsequently, we discuss three design approaches of transition ψ for the flatness-based control of finite-dim. systems and we explain how to compute its derivatives. These approaches have in common that the computation of transition derivatives is less challenging than in case of Gevrey functions.

In case, we do not require derivatives of the reference function, e.g. in a pure numerical control design as in article [40], then we may even assume very simple transitions like

$$\psi(t) = \frac{1}{2}\left[1 - cos\left(\pi \frac{t}{T_{ff}}\right)\right].$$

Standard Polynomial Approach

First of all, we present a polynomial approach

$$\psi(t,N) = \sum_{n=1}^{2N+1} c_n \left[\frac{t}{T_{ff}}\right]^n \tag{7.29}$$

with coefficients $c_n \in \mathbb{R}$ and $N > 0$ to model the transition. Here, the number $N > 0$ represents the order of the highest reference derivative. We drop the dependency of order N in $\psi(t,N)$ below to improve the readability. We require that the transition starts at zero and reaches one as

$$\psi(0) \stackrel{!}{=} 0 \quad , \quad \psi(T_{ff}) \stackrel{!}{=} 1 \tag{7.30a}$$

and the derivatives at the initial and final time must vanish as

$$\frac{d^n}{dt^n}\psi(t)\bigg|_{t=0} = \frac{d^n}{dt^n}\psi(t)\bigg|_{t=T_{ff}} \stackrel{!}{=} 0 \tag{7.30b}$$

for $n \in \{1,\dots,N\}$. We find the coefficients c_n of transition $\psi(t)$ with an evaluation of the constraints in Eq. (7.30). In particular, we evaluate Eq. (7.30a) at $t = T_{ff}$ as

$$\psi(T_{ff}) = \sum_{n=1}^{2N+1} c_n = 1. \tag{7.31}$$

We continue with i-th derivative at the initial time $t = 0$ as in Eq. (7.30b), where the i-th coefficient c_i must be zero for $i \in \{1,\dots,N\}$ because of

$$\frac{d^i}{dt^i}\psi(t)\bigg|_{t=0} = T_{ff}^{-i}\, i!\, c_i = 0. \tag{7.32}$$

We calculate the derivative at the final time $t = T_{ff}$ as

$$\frac{d^i}{dt^i}\psi(t)\bigg|_{t=T_{ff}} = T_{ff}^{-i}\sum_{n=i}^{2N+1}\frac{n!}{(n-i)!}\,c_n = 0.$$

As we know from Eq. (7.32) that $c_i = 0$ for $i \in \{1,\dots,N\}$, we start the summation at index $i = N$ and we yield

$$\sum_{n=N+1}^{2N+1}\frac{n!}{(n-i)!}\,c_n = 0. \tag{7.33}$$

We collect the previous findings from Eq. (7.31, 7.33) and we solve the linear equations

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \frac{(N+1)!}{(N-1)!} & \frac{(N+2)!}{N!} & \dots & \frac{(2N+1)!}{(2N)!} \\ \frac{(N+1)!}{(N-2)!} & \frac{(N+2)!}{(N-1)!} & \dots & \frac{(2N+1)!}{(2N-1)!} \\ \vdots & \vdots & & \vdots \\ \frac{(N+1)!}{1!} & \frac{(N+1)!}{2!} & \dots & \frac{(2N+1)!}{(N+1)!} \end{pmatrix}\begin{pmatrix} c_{N+1} \\ c_{N+2} \\ \vdots \\ c_{2N+1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{7.34}$$

for the coefficients c_i. Transition $\psi(t,N)$ and its first derivative $\frac{d}{dt}\psi(t,N)$ are portrayed in Fig. 7.10 for $N \in \{2,5,10\}$ and we note that a higher order N leads to a steeper transition. In summary, the polynomial approach (7.29) has in total N series elements c_i, which we find by solving Eq. (7.34), and the highest order term has the exponent $2N+1$.

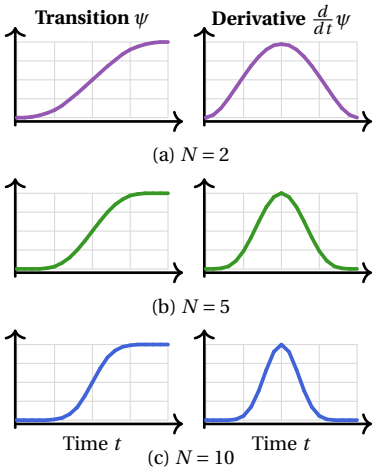


Figure 7.10: Transition ψ and derivative $\frac{d}{dt}\psi$ for order $N \in \{2,5,10\}$ as in Eq. (7.29). An increasing order N leads to a steep transition.

Integration of Bump Functions

In Section 7.1 we described how to compute the transition by integrating the bump function $\omega(t,p)$ in Eq. (7.12). Now, we transfer these concepts to transition functions with finite order. So, we consider the bump function

$$\omega(t,N) = \left[t - t^2\right]^N = \sum_{n=0}^{N}\binom{N}{n}(-1)^n\,t^{N+n}$$

and we integrate it as

$$\psi(t,N) = \frac{\int_0^t \omega\left(\frac{\tau}{T_{ff}},N\right)d\tau}{\int_0^{T_{ff}} \omega\left(\frac{\tau}{T_{ff}},N\right)d\tau} \tag{7.35}$$

to yield a transition ψ as in Eq. (7.29). We solve the integral in the numerator of Eq. (7.35) as

$$\int_0^t \omega\left(\frac{\tau}{T_{ff}},N\right)d\tau = T_{ff}\sum_{n=0}^{N}\binom{N}{n}(-1)^n\,[N+n]^{-1}\left[\frac{t}{T_{ff}}\right]^{N+n+1} \tag{7.36}$$

with the binomial coefficient

$$\binom{N}{i} = \frac{N!}{i!(N-i)!}.$$

We find the denominator in Eq. (7.35) by evaluating the right-hand side of Eq. (7.36) at $t = T_{ff}$ as

$$\int_0^{T_{ff}} \omega\left(\frac{\tau}{T_{ff}},N\right)d\tau = T_{ff}\sum_{n=0}^{N}\binom{N}{i}(-1)^n\,[N+n]^{-1}$$

and we yield the polynomial transition in Eq. (7.29) as

$$\psi(t,N) = \frac{\sum_{n=0}^{N} \binom{N}{n} (-1)^n \ [N+n]^{-1} \ \left[\frac{t}{T_{ff}}\right]^{N+n+1}}{\sum_{n=0}^{N} \binom{N}{n} (-1)^n \ [N+n]^{-1}}.$$

Beside a polynomial, we can also consider other bump functions. If we assume a sine function as $\omega(t,N) = [\sin(\pi t)]^{2N}$, then we find the transition

$$\psi_{sin}(t,N) = \frac{\int_0^t \left[\sin\left(\frac{\tau\,\pi}{T_{ff}}\right)\right]^{2N}}{\int_0^{T_{ff}} \left[\sin\left(\frac{\tau\,\pi}{T_{ff}}\right)\right]^{2N}}. \tag{7.37}$$

In Eq. (7.37), we may apply trigonometric identities like

$$\sin(z)^2 \equiv \frac{1}{2}(1-\cos(2z))$$

to compute the integrals in closed form.

Hyperbolic Tangent

Additionally to the transition functions, which are based on a polynomial in Eq. (7.29) or on a sine in Eq. (7.37), we propose the simple, but inexact, transition

$$\psi(t,p) = \frac{1}{2}\left[1+\tanh\left(p\ \left[\frac{t}{T_{ff}} - \frac{1}{2}\right]\right)\right]. \tag{7.38}$$

In case of this transition, we do not need to compute several coefficients or solve integrals as above, and we can simply specify the steepness with parameter $p > 1$. Though, the constraints at the initial and final time do not match exactly as

$$\psi(0,p) \neq 0 \quad \text{and} \quad \psi(T_{ff},p) \neq 1$$

for any parameter choice $p > 1$. We evaluate identity

$$\tanh(z) = 1 - \frac{2}{\exp(2z+1)}$$

in Eq. (7.38) and find at the initial and final time

$$\psi(0,p) = 1 - \frac{\exp(p)}{\exp(p)+1}, \tag{7.39a}$$

$$\psi(T_{ff},p) = 1 - \frac{1}{\exp(p)+1}. \tag{7.39b}$$

If parameter p increases, then the transition approaches the desired values as

$$\lim_{p\to\infty} \big(\psi(0,p), \psi(T_{ff},p)\big) = (0,1).$$

If we fix a certain initial value, e.g. $\psi(0,p) = \psi_0$, then we compute the necessary value of the steepness with Eq. (7.39a) as

$$p = \ln\left(\frac{1}{\psi_0} - 1\right). \tag{7.40}$$

Table 7.2: Steepness and initial value as in Eq. (7.40).

ψ_0	10^{-1}	10^{-2}	10^{-3}	10^{-4}
p	2.2	4.6	6.9	9.2

In Table 7.2, we list four example relations of Eq. (7.40). The reference tracking shall also perform well and robustly, if the initial reference value and the corresponding output signal do not match exactly, $r(0) - y(0) > 0$. For our simulations, we require $p \geq 7$ or $\psi_0 \leq 10^{-3}$ because we yield the initial reference as

$$r(0) = r_0 + \underbrace{\Delta r \; \psi(0,p)}_{<1} \approx r_0$$

with e.g. $\Delta r = 100$ Kelvin. We visualize the transition and its first derivative in Fig. 7.11 and we remark the offset for $p \in \{3,5\}$ at the initial and final time, $t = 0$ and $t = T_{ff}$. In case of $p = 7$, we notice almost no offset in the transition and its derivative.

We differentiate the transition in Eq. (7.38) and obtain

$$\begin{aligned} \frac{d}{dt}\psi(t,p) &= p\left[1 - \tanh\left(t\,\frac{p}{T_{ff}} - \frac{p}{2}\right)^2\right] \\ &= \frac{p}{2T_{ff}}\left[\psi(t,p) - \psi(t,p)^2\right], \end{aligned} \tag{7.41}$$

which is in form of the Riccati differential equation

$$\frac{d}{dz}f(z) = c_0 + c_1 f(z) + c_2 f(z)^2$$

with coefficients $c_0 = 0$, $c_1 = \frac{p}{2T_{ff}}$ $c_2 = -\frac{p}{2T_{ff}}$. According to article [135], we find the n-th order derivative of ψ as

$$\frac{d^n}{dt^n}\psi(t,p) = \left[\frac{p}{2T_{ff}}\right]^n \sum_{i=0}^{n-1} \left\langle {n \atop i} \right\rangle \left(\psi(t,p) - 1\right)^{i+1} \; \psi(t,p)^{n-i}$$

with the Eulerian number

$$\left\langle {n \atop i} \right\rangle = \sum_{j=0}^{i} (-1)^j \binom{n+1}{j} (i+1-j)^n. \tag{7.42}$$

For further information on Eulerian numbers we to the book [136, p. 242]. As for the transition, the constraints for the derivatives at the initial and final time do not match as

$$\left.\frac{d^n}{dt^n}\psi(t,p)\right|_{t=0} \neq 0 \quad \text{and} \quad \left.\frac{d^n}{dt^n}\psi(t,p)\right|_{t=T_{ff}} \neq 0.$$

We solve this issue with a proper choice of the steepness, e.g. $p > 7$, see Fig. 7.11.

In conclusion, we are able to compute reference transitions for differential equations with a high number of states either exactly, e.g. with a polynomial approach as in Eq. (7.29) and (7.35), or approximately with the hyperbolic tangent in Eq. (7.38). The steepness of the polynomial approach can only be specified for a discrete order of the polynomial, while the steepness of the tanh approach can be set continuously. In Section 7.6, we exemplify the reference design with the tanh approach.

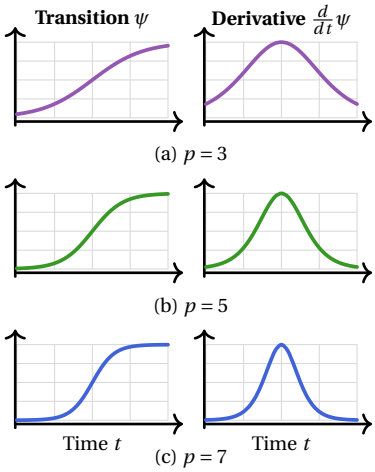


Figure 7.11: Transition ψ and derivative $\frac{d}{dt}\psi$ for $p \in \{3,5,7\}$ as in Eq. (7.38).

7.4 Optimization-based Feed-forward Control

In the previous sections, we discussed the design of a reference function and the computation of an input signal with flatness-based control. In case of the one-dim. heat conduction, the input signal tends to a bell-shaped function if the final time T_{ff} is long enough, see Fig. 7.4. We remind that this shape corresponds to the first derivative of reference $r(t)$ and the influence of higher order derivatives need to be minimized by choosing a suitable T_{ff} and steepness to guarantee only positive input values. As flatness-based control is rather limited to linear heat conduction problems[5] for our scenarios, we extend the feed-forward control design with numerical optimization approaches to handle our original nonlinear scenario. Therefore, we treat flatness-based control as a prototyping stage for the optimization-based control design, which is capable to handle the full heat conduction setup with thermal emissions, temperature-dependent material coefficients and spatial characteristics for actuators and sensors. This concept is described in our article [39]. We approximate bell-shaped flatness-based input signal by the Gaussian function

[5] The flatness-based control design for a specific type of quasilinear parabolic PDE is described in the doctoral thesis [111].

$$u_{oc}(t,p) := \exp\left(p_1 - p_3^2\left[\frac{t}{T_{ff}} - \frac{1}{p_2}\right]^2\right) \tag{7.43}$$

with signal gain $p_1 \geq 0$, time shift $p_2 > 0$ relative to final time T_{ff} and width or kurtosis of the bell shape $p_3 \geq 0$. An increasing value of p_2 shifts the center of u_{oc} closer to the origin, and an increasing value of p_3 decreases the shape width. The optimization-based input signal (7.43) is visualized with its parameters in Fig. 7.12. We find the first derivative in time as

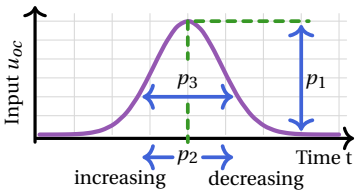


Figure 7.12: The optimization-based input signal is designed as a Gaussian function with parameters p_1 as gain, p_2 as time shift and p_3 as width of the bell shape.

$$\frac{d}{dt}u_{oc}(t;p) = -2\,\frac{p_3^2}{T_{ff}}\left[\frac{t}{T_{ff}} - \frac{1}{p_2}\right]\,u_{oc}(t;p)$$

and this derivative vanishes, $\frac{d}{dt}u_{oc}(t;p) = 0$, at the peak value of $u_{oc}(t,p)$, $\frac{d}{dt}u_{oc}(t;p) = 0$ at $t = \frac{T_{ff}}{p_2}$. At this time, we yield the maximum value as

$$\max_{t\in[0,T_{ff}]} u_{oc}(t,p) = u_{oc}\left(\frac{T_{ff}}{p_2}, p\right) = \exp(p_1). \tag{7.44}$$

In our subsequent examples, we have this peak value close to $\frac{T_{ff}}{2}$ or equally $p_2 \approx 2$. The input signal is positive for all $t \in [0, T_{ff}]$ and does not start at zero as

$$u_{oc}(0;p) = \exp\left(p_1 - \left[\frac{p_3}{p_2}\right]^2\right) \neq 0$$

for any parameter choice $p = (p_1, p_2, p_3)$. Thus, we specify a very small initial value $u_{oc}(0;p) = u_0 > 0$ and we determine the parameters such that equation

$$\exp\left(p_1 - \left[\frac{p_3}{p_2}\right]^2\right) \equiv u_0$$

holds. In the function approximation of the flatness-based input signal we easily derive the parameters p_1 and p_2 via the maximum value, but finding

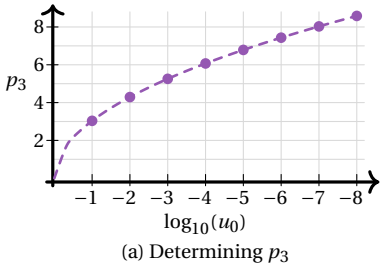
a suitable p_3 needs more effort. Hence, we may determine p_3 via the initial value as

$$p_3 = p_2\sqrt{p_1 - \ln(u_0)}. \tag{7.45}$$

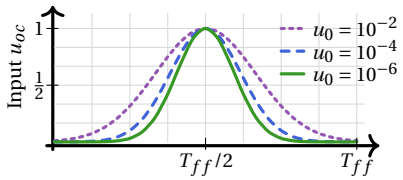
If we choose a small u_0, then yield a large kurtosis p_3 and a narrow bell shape. In Fig. 7.13 (a), we find for $u_0 = 10^{-6}$ the kurtosis $p_3 \approx 7.4$ and we yield the green input signal in Fig. 7.13 (b).

The parameter optimization of u_{oc} with gradient-based techniques in the next paragraphs and so we calculate the parameter gradient as

$$\nabla_p u_{oc}(t;p) = \begin{pmatrix} 1 \\ -2\left[\frac{p_3}{p_2}\right]^2\left[\frac{t}{T_{ff}} - \frac{1}{p_2}\right] \\ -2\,p_3\left[\frac{t}{T_{ff}} - \frac{1}{p_2}\right]^2 \end{pmatrix} u_{oc}(t;p).$$



(a) Determining p_3



(b) Width of bell shape

Figure 7.13: Determining parameter p_3 for a given initial value as in Eq. (7.45) with $p_1 = 0$ and $p_2 = 2$ in (a). The shape width is shrinking for a decreasing initial value u_0 in (b).

Approximation of Flatness-based Input Signal

In the next steps, we approximate the flatness-based input signal with u_{oc} and optimize the parameters of u_{oc} such that it fits to a heat conduction model with nonlinear terms. To exemplify these steps, we assume a one-dim. heat conduction model with boundary conditions

$$-\lambda \left.\frac{d}{dx_1}\vartheta(t,x)\right|_{x_1=0} = u(t) + \phi_{em}(t,0), \tag{7.46a}$$

$$\lambda \left.\frac{d}{dx_1}\vartheta(t,x)\right|_{x_1=L} = \phi_{em}(t,L) \tag{7.46b}$$

and thermal emissions as in Eq. (2.33). We assume the same material properties as in the example of Section 7.1, see also Table 7.3.

Table 7.3: Example coefficients.

L	λ	ρ	c
0.1	50	8000	400

We sample the heat conduction in space and we design a flatness-based control for the simplified model as described in Section 7.2, where we drop the thermal emissions as $\phi_{em} \equiv 0$. We set the final time $T_{ff} = 3000$ seconds and we specify reference signal $r(t)$ with the Gevrey-type transition as explained in Section 7.1. We yield the input signal u_{fbc} as in Fig. 7.7 and we restrict it to positive values only as

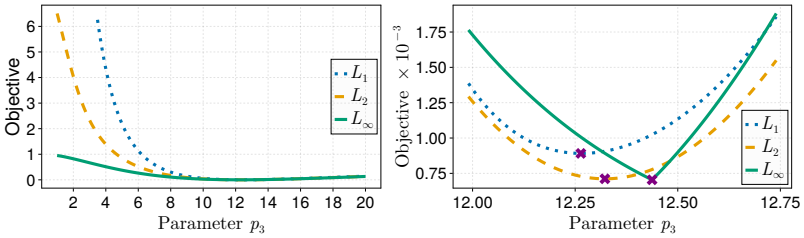
$$\hat{u}_{fbc}(t) := \max\big(u_{fbc}(t), 0\big).$$

The main idea of this first step is to find suitable parameters $p = (p_1, p_2, p_3)$ to minimize the error between the flatness-based and optimization-based input signal as

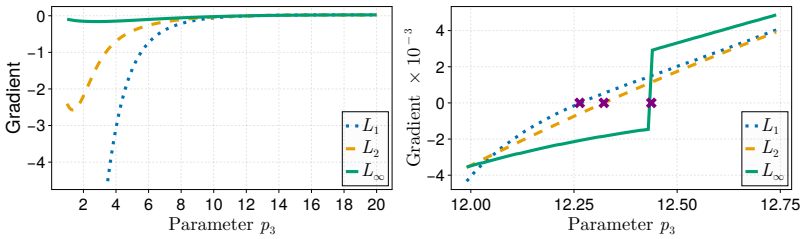
$$\min e_{fbc}(t,p) \quad \text{with} \quad e_{fbc}(t,p) := \hat{u}_{fbc}(t) - u_{oc}(t,p). \tag{7.47}$$

We derive gain p_1 and time shift p_2 directly from the data of $\hat{u}_{fbc}$, but we need to approach the width p_3 via a numerical optimizer. This procedure is visualized in Fig. 7.14. The found parameters of this step are treated as initial values for the optimization routines of the subsequent steps. The flatness-based input signal $\hat{u}_{fbc}$ reaches its peak value at $t = t_{max}$ and we obtain with Eq. (7.44) the identity

$$\begin{aligned} \max_{t\in(0,T)} \hat{u}_{fbc}(t) &= \hat{u}_{fbc}(t_{max}) \\ &= u_{oc}(t_{max}, p) = \max_{t\in(0,T_{ff})} u_{oc}(t,p) \\ &= \exp(p_1). \end{aligned}$$



(a) Objective function $J(p_3)$



(b) Objective function gradient $\frac{d}{dp_3} J(p_3)$

Figure 7.15: Objective function and its gradient for the norms L_1, L_2 and L_∞. The minimum costs (purple crosses) reach circa 12.3 for L_1 and L_2, and 12.4 for L_∞, see also Table 7.4. The objective function for L_∞ is not continuous at its minimum and so its gradient shows a jump.

Thus, we derive the parameters

$$p_1 = \ln(\hat{u}_{fbc}(t_{max})) \quad \text{and} \quad p_2 = \frac{T_{ff}}{t_{max}}.$$

We find $t_{max} = 1401$ seconds and $\hat{u}_{fbc}(t_{max}) \approx 74.8 \cdot 10^3$ and so we yield the parameters of $u_{oc}(t,p)$ as

$$p_1 \approx 11.22 \quad \text{and} \quad p_2 \approx 2.14.$$

In a similar way, we can pick an arbitrary time point $\tilde{t} \in (0, T_{ff}) \setminus \{t_{max}\}$ to find parameter p_3 by solving the equation $\hat{u}_{fbc}(\tilde{t}) = u_{oc}(\tilde{t}, p)$ as

$$p_3 = \left| \frac{\tilde{t}}{T_{ff}} - \frac{1}{p_2} \right|^{-1} \sqrt{p_1 - \ln\left(\hat{u}_{fbc}(\tilde{t})\right)}.$$

Though, we may find for each time $\tilde{t}$ a different parameter p_3 because the flatness-based and optimization-based input signal do not fit perfectly. We wish to avoid such a sensitivity and design an unconstrained optimization problem, which fits the parameter robustly as

$$p_3^* = \arg \min_{p_3 \in (0,\infty)} J(p_3). \tag{7.48}$$

We consider the quadratic objective function

$$J(p_3) = \frac{\|\hat{u}_{fbc}(t) - u_{oc}(t,p)\|^2}{\|\hat{u}_{fbc}(t)\|^2} \tag{7.49}$$

where we distinguish three norms as

$$\|f(t)\|_{L_1} := \int_0^{T_{ff}} |f(t)| dt,$$

$$\|f(t)\|_{L_2} := \sqrt{\int_0^{T_{ff}} f(t)^2 dt},$$

$$\|f(t)\|_{L_\infty} := \max_{t \in (0,T_{ff})} |f(t)|.$$

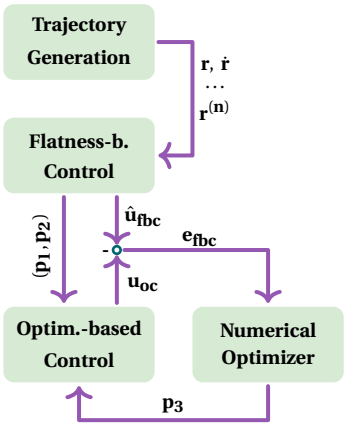
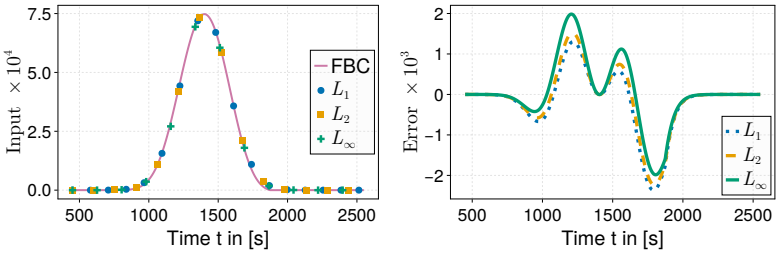


Figure 7.14: Scheme to approximate the flatness-based input signal. Parameters p_1 and p_2 are derived directly from $\hat{u}_{fbc}$, and p_3 is found numerically via the minimization of the error between flatness-based and optimization-based input function.



(a) Input Signal

(b) Error $\hat{u}_{fbc}(t) - u_{oc}(t,p)$

Figure 7.16: Match between the flatness-based and optimization-based input signal with p_3 as in Table 7.4. The start and end time, $t \in [0,500]$ and $t \in [2500,3000]$, is cut to improve the readability. Two remarkable errors occur during the ramp-up and ramp-down phase of the bell function, circa at $t = 1200$ and $t = 1800$ seconds.

We solve the optimization problem (7.48) numerically with a Conjugate Gradient optimizer for these three norms and we yield three different optimal values p_3^*. In Fig. 7.15 we depict function (7.49) and its gradient for each norm. The optimal values of p_3^* are marked as purple crosses and its approximate values are listed in Table 7.4. The objective functions start at high values for L_1 and L_2, they approach zero and continue on low values for $p_3 > 10$. The objective function of L_∞ has a discontinuity at its minimum value, which leads to a jump in the gradient.

We visualize $u_{oc}(t,p)$ for the three p_3 values in Fig. 7.16 (a) with data samples as markers and the original $\hat{u}_{fbc}$ as a line, and we notice that the optimization-based input signal $u_{oc}(t,p)$ fits the flatness-based input $\hat{u}_{fbc}$ quite precisely. Error e_{fbc}, as in Eq. (7.47), shows in Fig. 7.16 (b) two significant peaks during the start-up and shutdown phase, e.g. $t \in (1000,1300)$ and $t \in (1600,2000)$.

In the subsequent optimization routines, we assume the L_2 norm to evaluate the error.

Table 7.4: Found Parameter p_3.

L_1	L_2	L_∞
12.26	12.32	12.44

Parameter Fitting for Reference Tracking

Here, we design the optimization-based input signal for the full, original, heat conduction scenario and we visualize this approach in Fig. 7.17. Input signal $u_{oc}(t,p)$ shall steer the temperature measurements $y(t)$ of the nonlinear thermal dynamics, see Definition 6.1, along a predefined reference trajectory $r(t)$. The numerical optimizer shall find suitable parameters for $u_{oc}(t,p)$ to minimize the error between the reference and the temperature measurement as

$$e_{r,oc}(t,p) := r(t) - y(t,p).$$

The temperature measurement depends on parameter set p because each variation of p changes the thermal dynamics and consequently the measurement y as

$$y(t,p) = C\,\Theta(t,p) = C\left[\int_0^t f_{\mathscr{D}}(\Theta) + B(\Theta)\,u(\tau,p) + w(\tau,\Theta)\,d\tau\right]$$

This means, we do not approximate the input signal u_{oc} via a candidate function as in the previous step. Instead, we need to integrate the thermal dynamics in an intermediate step and compare the resulting output temperature with the reference signal. Here, we adapt only gain p_1 and time shift p_2 while we fix kurtosis p_3 with the value of the previous step. We find

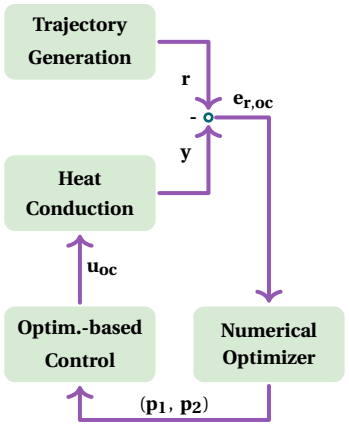
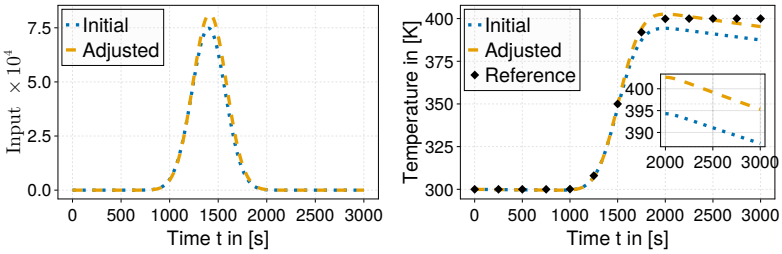


Figure 7.17: Scheme to fit the parameters for reference tracking. The optimization-based input u_{oc} steers the heat conduction. The measured temperature y is compared with the reference r and their difference shall be minimized by the numerical optimizer to find optimal parameters p_1 and p_2.



(a) Input Signal (b) Temperature Measurement

Figure 7.19: Adjusted input signal and resulting temperature measurement for reference tracking. The larger adjusted input signal in (a) improves the reference tracking as it leads to higher measured temperatures in the end of the heating-up phase in (b). The thermal emissions forces a notable temperature drop of output signals.

the parameters via minimizing the quadratic optimization problem

$$(p_1^*, p_2^*) = \arg\min_{(p_1,p_2)} \frac{1}{T} \left\| e_{r,oc}(t,p) \right\|_{L_2}^2 \tag{7.50}$$

with subject to the nonlinear thermal dynamics as in Eq. (6.11). Now, we consider thermal emissions in the boundary conditions again, see Eq. (7.46). The emissive heat flux ϕ_{em} disturbs the desired thermal dynamics because it forces a cooling on both sides of the rod. Thus, the original input signal in Fig. 7.16 (a) must be amplified to compensate the temperature drop caused by the thermal emissions.

We initialize the optimization problem in Eq. (7.50) with $p_1 \approx 11.22$ and $p_2 \approx 2.14$ from the previous step and we fix $p_3 = 12.32$, see Table 7.4. The numerical optimizer finds the minimum value at $p_1^* \approx 11.31$, $p_2^* \approx 2.12$. The objective function $J(p_1,p_2) := \frac{1}{T} \left\| e_{r,oc}(t,p) \right\|_{L_2}^2$ is convex in a region around the minimum value as depicted in Fig. 7.18. So, the optimizer finds the local optimal value and even the global optimum, if $J(p_1,p_2)$ is convex for all p_1, p_2. We compute the input signal $u_{oc}(t,p)$ with p_1^*, p_2^* and we apply it on the heat conduction problem, see Fig. 7.19. The adjusted input signal in Fig. 7.19 (a) is slightly larger than the initial one and consequently, we have higher temperatures of the output signal for $t > 1750$ seconds in Fig. 7.19 (b). The thermal emission are partially compensated, but the adjusted input signal is not able to prevent the temperature drop. We face this situation because u_{oc} has a significant impact only during the time $t \in [1000, 2000]$ seconds while the thermal emissions operate intensively after reaching the desired temperature, e.g. $t > 2000$ seconds. We address this issue in Chapter 8 where we design a feedback controller to stabilize the measured temperature at the desired value.

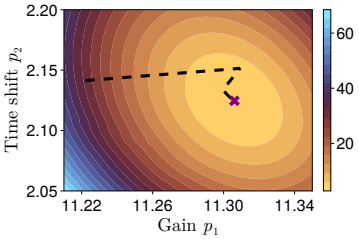


Figure 7.18: Convex objective function $J(p_1,p_2)$ of reference tracking problem (7.50). The dotted line shows the path of optimization routine, starting at $(p_1,p_2) \approx (11.22, 2.14)$ and reaching the optimal value $(p_1^*, p_2^*) \approx (11.31, 2.12)$ (purple cross).

Numerical Optimization Methods

We implement the numerical optimization routines with the Julia libraries *Optimization.jl* [139], *Optim.jl* [140] and *ForwardDiff.jl* [141]. The library *Optim.jl* provides a Conjugate Gradient method, which is implemented with concepts from the articles [142,143], see also the documentation [144]. For an introduction to the Conjugate Gradient method, we refer to the article [145] and to the books [146, p. 121] and [147, p. 70] . This optimization technique requires the specification of a gradient of the objective function, which is neither computed analytically nor numerically here. Instead, we compute the gradients with Algorithmic Differentiation[6] in forward accumulation mode, which is implemented in the library [141]. Algorithmic

[6] Also known as *Automatic Differentiation.*

differentiation applies gradients via the chain rule on source code, we refer to the literature [147, p. 27] and [148, 149] for further information.

7.5 Energy-based Feed-forward Control

The main concept to heat up the considered object is to supply thermal energy via distributed actuators on the boundary sides. In the previous sections, we derived the input signals from a pure "equation-based" point of view without an intense consideration of the physical model. Now, we include the amount of thermal energy to heat up the object and we extend these ideas in Chapter 8 to stabilize the reached operating temperature. This energy-based control design is also explained in article [40]. The energy-based formulation is described by the first law of thermodynamics in Eq. (2.8) and we know that the supplied and emitted power $P(t)$ changes the internal energy $U(t)$ as described by

$$\frac{d}{dt}U(t) = \frac{d}{dt}Q(t) + P(t) \tag{7.51}$$

with the rate of heat flow $\frac{d}{dt}Q(t)$, see Eq. (2.19). The variation of the internal energy $\frac{d}{dt}U(t)$ is solemnly driven by the supplied and emitted power $P(t)$ because the rate of heat flow $\frac{d}{dt}Q(t)$ describes the spatial temperature variation and does not generate energy. This fact leads to

$$\frac{d}{dt}Q(t) = \int_\Omega \operatorname{div}\left[\lambda(\vartheta(t,x))\,\nabla\vartheta(t,x)\right] dx \equiv 0 \tag{7.52}$$

for all $t \geq 0$. At the initial time $t = 0$ we consider a uniform temperature distribution $\vartheta(0,x) = r(0)$ with reference signal $r(t)$. So, we have a vanishing temperature gradient $\nabla\vartheta(t,x) \equiv 0$, and we have an initial internal energy $U(0) = U_0$ and power $P(0) = 0$. As we supply power $P(t) > 0$ for $t > 0$, the internal energy and the temperature increase and we have a temperature gradient $\nabla\vartheta(t,x) \neq 0$. We desire to reach a constant temperature and internal energy level at $t = T_{ff}$, which requires again a uniform temperature distribution $\vartheta(T_{ff},x) = r(T_{ff})$ with $\nabla\vartheta(t,x) \equiv 0$ and $P(T_{ff}) = 0$. In Section 2.4, we introduced the supplied and emitted power, P_{in} and P_{em}, as the integral of their corresponding heat fluxes. In accordance with these ideas, we note the overall sum of both parts as

$$\begin{aligned} P(t) &= \int_{\partial\Omega} \phi(t,x)dx \\ &= \int_{B_{in}} \phi_{in}(t,x) + \phi_{em}(t,x)dx + \int_{\partial\Omega\setminus B_{in}} \phi_{em}(t,x)dx \\ &= \underbrace{\int_{B_{in}} \phi_{in}(t,x)dx}_{=P_{in}(t)\text{ (heating)}} + \underbrace{\int_{\partial\Omega} \phi_{em}(t,x)dx}_{=P_{em}(t)\text{ (cooling)}} \end{aligned}$$

in which the right-hand side is split into the heating up and cooling down phenomena, see Eq. (2.24, 2.25). We continue these ideas for the energy and we integrate Eq. (7.51) in time to find the change of internal energy as

$$\begin{aligned} \Delta U &= \int_0^{T_{ff}} \frac{d}{dt}U(t)dt = \underbrace{\int_0^{T_{ff}} \frac{d}{dt}Q(t)dt}_{=0} + \int_0^{T_{ff}} P(t)dt \\ &= E_{in} + E_{em} \end{aligned} \tag{7.53}$$

with the **supplied thermal energy**

$$E_{in} := \int_0^{T_{ff}} P_{in}(t)\,dt = \int_0^{T_{ff}} \left[\int_{B_{in}} \phi_{in}(t,x)\,dx \right] dt \tag{7.54}$$

and the **emitted thermal energy**

$$E_{em} := \int_0^{T_{ff}} P_{em}(t)\,dt = \int_0^{T_{ff}} \left[\int_{\partial\Omega} \phi_{em}(t,x)\,dx \right] dt. \tag{7.55}$$

In the beginning of Chapter 6 we already discussed that we are in general not able to capture the total emitted heat flux in case of convective and radiative boundary conditions. So, we cannot find a suitable ϕ_{in} to reach $P(t) = P_{in}(t) + P_{em}(t) = 0$. In this section, we discuss in three steps how to design the optimization-based input signal u_{oc} via quantifying the supplied and emitted thermal energy. Firstly, we assume to know the emitted thermal energy E_{em} and we derive a parameter fitting problem to compute an appropriate supplied energy $E_{in}(p)$ such that identity (7.53) is guaranteed. Secondly, we estimate the emitted energy E_{em} during the heating-up phase using the reference signal as an assumption of the temperature prediction. Finally, we discuss further applications of the energy considerations to fine-tune the found parameters.

We calculate the change of internal energy in energy balance (7.53) as

$$\Delta U = \int_0^{T_{ff}} \int_{\Omega_3} c\rho\dot{\vartheta}(t,x)\,dx\,dt = c\,\rho\,|\Omega_3|\,\Delta r \tag{7.56}$$

with $\Delta r = r(T_{ff}) - r(0)$ and volume $|\Omega_3| = L \cdot W \cdot H$ for a cuboid.[7] We assume a constant density ρ and specific heat capacity c in Eq. (7.56). If these material properties are temperature-dependent, then we need to approximate ΔU in a similar way as the thermal emissions, see Eq. (7.62).

[7] For the one-dim. and two-dim. problem we have $|\Omega_1| = L$ and $|\Omega_2| = L \cdot W$.

We formulate the supplied energy E_{in} in Eq. (7.54) with the spatial characteristics of boundary actuation $b(x)$ in Eq. (6.4) as

$$\begin{aligned} E_{in} &= \int_0^{T_{ff}} \int_{\partial\Omega} \phi_{in}(t,x)\,dx\,dt \\ &= \sum_{n=1}^{N_u} \left(\int_{B_{in}} b_n(x)\,dx \right) \left(\int_0^{T_{ff}} u_n(t)\,dt \right). \end{aligned} \tag{7.57}$$

We define the signal energy of the n-th input signal $u_n(t) = u_{oc,n}(t,p)$, see Eq. (7.43), as

$$\begin{aligned} E_{oc,n}(p) &:= \int_0^{T_{ff}} u_{oc,n}(t)\,dt \\ &= \exp(p_1)\,\frac{\sqrt{\pi}\,T_{ff}}{2\,p_3} \left[\operatorname{erf}\left(p_3 - \frac{p_3}{p_2} \right) - \operatorname{erf}\left(-\frac{p_3}{p_2} \right) \right] \end{aligned} \tag{7.58}$$

with error function $\operatorname{erf}(z) = \frac{2}{\pi} \int_0^z \exp(-\tau^2)\,d\tau$, see also Fig. 7.20. A brief discussion of the error function unveils how the parameters influence the

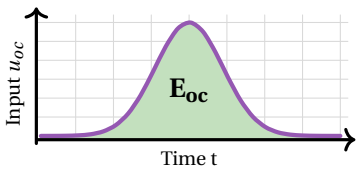


Figure 7.20: Energy of the optimization-based input signal E_{oc} as in Eq. (7.58).

signal energy. The error function behaves similar like a hyperbolic tangent and approaches

$$\lim_{z\to\pm\infty} \operatorname{erf}(z) = \pm 1.$$

If we consider a time shift as $p_2 \approx 2$, then we find

$$\operatorname{erf}\left(p_3 - \frac{p_3}{p_2}\right) - \operatorname{erf}\left(-\frac{p_3}{p_2}\right) \approx 2\operatorname{erf}\left(\frac{p_3}{p_2}\right)$$

and depending on the precision we have $\operatorname{erf}\left(\frac{z}{2}\right) \approx 1$ for $z > 4$. If the kurtosis parameter as p_3 is large enough[8], then we can approach the signal energy as

$$E_{oc,n}(p) \approx \exp(p_1)\frac{\sqrt{\pi}\, T_{ff}}{p_3}. \tag{7.59}$$

Thus, the signal energy can be amplified by increasing gain parameter p_1 or reducing kurtosis parameter p_3.

[8] In Section 7.4 we found $p_3 \approx 12.3 > 4$, see Table 7.4.

We wish to find optimal parameters to compute a suitable supplied energy $E_{in}(p)$ such that the energy balance (7.53) holds. In general, we have N_u actuators with 3 parameters per input signal and so we need to vary $3N_u$ values to find a suitable supplied energy. We simplify this situation as we assume the same parameter set $p = (p_1, p_2, p_3)^\top$ for all actuators and we obtain the supplied energy as

$$E_{in}(p) = E_{oc}(p)\left[\sum_{n=1}^{N_u}\left(\int_{B_{in}} b_n(x)dx\right)\right]$$

Consequently, the distance

$$\Delta U - E_{em} - E_{in}(p) = \Delta U - E_{em} - E_{oc}(p)\left[\sum_{n=1}^{N_u}\left(\int_{B_{in}} b_n(x)dx\right)\right] \tag{7.60}$$

shall be minimized. We may minimize the distance (7.60) numerically as we formulate a quadratic objective function

$$J(p) = \left[\Delta U - E_{em} - E_{in}(p)\right]^2$$

and we search for the minimum with a Conjugate Gradient optimizer. As an alternative way, we may add further conditions to reduce the number of free parameters and to formulate a system of nonlinear equations, which are solved with root-finding algorithms. The latter procedure is described in the end of this section.

Approximation of Emitted Energy

In the previous paragraph, we assumed to know the emitted energy E_{em} in the energy balance $\Delta U = E_{in} + E_{em}$. In some scenarios, we can even neglect E_{em} if its amount is much smaller than the supplied energy. However, it is relevant when we reach the desired temperature $r(T_{ff}) = \Theta_{des}$ because the cooling effect leads to a measurable temperature drop, see also Fig.7.19. We determine the amount of emitted thermal energy as

$$\begin{aligned} E_{em} &:= \int_0^{T_{ff}}\int_{\partial\Omega} \phi_{em}(t,x)dxdt \\ &= \int_{\partial\Omega}\int_0^{T_{ff}} \phi_{em}(t,x)dtdx \\ &= \int_{\partial\Omega}\int_0^{T_{ff}} -h(x)\,[\vartheta(t,x) - \vartheta_{amb}(x)] - \sigma\,\varepsilon(x)\,\vartheta(t,x)^4 dtdx \end{aligned} \tag{7.61}$$

with the heat transfer and radiation coefficients h and ε as described in Definition 2.3. We compute the feed-forward control before the operation of the heating-up process and so we do not know the temperatures along the boundary sides $\partial\Omega$. If we assume small temperature gradients $\nabla\vartheta$ then we may have small variations of the temperatures on boundary $\partial\Omega$. We concentrate the temperatures of the entire object as

$$\tilde{\vartheta}(t) \approx \frac{1}{|\Omega|}\int_\Omega \vartheta(t,x)dx$$

and we note the approximated emitted energy as

$$\begin{aligned}\tilde{E}_{em} := &-\int_{\partial\Omega} h(x)\left[\int_0^{T_{ff}} \tilde{\vartheta}(t)dt - T_{ff}\ \vartheta_{amb}(x)\right]dx\\ &-\sigma\int_{\partial\Omega}\varepsilon(x)dx\int_0^{T_{ff}}\tilde{\vartheta}(t)^4dt.\end{aligned}$$

The measured temperature y has to follow the reference signal r and hence we identify $\tilde{\vartheta}(t) = r(t)$ to compute the energy

$$\begin{aligned}\tilde{E}_{em} = &-\int_{\partial\Omega} h(x)\left[\int_0^{T_{ff}} r(t)dt - T_{ff}\ \vartheta_{amb}(x)\right]dx\\ &-\sigma\int_{\partial\Omega}\varepsilon(x)dx\int_0^{T_{ff}} r(t)^4dt.\end{aligned} \tag{7.62}$$

We remark that the integrals $\int_0^{T_{ff}} r(t)dt$ and $\int_0^{T_{ff}} r(t)^4dt$ can be found symbolically for reference signals with transitions, which are described by polynomials as in Eq. (7.29) or a hyperbolic tangent in Eq. (7.38).

Algebraic Parametrization and Fine-Tuning

We can simplify the parameter search when we include further assumptions. In Section 7.4, we state that the input at the initial time $u_{oc}(t,p) \neq 0$ and we can assume a small value, e.g. $u_0 \ll 1$, to specify the kurtosis, see Eq. (7.45). Here, we approximate the signal energy in Eq. (7.59) by assuming p_3 sufficiently large, e.g. $p_3 > 4$. We sum up both ideas and note the nonlinear equations

$$\exp\left(p_1 - \left[\frac{p_3}{p_2}\right]^2\right) = u_0, \tag{7.63a}$$

$$\exp(p_1)\frac{\sqrt{\pi}\ T_{ff}}{p_3}\ I_\beta\ = \Delta U - E_{em} \tag{7.63b}$$

with integral

$$I_\beta := \sum_{n=1}^{N_u}\int_{B_{in}} b_n(x).$$

If we fix time shift parameter p_2 then we can reformulate the implicit Eq. (7.63) to separate p_1 and p_3 as

$$p_1 - \exp(2p_1)\left[\frac{\sqrt{\pi}\ T_{ff}\ I_\beta}{p_2\ [\Delta U - E_{em}]}\right]^2 - \ln(u_0) = 0, \tag{7.64a}$$

$$u_0\ I_\beta\ \sqrt{\pi}\ T_{ff}\ \exp\left(\left[\frac{p_3}{p_2}\right]^2\right) - p_3\ [\Delta U - E_{em}] = 0. \tag{7.64b}$$

We do not have a trivial solution of Eq. (7.64) at hand and so we need to apply root-finding algorithms to find parameters p_1 and p_3. However,

the algebraic parametrization is sensitive with respect to numerical errors because the right-hand side of (7.63a) is much smaller than (7.63b) and we usually do not find exact parameters such that Eq. (7.63) holds. Instead, we have

$$\exp\left(p_1 - \left[\frac{p_3}{p_2}\right]^2\right) - u_0 = \epsilon_1,$$
$$\exp(p_1)\frac{\sqrt{\pi}\ T_{ff}}{p_3}\ I_\beta - [\Delta U - E_{em}] = \epsilon_2$$

with small errors $\epsilon_1, \epsilon_2 \neq 0$ and we reformulate these equations in an implicit form as

$$p_1 - \exp(2p_1)\left[\frac{\sqrt{\pi}\ T_{ff}\ I_\beta}{p_2\ [\Delta U - E_{em} - \epsilon_2]}\right]^2 - \ln(|u_0 - \epsilon_1|) = 0, \tag{7.65a}$$

$$[u_0 - \epsilon_1]\ I_\beta\ \sqrt{\pi}\ T_{ff}\ \exp\left(\left[\frac{p_3}{p_2}\right]^2\right) - p_3\ [\Delta U - E_{em} - \epsilon_2] = 0. \tag{7.65b}$$

Numerical issues occur in Eq. (7.65) if $|\epsilon_1| \approx u_0$ or $|\epsilon_1| > u_0$ because the natural logarithm in Eq. (7.65a) is very sensitive with respect to ϵ_1 as

$$\left|\frac{d}{d\epsilon_1}\ln(|u_0 - \epsilon_1|)\right| = \left|\frac{-1}{|u_0 - \epsilon_1|}\right| \gg 1$$

for $|\epsilon_1| < 1$. Additionally, error ϵ_1 occurs in Eq. (7.65b) as a linear offset. The other error ϵ_2 has a much smaller impact in Eq. (7.65) because

$$|\epsilon_2| \ll U - E_{em}.$$

Hence, solving the algebraic equations Eq. (7.63) or (7.64) might be a simple and fast procedure to compute parameters p_1 and p_3, but we need to consider the mentioned numerical issues, in particular for Eq. (7.64).

In the end of a parameter optimization we may fine-tune the values to adapt them for specific needs, for example reducing the peak value of an input signal u_{oc}. In this case the signal energy has be constant

$$E_{oc,fix} = E_{oc}(p) = \exp(p_1)\frac{\sqrt{\pi}\ T_{ff}}{p_3}$$

and we adapt the kurtosis parameter p_3 as

$$p_3 = \exp(p_1)\frac{\sqrt{\pi}\ T_{ff}}{E_{oc,fix}}.$$

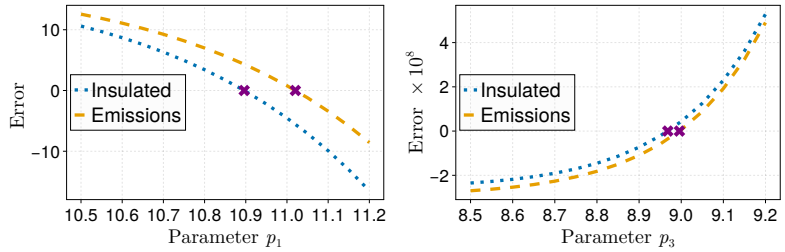
In this manner we can reduce p_1 and increase the bell shape width. However, p_3 still has to be sufficiently large to guarantee the correctness of approximation (7.59).

Example: Energy-based Control

We apply the presented concepts on the same one-dim. heat conduction example as in the previous sections, see Table 7.5. We assume two scenarios of the thermal emissions for the input design: firstly, a completely insulated rod and secondly, heat transfer and radiation on both boundary sides. The found input signal is finally applied on the original model with

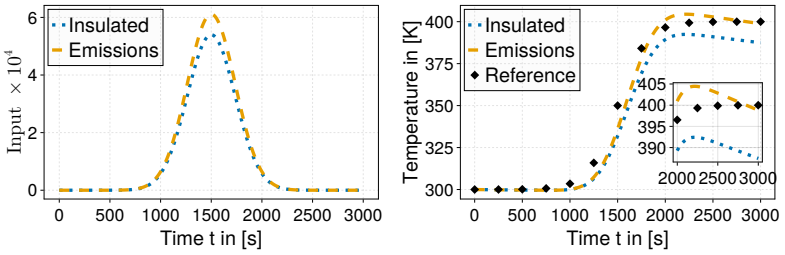
Table 7.5: Example coefficients.

L	λ	ρ	c
0.1	50	8000	400



(a) Gain parameter p_1 (b) Kurtosis parameter p_3

Figure 7.21: Evaluation of implicit function (7.64) with the found parameters (p_1^*, p_3^*) (purple cross). The error denotes the left hand side of function (7.64a) in (a) and (7.64b) in (b).



(a) Input Signal (b) Temperature Measurement

Figure 7.22: Energy-based input design with and without thermal emissions in (a) and resulting output measurement $y(t) = \theta(t, L)$ for a one-dim. model rod in (b). The input signal in (a) for $E_{em} = 0$ does not steer the output to the desired temperature $\Theta_{des} = 400$ Kelvin, but in case of $E_{em} = \tilde{E}_{em} \neq 0$ the output almost reaches Θ_{des}.

thermal emissions on both sides. The coefficients of the thermal emissions are $h = 10$, $\Theta_{amb} = 300$ Kelvin and $\varepsilon = 0.2$.

We wish to change the operating temperature for $\Delta r = 100$ Kelvin and so we evaluate Eq. (7.56) with $|\Omega_1| = L$ to find the change of internal energy as $\Delta U = 32 \cdot 10^6$ Joule. The parameter fitting problem for $u_{oc}(t, p)$ is simplified by fixing $p_2 = 2$ and the initial input value $u_{oc}(0, p) = u_0 = 10^{-4}$. Accordingly, we find the optimal values p_1 and p_3 by solving the implicit functions (7.64). In the first case, we have $E_{em} = 0$ and we find the parameters as listed in the first row of Table 7.6. In the second case, we consider the thermal emissions on both boundaries B_W and B_E and we approximate the emitted energy $E_{em} = \tilde{E}_{em}$ as described in Eq. (7.62). For this purpose we consider the reference signal

$$r(t) = 300 + 50\left[1 + \tanh\left(10\left[\frac{t}{T_{ff}} - \frac{1}{2}\right]\right)\right]$$

as introduced in Section 7.3, see Eq. (7.38). We find the approximated emitted energy $\tilde{E}_{em} \approx 4.12 \cdot 10^6$ Joule and we compute the parameters as noted in the second row of Table 7.6. We notice that the input design with thermal emissions shows an increasing value of p_1 while p_3 is almost on the same level. Hence, increasing the gain is more important here than a wide kurtosis. In Fig. 7.21, we present the value of the left-hand side of function (7.64), which we call "error" here. This error is on a magnitude of 10^8 larger for variations of p_3 than p_1 in a small interval close to the best values. We compute the input signal for both parameter sets of Table 7.6 and apply them on the original model with thermal emissions.

The input signal for $E_{em} = 0$ in Fig. 7.19 (a) is significantly smaller than for $E_{em} = \tilde{E}_{em} \neq 0$ and leads to output temperatures in Fig. 7.19 (b), which do not reach the desired operating temperature $\Theta_{des} = 400$ Kelvin in (b). The input signal for $E_{em} = \tilde{E}_{em} \neq 0$ results in almost proper output temperatures reaching the desired temperature.

Table 7.6: Parameter Fitting.

Scenario	p_1	p_2	p_3
Insulation	10.896	2	8.968
Emissions	11.020	2	8.996

Direction	Temperature 300	350	400	450	500
x_1: λ_1	40	44	50	52	52.5
x_2: λ_2	40	55	60	65	68

Table 7.7: Anisotropic and temperature-dependent thermal conductivity.

7.6 Simulation of the Feed-forward Controlled System

In this section, we demonstrate the full procedure of the control design from modeling to parameter optimization. First of all, we create a full nonlinear heat conduction model including thermal emissions, and spatial characteristics of actuators and sensors. In the second step, we design a prototype input signal with flatness-based control for a simplified version of the original complex model. This simplified model does not contain thermal emissions and temperature-dependent material coefficients. We continue with an approximation of the prototype input using a parameterized Gaussian function $u_{oc}(t,p)$. Here, we return to the original full model and improve the input signal with energy-based considerations and a final parameter optimization.

We consider a flat rectangle Ω_2 with length $L = 0.3$, width $W = 0.05$. The density and specific heat capacity are assumed to be constant as

$$\rho = 8000\,\frac{kg}{m^3} \quad \text{and} \quad c = 400\,\frac{J}{kgK},$$

and the thermal conductivity is considered to be anisotropic and temperature-dependent as $\lambda(\theta) = \text{diag}(\lambda_1(\theta), \lambda_2(\theta))$. We approximate two nonlinear functions of fifth order with the data in Table 7.7 as

$$\lambda_1(\theta) \approx 1465 - 14.8\theta + 56.3\cdot 10^{-3}\theta^2 - 93\cdot 10^{-6}\theta^3 + 56.7\cdot 10^{-9}\theta^4 \quad \text{and}$$
$$\lambda_2(\theta) \approx -2332 + 23\theta - 83\cdot 10^{-3}\theta^2 + 133.3\cdot 10^{-6}\theta^3 - 80\cdot 10^{-9}\theta^4$$

and we visualize them in Fig. 7.23. The rectangle has four boundary sides where B_S is insulated with respect to thermal emissions and the other sides - B_W, B_E and B_N - are open to emit thermal energy. We specify the emitted heat flux as

$$\phi_{em}(t,x) = 10\,[\vartheta(t,x) - \vartheta_{amb}] - 0.1\,\sigma\,\vartheta(t,x)^4 \tag{7.66}$$

with Stefan-Boltzmann constant $\sigma \approx 5.67\cdot 10^{-8}\,\frac{W}{m^2\,K^4}$, see Definition 2.3. We supply energy via three actuators on boundary B_S, which have the spatial characteristics

$$b_n(x) = \exp\left(-\left[30(x - x_{c,n})\right]^4\right) \tag{7.67}$$

as defined in Eq. (6.3) with central points $x_{c,n} = \frac{L}{2}\left[n - \frac{1}{2}\right]$ and $n \in \{1,2,3\}$, see Fig. 7.24. The temperature is measured on B_N with three ideal sensors, $g_n(x) = 1$, and we note the n-th output signal as

$$y_n(t) = \frac{3}{L}\int_{\gamma_n} \vartheta(t,x)dx$$

with $\gamma_n = \left(\frac{L}{3}\,[n-1], \frac{L}{3}\,n\right) \times \{W\}$. The model setup with actuators, sensors and emitted heat flux is visualized in Fig. 7.25

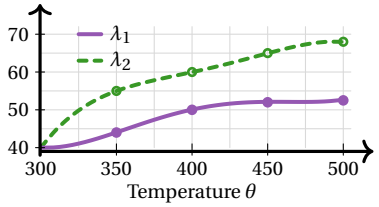


Figure 7.23: Anisotropic and temperature-dependent thermal conductivity $\lambda(\theta) = \text{diag}(\lambda_1(\theta), \lambda_2(\theta))$. The circles mark the data from Table 7.7.

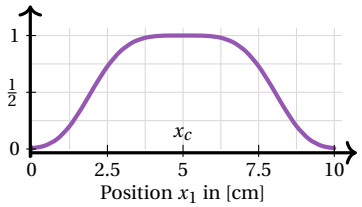


Figure 7.24: Spatial characteristics of first actuator $b_1(x)$ as in Eq. (7.67).

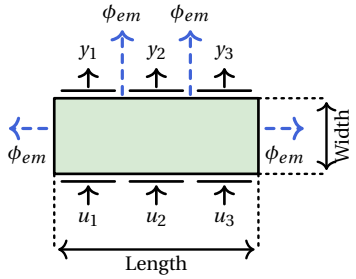


Figure 7.25: Rectangle with three actuators on boundary B_S and three sensors on B_N. Thermal emissions occur on boundaries B_W, B_E and B_N. Boundary B_S is thermally insulated.

Flatness-based Reference Tracking

The measured temperatures shall be steered from the initial temperature

$$\vartheta(0,x) = r(0) = 300 \quad \text{Kelvin}$$

Kelvin towards the desired final temperature

$$\Theta_{des} = r(T_{ff}) = 50 \quad \text{Kelvin.}$$

We design one reference signal

$$\begin{aligned} r(t,p_s) &= 300 + 200\,\psi(t,p_s) = 300 + 100\,\left[1 + \tanh\left(p_s\,\left[\frac{t}{T_{ff}} - \frac{1}{2}\right]\right)\right] \\ &= 400 + 100\,\tanh\left(p_s\,\left[\frac{t}{T_{ff}} - \frac{1}{2}\right]\right) \end{aligned} \tag{7.68}$$

for all actuator / sensor pairs, see Fig. 7.26, and we fix steepness parameter $p_s = 10$ and final time $T_{ff} = 1200$ seconds. For the flatness-based control, we simplify the original model twice. Firstly, we set the thermal conductivity to a constant value, and neglect the thermal emissions. Secondly, we reduce the two-dim. geometry to one dimension along coordinate x_2 (width) because all spatial characteristics of actuators and sensors are identical, $b_1(x) \equiv b_2(x) \equiv b_3(x)$, and we only have one reference signal for all actuator / sensor pairs. The resulting one-dim. rod is spatially approximated with five nodes, $N_c = 5$, and the thermal conductivity is considered as $\tilde{\lambda} = 60\ \frac{W}{mK}$. We follow the ideas of flatness-based control design in Section 7.2 for the one-dim. scenario and we compute input signal u_{fbc} as in Eq. (7.22) with M_u as in Eq. (7.25). For this purpose, we note the reference derivatives as

$$\frac{d^n}{dt^n} r(t,p_s) = 100 \frac{d^n}{dt^n} f(t,p_s)$$

with $f(t,p_s) := \tanh\left(p_s\,\left[\frac{t}{T_{ff}} - \frac{1}{2}\right]\right)$ and its required five derivatives as

$$\begin{aligned} \dot{f}(t,p_s) &= p\left[1 - f(t,p_s)^2\right], \\ \ddot{f}(t,p_s) &= 2p^2\left[-f(t,p_s) + f(t,p_s)^3\right], \\ f^{(3)}(t,p_s) &= 2p^3\left[-1 + 4f(t,p_s)^2 - 3f(t,p_s)^4\right], \\ f^{(4)}(t,p_s) &= 8p^4\left[-2f(t,p_s) - 5f(t,p_s)^3 + 3f(t,p_s)^5\right], \\ f^{(5)}(t,p_s) &= 8p^5\left[2 - 17f(t,p_s)^2 + 30f(t,p_s)^4 - 15f(t,p_s)^6\right]. \end{aligned}$$

We need to restrict the obtained input signal as $\hat{u}_{fbc}(t) = \max(u_{fbc}(t), 0)$ and we approximate it with the parametrized Gaussian function $u_{oc}(t,p)$ as in Eq. (7.43). We find that $\hat{u}_{fbc}(t)$ reaches its maximum at $t = t_{max} = 579.6$ seconds and we have

$$p_1 = \ln(\hat{u}_{fbc}(t_{max})) \approx 11.82 \quad \text{and} \quad p_2 = \frac{T_{ff}}{t_{max}} \approx 2.07.$$

We search for the remaining parameter p_3 by minimizing the objective function (7.49) numerically with the L_2 norm. We depict the objective function in Fig. 7.27 and we find its minimum with a Conjugate Gradient optimizer for parameter $p_3 \approx 9.21$. We assemble the bell-shaped input

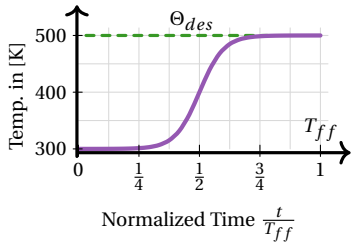


Figure 7.26: Reference signal with hyperbolic tangent as in Eq. (7.68).

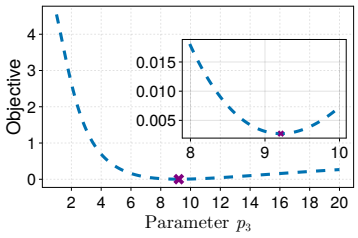
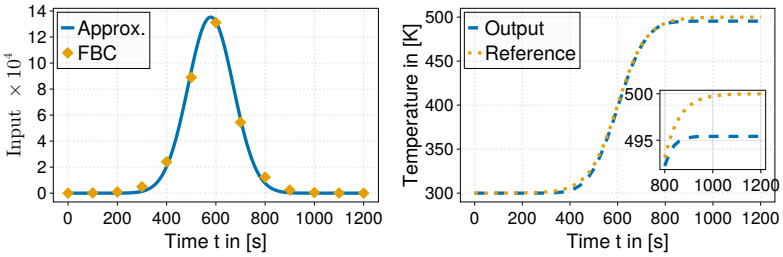


Figure 7.27: Objective function $J(p_3)$ for the norms L_2 norm with its minimum (purple crosses) at $p_3 \approx 9.21$.



(a) Input Signal

(b) Temperature Measurement

Figure 7.28: Approximation of the flatness-based input signal with the Gaussian function u_{oc} in (a). The resulting output measurement $y(t)$ follows the reference, but it does not reach the desired final temperature $\Theta_{des} = 500$ Kelvin in (b).

signal $u_{oc}(t,p)$ with parameters $p = (p_1, p_2, p_3)$, apply it on the one-dim. model and we portray the results in Fig. 7.28. Input $u_{oc}(t,p)$ imitates the flatness-based signal in Fig. 7.28 (a) and so the output $y(t)$ tracks the reference signal in Fig. 7.28 (b) properly, but output $y(t)$ does not reach the desired final temperature $\Theta_{des} = 500$ Kelvin.

The computed parameters of the approximated flatness-based control are treated as initial values for the next optimization step. The parameters of all three steps are listed in Table 7.8 in the end of this section.

Energy Supply

We consider again the original full model. In the previous paragraph we found that the amount of supplied thermal energy is too less to reach the desired temperature of 500 Kelvin. This situation is here even worse because the two-dim. model is equipped with non-ideal actuators, see the spatial characteristics in Eq. (7.67), and thermal emissions. To gain an overview about the energetic situation, we list and compare the internal, supplied and emitted energy. We wish to heat up the two-dim. geometry with area $|\Omega_2| = L \cdot W = 0.015\ m^2$ for $\Delta r = 200$ Kelvin. So, the internal energy shall increase by

$$\Delta U = c\ \rho |\Omega_2| \Delta r = 9.6 \cdot 10^6 \text{ Joule}$$

as described in Eq. (7.56). The initial parameter set $p \approx (12.2, 2.07, 7.88)$ leads to the input signal energy for each actuator of

$$E_{oc}(p) = \int_0^{T_{ff}} u_{oc}(t,p)\, dt \approx 31.24 \cdot 10^6$$

as noted in Eq. (7.58). This amount is multiplied with the integral of the spatial characteristics

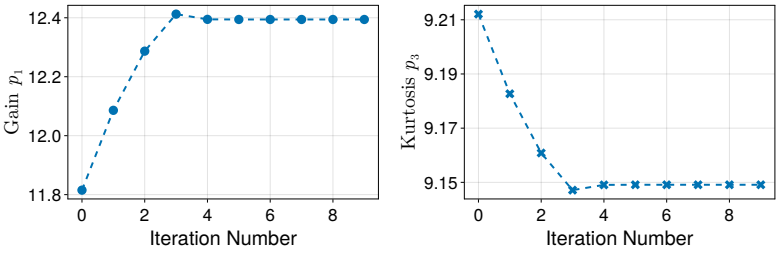
$$\sum_{n=1}^{3} \left(\int_{\beta_n} b_n(x)\, dx \right) \approx 181.24 \cdot 10^{-3}$$

to compute the supplied energy with Eq. (7.57) as

$$E_{in} = E_{oc}(p) \sum_{n=1}^{3} \left(\int_{\beta_n} b_n(x)\, dx \right) \approx 5.66 \cdot 10^6 \text{ Joule}.$$

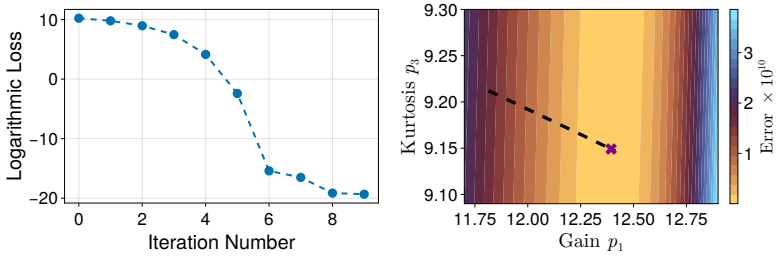
Additionally, we estimate the emitted heat flux with Eq. (7.66) and we approach the emitted energy, see Eq. (7.62), as

$$\tilde{E}_{em} = -571.33 \cdot 10^3 \text{ Joule}.$$



(a) Increasing p_1 (b) Decreasing p_3

Figure 7.29: Parameter values of p_1 and p_3 per iteration in the optimization of the supplied energy. The gain parameter p_1 increases in (a) while the kurtosis parameter p_3 decreases in (b).



(a) Loss $\log_{10}\left(J(p_1,p_3)\right)$ (b) Objective function $J(p_1,p_3)$

Figure 7.30: Loss per iteration in logarithmic scale $\log_{10}(J(p_1,p_3))$ and objective function $J(p_1,p_3)$. The loss approaches zero from the fifth iteration onward in (a). The objective function in (b) has a significant gradient for p_1 in contrast to p_3. The dotted line shows the path of computed parameters reaching the optimal values $(p_1^*,p_3^*)\approx(12.39,9.19)$ (purple cross).

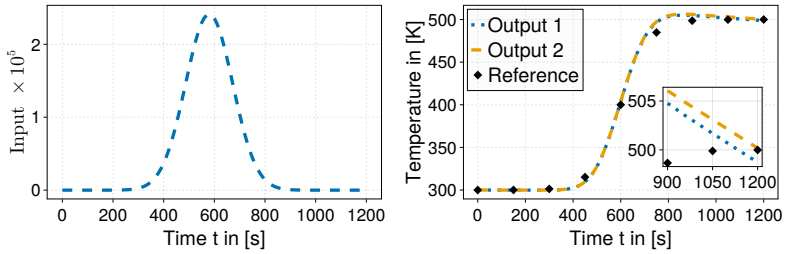
We find that the supplied energy E_{in} for the initial parameters offers only $\frac{E_{in}}{\Delta U-\tilde{E}_{em}}\approx 55.7$ percent of the necessary energy amount to heat up the object properly. Hence, we minimize the distance between the supplied E_{in} and necessary energy $\Delta U+\tilde{E}_{em}$. For this purpose, we fix parameter p_2 and we search with objective function

$$J(p_1,p_3):=\left[\Delta U-\tilde{E}_{em}-E_{oc}(p)\sum_{n=1}^{3}\left(\int_{\beta_n}b_n(x)dx\right)\right]^2$$

for the best parameters p_1^* (gain) or decrease p_3^* (kurtosis) by solving the optimization problem

$$(p_1^*,p_3^*)=\arg\min_{(p_1,p_3)}J(p_1,p_3). \tag{7.69}$$

We know that gain p_1 need to be increased and kurtosis p_3 must be decreased to raise the supplied energy E_{in}. We visualize in ig. 7.29 the intermediate parameters of the numerical optimization and we notice that the optimization behaves as expected. Furthermore, the optimizer finds a local minimum after four iterations and in Fig. 7.30 (a), we see how the loss is driven towards zero from the fifth iteration on. The objective function in Fig. 7.30 (b) looks like a valley because it is significantly steeper in direction p_1 in contrast to p_3. Hence, we see larger variations for p_1 than p_3 in the parameter path approaching the best values $(p_1^*,p_3^*)\approx(12.39,9.19)$. We design all three input signals $u_{oc,n}(t,p)$ with the found parameters and we simulate the heat conduction problem, see Fig. 7.31. The output signals in Fig. 7.31 (b) do not match the reference signal in the second part of the heating-up phase, e.g. $t>600$, because we supply more power than necessary for a reference tracking during this time. On the other side, the output signals reach the desired temperature at $t=T_{ff}=1200$ seconds with this additional power.



(a) Input Signals

(b) Output Signals

Figure 7.31: Input and output signals of the first and second actuator and sensor of the energy-based parameter search. The input signal in (a) is applied on all three actuators. The output signals in (b) are higher than the reference signal for $t > 600$ seconds, but they match almost the desired final temperature $\Theta_{des} = 500$ Kelvin.

Optimization-based Reference Tracking

The energy-based input design is a simple and fast tool to fit proper parameters, but it neglects the dynamical behavior of the heat conduction phenomena and the output tracking with reference signal $r(t)$. We recapitulate the nonlinear heat conduction problem (6.11) with material properties and spatial characteristics as described in the beginning of this section. We notice that the first and third actuator, u_1 and u_3, face the same physical situation, because

- the boundary conditions on B_W and B_E,
- the spatial characteristics of actuators $b_n(x)$ and sensors $g_n(x)$, and
- the reference signals r_n

are equal. Due to this symmetry, we consider the same set of parameters $p_1 := (p_{1,1}, p_{1,2}, p_{1,3})$ for $u_{oc,1}$ and $u_{oc,3}$, while the central actuator $u_{oc,2}$ has a different set of parameters $p_2 := (p_{2,1}, p_{2,2}, p_{2,3})$. This means, we apply the input signals

$$u(t) := \begin{pmatrix} u_{oc,1}(t, p_1) \\ u_{oc,2}(t, p_2) \\ u_{oc,3}(t, p_1) \end{pmatrix}$$

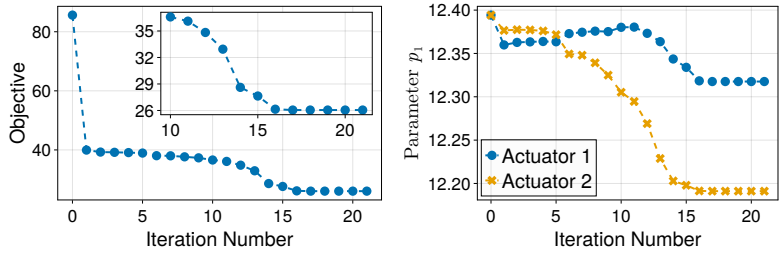
to steer the output signals y_n along the specified reference signal $r(t)$ in Eq. (7.68) with steepness $p_s = 10$. So, we wish to reduce the distance between reference and output

$$e_n(t, p) = r(t) - y_n(t, p)$$

for $n \in \{1, 2, 3\}$ by a suitable choice of parameter sets p_1 and p_2. As we have $u_{oc,1} \equiv u_{oc,3}$, we know that $y_1 \equiv y_3$ and we only need to consider the errors e_1 and e_2. In order to find suitable parameter sets for $u(t)$, we solve the optimization problem

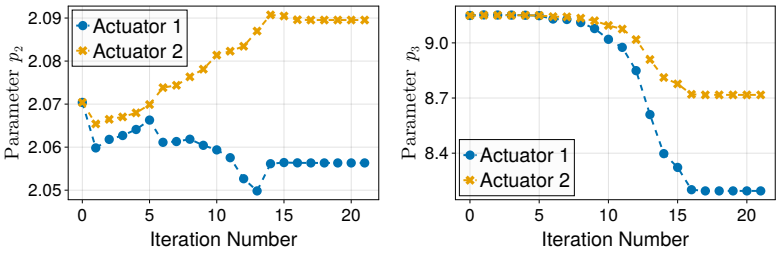
$$(p_1^*, p_2^*) = \arg\min_{(p_1, p_2)} \frac{1}{T} \left\| 2\mu_1 \, e_1(t, p) + \mu_2 \, e_2(t, p) \right\|_{L_2}^2$$

with hyper-parameters $\mu_1 = \mu_2 = 1$. The parameter search is computed numerically with a Conjugate Gradient optimizer for 21 iterations and the resulting parameters are listed in Table 7.8. We notice in Fig. 7.32 that the loss is halved in first iteration of Fig. 7.32 (a) by separating the gain and time shift parameters, p_1 and p_2, for the inner and outer actuators. The loss decreases further and reaches a local minimum where all three parameters show a noticeable separation between $u_{oc,1} = u_{oc,13}$ and $u_{oc,2}$.



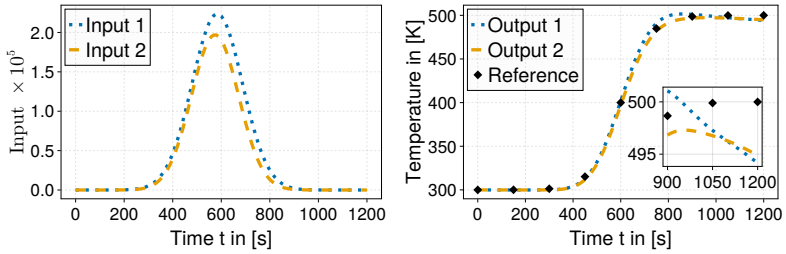
(a) Objective value

(b) Gain p_1



(c) Time Shift p_2

(d) Kurtosis p_3



(a) Input Signals

(b) Output Signals

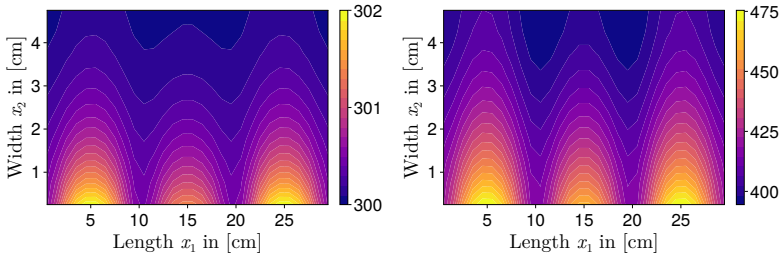
The input signals of the outer actuators have a higher peak value, their peak times are later and their shape kurtosis is wider in comparison to the inner actuator. This means, the actuators close to the boundary sides need to supply more energy than the central actuator to reduce the impact of thermal emissions.

In Fig. 7.33, we portray the computed input and resulting output signals of the first and second actuator and sensor. We find the higher peak value and wider bell shape of input signal $u_{oc,1} = u_{oc,3}$ in Fig. 7.33 (a). The output signals in Fig. 7.33 (b) follow the specified reference well in the first part of the heating-up phase. However, the thermal emissions on boundaries B_W, B_E and B_N cause a temperature drop for $t > 900$ seconds and the output signals do not reach the desired temperature at the final time T_{ff}.

In Fig. 7.34, we present the evolution of the thermal dynamics via snapshots of a temperature distribution in the rectangle. In particular in Fig. 7.34 (a) and (b) we remark the influence of the actuators' spatial characteristics and the higher conductivity λ_2 along the x_2-axis on the temperature distribution. In Fig. 7.34 (c), we find higher temperatures close to the boundary sides B_W and B_E, which are caused by a higher energy supply with $u_{oc,1}$ and $u_{oc,3}$. As the heating stops after $t = 900$ seconds the temperatures close to the boundaries drop due to cooling, and we yield temperatures below the reference value in Fig. 7.34 (d).

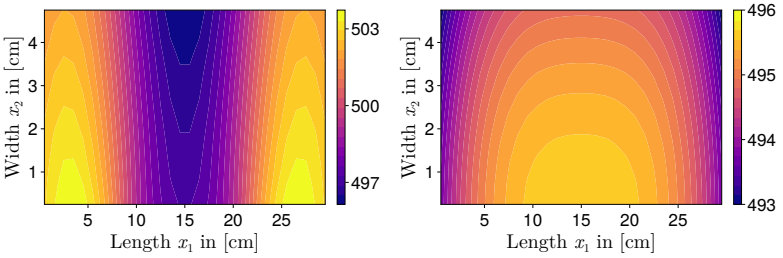
Figure 7.32: Loss and parameters in each optimization iteration of Eq. (7.6). The loss is mainly reduced via separation of inner versus outer actuator parameters. The found gain parameter p_1 in (b) is larger for the first (and third) actuator $u_{oc,1} = u_{oc,3}$ to counteract the cooling on boundaries B_W and B_E. In the same way, a smaller kurtosis parameter of $u_{oc,1} = u_{oc,3}$ in (c) leads to a wider bell shape of the input signals. The time shift parameter is close to 2 for both parameter sets in (b). The time of the peak input value of inner versus outer actuators differ for $\frac{T_{ff}}{p_{1,2}} - \frac{T_{ff}}{p_{2,2}} \approx 8$ seconds.

Figure 7.33: Input and output signals of the first and second actuator and sensor. The input signal of the outer actuators $u_{oc,1} = u_{oc,3}$ shows a higher peak value and a wider kurtosis in (a). The output signals in (b) track the reference function well until they reach the desired temperature and drop below this value afterwards.



(a) Time $t = 300$ seconds

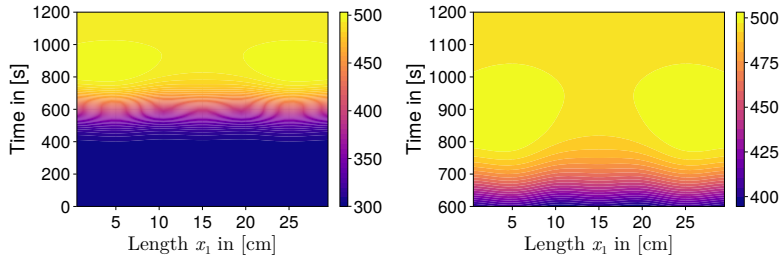
(b) Time $t = 600$ seconds



(c) Time $t = 900$ seconds

(d) Time $t = 1200$ seconds

Figure 7.34: Snapshots of the temperature distribution during the heating process. The warm areas in (b) and (c) illustrate the actuators' spatial characteristics. The regions close to boundary B_W and B_E are warmer than the central part in (c) because of a higher amount of supplied energy with the corresponding actuators. The thermal emissions force a cooling-down along the boundary sides B_W, B_E and B_N in the end of the heating-up phase, $t > 900$ seconds. The warmest region in (d) is close to the center of B_S because of the thermal insulation along this boundary side.



(a) Full time span

(b) Second half

Figure 7.35: Temperature distribution along boundary B_N during the heating-up phase. All temperatures are close to the initial value of 300 Kelvin until $t = 400$ seconds and increase notable in the time span $t \in (400, 800)$ seconds. The regions close to B_W and B_E reach the desired temperature $\Theta_{des} = 500$ Kelvin in (b) during $t \in (800, 1000)$ seconds.

The output signals in Fig. 7.33 only present a mean value of the true temperature on boundary B_N. Hence, we visualize the temperature distribution on B_N in Fig. 7.35. In contrast to the temperature distribution of the whole rectangle in Fig. 7.34, we find in Fig. 7.35 an almost uniform temperature transition on B_N. In the second part of the heating-up phase in Fig. 7.35 (b), we notice small temperature variations in space for $t \in (600, 800)$ and temperature peaks in a region close to B_W and B_E for $t \in (800, 1000)$. These peak values reach the desired temperature $\Theta_{des} = 500$ Kelvin and decrease for $t > 1000$ seconds because of the non-insulated boundary sides.

In this example, we showcased the complete feed-forward control design for the heating-up procedure. In the beginning, we simplified the thermal dynamics to a linear one-dim. model without thermal emissions and we designed a flatness-based control, which steers the output along a predefined reference signal. This flatness-based input is approximated by a bell-shaped parametrized function u_{oc}. These initial parameter values do not lead to sufficient reference tracking for the simplified model, see Fig. 7.28. Thus, we improve these parameters by approaching the supplied and emitted thermal energy, while we ignore the nonlinear thermal dynamics and the reference tracking. These considerations of the energy balance lead to a well temperature transition and the final output values

Scenario	**Parameters**		
	p_1	p_2	p_3
Approximation of FBC	11.815	2.070	9.212
Energy Supply	12.394	2.070	9.149
Optimization-based Design			
Actuator 1 & 3 (outer)	12.318	2.056	8.195
Actuator 2 (inner)	12.191	2.090	8.717

Table 7.8: Input Parameters of Feedforward Control Example.

almost match the desired temperature $\Theta_{des} = 500$, see Fig. 7.31. As the reference tracking is not included in the energy-based design, we return in the last step to the original thermal model and solve an optimization problem to include the reference tracking again. In this step, we find individual parameter sets p_1 and p_2 for the inner and outer actuators. In this example, we assumed a simple actuator and sensor setup and we obtained remarkable differences between the inner and outer actuator, see Fig. 7.32 and 7.34. In scenarios with more complex actuator and sensor setups, this last step of optimization-based reference tracking may be even more crucial for a well temperature transition. As we face thermal emissions, which force a cooling of the rectangle, we need to apply a feedback control to counteract this cooling and to stabilize the measured temperatures at the desired temperature. This concept is introduced in the next chapter.

8

Closed-Loop Control Design

In the previous chapter, we described the feed-forward control design to heat up the object and steer the measured temperatures along a predefined reference. After this heating-up procedure, we wish to stabilize the measured temperatures at the reached and desired value. Here, we need to counteract the cooling, which is driven by thermal emissions and forces the measured temperature to depart from the reference value. Hence, we return to the elementary physical situation and consider the balance of supplied versus emitted power as noted for the energy in Section 7.5. In particular, we seek for a control law that guarantees the equilibrium of supplied and emitted power as

$$0 = P_{in}(t) + P_{em}(t) \quad \text{for} \quad t > T_{ff}.$$

We know that the actuators need to supply the same amount of thermal power, which is emitted along the boundary sides as

$$\begin{aligned} P_{em}(t) &= \int_{\partial\Omega} \phi_{em}(t,x)dx \\ &= \int_{\partial\Omega} -h(x)\,[\vartheta(t,x) - \vartheta_{amb}(x)] - \sigma\,\varepsilon(x)\,\vartheta(t,x)^4 dx \end{aligned} \tag{8.1}$$

to hold the average temperature of the object on a constant level. As we are usually not able to measure temperatures on the entire surface, we are not able to determine the actual value of the emitted power P_{em}. We solve this issue with the same "approximation trick" as in Section 7.5: we replace the actual temperature $\vartheta(t,x)$ by the desired temperature Θ_{des} to yield the approximated emitted power

$$\tilde{P}_{em} = \int_{\partial\Omega} -h(x)\,[\Theta_{des} - \vartheta_{amb}(x)] - \sigma\,\varepsilon(x)\,\Theta_{des}^4 dx. \tag{8.2}$$

In the long run our feedback control shall drive the object's temperatures inside and on the boundary towards the desired value Θ_{des} such that the actual emitted power is leveling off and P_{em} approaches $\tilde{P}_{em}$. We find the necessary power supply according to Section 7.5 as

$$P_{in}(t) = \int_{\partial\Omega} \phi_{in}(t,x)dx = \sum_{n=1}^{N_u} \left(\int_{B_{in}} b_n(x)dx \right) u_n(t).$$

When the supplied and emitted power is in balance, then we have a constant power supply

$$\overline{P}_{in} = \sum_{n=1}^{N_u} \left(\int_{B_{in}} b_n(x)dx \right) \overline{u}_n, \tag{8.3}$$

with constant input signals $\lim_{t\to\infty} u_n(t) = \overline{u}_n$.

In this chapter, we realize the feedback control design with two common approaches. First of all, we introduce in Section 8.1 a state feedback with the linear-quadratic regulator (LQR) design and we show that the found static feedback law leads after some time to a balanced sum of supplied and emitted power with constant input signals as in Eq. (8.3). The LQR design provides a static state feedback and it is tailored for linear dynamical systems, but we desire a control with output feedback for our nonlinear heat conduction model as noted in Definition (6.1). Hence, we present in Section 8.2 an output feedback via model predictive control (MPC), which computes iteratively a new feedback law depending on the previous measurements. Accordingly, we find that this MPC approach stabilizes the thermal dynamics at the desired steady state.

8.1 Linear-Quadratic Regulator

The linear-quadratic regulator is a control design for linear dynamical systems with multiple input and output signals. On one hand, we obtain a common matrix-vector multiplication as feedback law with this technique. On the other hand, we need to solve an algebraic Riccati equation[1] numerically to yield this static feedback law and the complexity of this numerical solution scales with the system dimension. Moreover, this approach requires access to all system states, here temperatures, which are usually available with additional tools like state observers or Kalman filters.[2] The books [152, p. 7-28] and [153, p. 99, 211] present an introduction to LQR design and an extension for large-scale systems and partial differential equations is described in the book [154, p. 103, 107]. Furthermore, the LQR design for a two-dim. heat conduction in a time-discrete form is noted in the article [37].

[1] Jacopo Francesco Riccati (*1676, †1754) studied this type of equations, see [150].

[2] This filter is named after Rudolf Emil Kálmán (*1930, †2016), see [151].

In this section, we design the LQR control for the time-continuous linear heat conduction problem

$$\frac{d}{dt}\Theta(t) = A\,\Theta(t) + B\,u(t)$$

as described in Definition 6.1. The aim of a LQR design is to determine a feedback matrix $K \in \mathbb{R}^{N_u \times N_c}$, which is used in a full-state feedback to compute the input signals as

$$u(t) = -K\,\Theta(t).$$

We derive the closed-loop system when we identify input $u(t)$ in the linear heat conduction problem by the feedback law as

$$\begin{aligned}\frac{d}{dt}\Theta(t) &= A\,\Theta(t) - B\,K\Theta(t)\\ &= \underbrace{[A - B\,K]}_{=:A_{cl}}\Theta(t) = A_{cl}\,\Theta(t).\end{aligned}$$

The resulting system matrix of the closed-loop A_{cl} does not have the previous banded or Toeplitz-like shape of A, and all eigenvalues are smaller than zero. Hence, the closed-loop system approaches a steady-state as

$$A_{cl}\,\Theta(t) \to 0 \quad \text{and} \quad \Theta(t) \to 0.$$

As we wish to drive the temperatures towards a desired value Θ_{des} and not to zero, we consider the state feedback with offset as

$$u(t) = -K\ [\Theta(t) - \Theta_{des}]\,. \tag{8.4}$$

We visualize the procedure of the state feedback control for the heat conduction problem in Fig. 8.1.

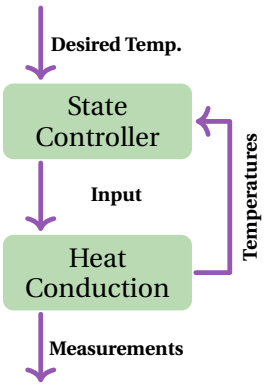


Figure 8.1: Scheme of state feedback control.

The LQR design exists for time-discrete and time-continuous systems, where the final optimal state shall be reached either on a finite or infinite time horizon. We choose the infinite horizon because this simplifies the computation of feedback matrix K. In this control technique, we wish to solve the optimization problem

$$\min\left\{J(u) = \int_0^\infty \Theta(t)^\top Q\Theta(t) + u(t)^\top R u(t) dt\right\} \tag{8.5}$$

with subject to the linear heat conduction problem (6.12) in a closed form. The matrices $Q \in \mathbb{R}^{N_c \times N_c}$ and $R \in \mathbb{R}^{N_u \times N_u}$ in Eq. (8.5) weigh the influence of states versus input signals in the resulting feedback law. These matrices must be positive definite and they are usually designed as diagonal matrices. A matrix $M \in \mathbb{R}^{N \times N}$ with $N > 0$ is called positive definite if the inequality $v^\top M v > 0$ holds for all vectors $v \in \mathbb{C}^N$. The speed of the closed-loop operation depends on choice of the matrix values: if $Q \gg R$ (element-wise), then we yield a fast operation and otherwise a slow or energy-efficient execution.

The closed-form solution of the optimal control problem (8.5) provides the feedback matrix as

$$K = R^{-1} B^\top\ P \tag{8.6}$$

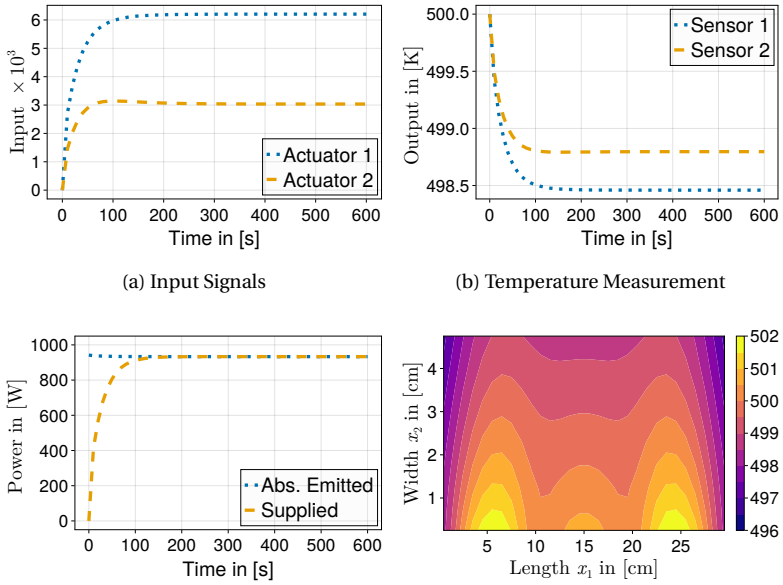
with matrix $P \in \mathbb{R}^{N_c \times N_c}$, which has to be found numerically by solving the algebraic Riccati equation for time-continuous dynamical systems

$$0 = Q + P\ A + A^\top\ P - P\ B\ R^{-1}\ B^\top P. \tag{8.7}$$

In Appendix A.2 we derive the feedback law (8.6) and the Riccati equation (8.7) from the optimal control problem (8.5). Mathematical software like the MATLAB functions *lqr* [155] for the LQR design and *icare* [156] to solve the the time-continuous algebraic Riccati equation offer well established tools to treat the LQR design. In the subsequent example, we find the solution of the algebraic Riccati equation with the JULIA library *MatrixEquations.jl* [157].

Example: Linear-Quadratic Regulation of 2-Dim. Heat Conduction

We return to the example in Section 7.6, where we consider a rectangle with length $L = 0.3$, width $W = 0.05$, density $\rho = 8000$ and specific heat capacity $c = 400$, and three actuators along boundary B_S. Here, we assume constant anisotropic thermal conductivity $(\lambda_1, \lambda_2) = (40, 60)\ \frac{W}{mK}$. We approximate the heat conduction problem in space and we yield a linear system as in Eq. (6.12). We do not include the thermal emissions (7.66) explicitly in our heat conduction model, instead we treat it as an (unknown) external disturbance here. Furthermore, we consider three ideal sensors as in Section 7.6 to evaluate temperatures on boundary B_N. We remind



(a) Input Signals

(b) Temperature Measurement

(c) Supplied and abs. Emitted Power

(d) Temperature Distribution

Figure 8.2: Simulation results of the LQR design. The input signals in (a) and the resulting temperature measurements in (b) settle after 200 seconds, because the supplied power compensate the emissions, see (c). The emitted power in (c) is noted as absolute values, $|P_{em}|$. The temperature distribution in (d) unveils a maximum variation of 6 Kelvin between actuators and the upper left and right corners.

that we do not include the sensors in the control design because the LQR technique is a state space approach and treats all temperatures. We build the state space system with A, B and we adjust the weighing matrices as

$$Q = 10^8 \, I_{N_c} \quad \text{and} \quad R = I_3$$

because we wish to reach the steady state quickly. The initial temperatures are set to 500 Kelvin because we are only interested in the stabilization at the desired value $\Theta_{des} = 500$ Kelvin. We compute the feedback matrix K by solving the algebraic Riccati equation (8.7) numerically, and we compute the input signals as in Eq. (8.4). We simulate the close-loop system for 600 seconds and we visualize our results in Fig. 8.2. The input signals in Fig. 8.2 (a) and temperature measurements in (b) converge in 200 seconds because the supplied power is able to compensate the thermal emissions for $t > 0$, see Fig. 8.2 (c). At the initial temperatures, we find the emitted power with Eq. (8.2) as $P_{em}(0) \approx -942$ Watt. These thermal emissions cause a temperature drop, see Fig. 8.2 (b), and consequently, the emitted power decreases to ca. $P_{em}(600) \approx -933$ Watt. Hence, we need to supply even a bit more energy in the balanced situation to reach the desired temperatures $\Theta_{des} = 500$ Kelvin exactly.

8.2 Model Predictive Control

In the previous section we designed a state feedback for the linear heat conduction. However, we do not have access to all states in general because we cannot measure temperatures inside the object. Hence, we design an output feedback in this section, which is also able to treat nonlinear systems. Model predictive control (MPC) is a well-established feedback approach and it is described in detail in several books, see e.g. [119, 120]. Moreover, we find examples of MPC approaches applied on the heat equation in the articles[3] [132–134], in the doctoral thesis [121, p. 51] and in our contribution [40]. Here, we consider in general the nonlinear spatially approximated heat conduction system as noted in Definition 6.1. The MPC approach is usually designed for sampled systems and so we convert the time-continuous state space (6.11) to a time-discrete one. We sample the remaining time interval $(T_{ff}, T_{final}]$ with $N_t \in \mathbb{N}_{>0}$ equidistant time steps

$$t_n = n\Delta T + t_0 \quad \text{with} \quad \Delta T = \frac{1}{N_t}\left[T_{final} - T_{ff}\right]$$

and for $n = 0$ we define $t_0 := T_{ff}$. The input signal is kept constantly from one step to the next one as $u(\tau) = u(t_n)$ for $\tau \in [t_n, t_{n+1})$. We apply an one-step integration method as described in Chapter 5 on Eq. (6.11) and we yield the time-discrete state space

$$\Theta(t_{n+1}) = \tilde{f}\left(\Theta(t_n), u(t_n), w(t_n, \Theta(t_n))\right), \tag{8.8a}$$

$$y(t_n) = C\Theta(t_n) \tag{8.8b}$$

in which $\tilde{f}$ describes the sampled right-hand side of Eq. (6.11a) including the sampling time ΔT.

The MPC routine is described in two nested iterations. The outer iteration describes a temporal behavior of the real system dynamics in each step $n \in \{0, 1, \ldots, N_t - 1\}$. The inner iteration contains a simulation of the system dynamics and an optimization routine to compute suitable constant input signals. The controller applies the found input signals on the simulation to predict the future states and it checks, whether the system dynamics behaves as desired. If suitable input signals are found, then the first input value, $u(t_n)$, is applied on the real system in the outer loop. The real system reacts on this input signal and we measure the output in the next step $y(t_{n+1})$, which is fed back to our controller to compute the input signal for the next iteration. This procedure is visualized in Fig. 8.2. Here, we remark that we need a state observer in real world experiments to update the states in the simulation with data from the measurement of the real system. In our examples, we assume that the simulation and the real system work identically. Inside the inner loop, we calculate the error between the desired temperature and the measurements as

$$e(t_n) := \Theta_{des} - y(t_n)$$

and the difference between subsequent input signals

$$\Delta u(t_n) := u(t_{n+1}) - u(t_n)$$

[3] Due to the wide range of heat conduction and diffusion models, e.g. with Dirichlet or Neumann boundary conditions, we find several different MPC approaches.

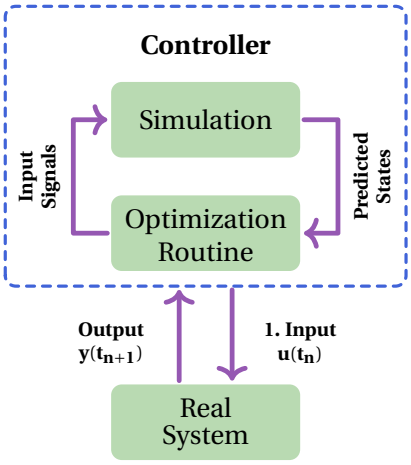


Figure 8.3: Scheme of model predictive control.

for $N_{mpc} \in \mathbb{N}_{>0}$ iterations, this number is also called *control horizon*. Here, we consider the state space (8.8) for both: the internal simulation and the real system. In practice both systems differ because we are not able to create an ideal model of the real process. The input signals of N_{mpc} inner iterations are found by solving the optimization problem

$$u^*(t_n), \ldots, u^*(t_{[n+N_{mpc}-1]}) = \arg\min_u \sum_{l=n}^{N_{mpc}-1} e(t_l)^\top Q\, e(t_l) + \sum_{l=n}^{N_{mpc}-2} \Delta u(t_l)^\top R\, \Delta u(t_l) \tag{8.9}$$

subject to Eq. (8.8) as the internal simulation model.

In Eq. (8.9) we consider the weighing matrices $Q \in \mathbb{R}^{N_y \times N_y}$ and $R \in \mathbb{R}^{N_u \times N_u}$ like for the linear-quadratic regulator design. We solve the optimization problem (8.9) and we apply the first input signal $u^*(t_n)$ on the real model (8.8). The remaining input values $u^*(t_{n+1}), \ldots, u^*(t_{[n+N_{mpc}-1]})$ are treated as initial values for the next MPC iteration step.

Adjusting the Sampling Time

In the MPC design, we face the task to choose a suitable sampling time ΔT. We know that heat conduction is a slow process and thermal energy needs some time to conduct from the actuators to sensors. This is an advantage here because the computation of input signals takes some time: we need to simulate and optimize a large scale system several times during one time step. However, we face in practice unknown external disturbances, which need to be rejected quickly. So, the sampling time should not be too long in accordance to receive quickly fresh measurements.

In Chapter 5, we discussed the numerical stability of integration methods and we found that explicit solvers like the forward Euler method require an upper limit of sampling time ΔT, while implicit solvers do not so. In Section 5.2, we stated that we implement our simulations with the implicit Runge-Kutta solver KenCarp5.

In the next step, we discuss the step and impulse response of the linear heat conduction to gain an idea of a suitable choice of ΔT. In Section 4.3, we note the solution of the linear system with a constant heat flux in Eq. (4.52). If we consider a one-dim. heat conduction with one actuator on B_W (left) and one sensor on B_E (right), then we note the output as

$$y(t) = C\Theta(t) = C\overline{V}^\top \exp(\tilde{A}_1 t)\overline{V}\Theta(0) + C\overline{V}^\top M(t)\overline{V}E_1 \frac{\Phi_1}{\Delta x_1}$$

with we have $C = (0, \ldots, 0, 1)$ and a constant heat flux Φ_1 as the step input. Furthermore, we yield the first derivative of the output as

$$\frac{d}{dt}y(t) = C\overline{V}^\top \tilde{A}_1 \exp(\tilde{A}_1 t)\overline{V}\Theta(0) + C\overline{V}^\top \exp(\tilde{A}_1 t)\overline{V}E_1 \frac{\Phi_1}{\Delta x_1}$$

where we have $\frac{d}{dt}M(t) = \exp(\tilde{A}_1 t)$. We evaluate the step response $y(t)$ and impulse response $\frac{d}{dt}y(t)$ with $\lambda = c = \rho = 1$, $\Delta x_1 = 10^{-2}$ and $N_j = 100$ nodes and we visualize the results in Fig. 8.4. As this heat conduction problem has no thermal emissions, we gain a pure temperature integration. In the beginning of the heating process, we obtain a small time lag

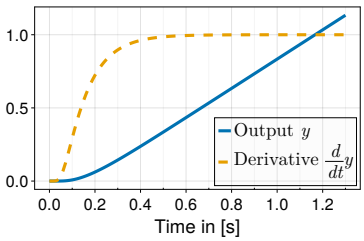


Figure 8.4: Step and impulse response of a one-dimensional linear heat conduction problem. The output $y(t)$ integrates the supplied heat flux $\Phi_1 = 1$. The integration has a time lag of ca. 0.2 seconds.

until ca. 0.2 seconds, where the integration is significantly below the ratio of one Kelvin per second. The reason of this time lag is the slow heat conduction from actuator to sensor. We can apply this finding on the choice of sampling time ΔT when we evaluate a step response.

We find a similar concept in the fundamentals of thermodynamics regarding the Fourier number

$$Fo(t) := \frac{\alpha}{\mathscr{L}^2}\, t$$

with diffusivity α, time $t \geq 0$ and characteristic length $\mathscr{L} \geq 0$, see also [49, p. 129]. The definition of the characteristic length depends on the geometry and the physical process, see also [137]. The dimensionless Fourier number qualifies in a heat conduction process as described above, whether enough time has passed to sense a noticeable temperature change or not, see [138, p. 69]. In case of very small Fourier numbers as $Fo(t) \ll 1$, we do not notice any temperature change and for $Fo(t) \geq 1$, we definitely yield significant values. In the previous example, see Fig. 8.4, we have $\alpha = 1\ \frac{m^2}{s}$ and a characteristic length $\mathscr{L} = L = N_j \Delta x_1 = 1$ meter and so we have $Fo(\tau) = \tau$. Hence, a sampling time $\Delta T = 1$ may guarantee a proper step response, but we can even admit lower values as long as $\Delta T > 0.2$ seconds. In this context, we see that $Fo < 0.2 \ll 1$.

Example: Model Predictive Control of 2-Dim. Heat Conduction

We consider the two-dim. example from Section 8.1 with three actuators on B_S, three ideal sensors on B_N and an initial temperature $\Theta(T_{ff}) = \Theta_{des} = 500$ Kelvin. We apply a step response with $u_1(t) = u_3(t) = 6000$ and $u_2(t) = 3000$ and we depict the simulation results in Fig. 8.5. We see that a sampling time $\Delta T = 30$ seconds provides a sufficient temperature change to compensate small thermal emissions. We set the number of iterations $N_t = 10$ and the control horizon $N_{mpc} = 3$. We emphasize that the choice of the control horizon may have a crucial impact on the closed-loop performance, this issue is analyzed in the doctoral thesis [121, p. 35]. Regarding the weighing matrices of errors Q and input signals R in the optimization problem (8.9), we have small errors compared to large input signals, e.g. $e_{n_1} \in (0, 10)$ and $\Delta u_{n_2} \in (10^3, 10^4)$ for the n_1-th sensor and n_2-th actuator. Thus, we specify the weighing matrices as

$$Q = I_{N_y} \quad \text{and} \quad R = \frac{10^{-7}}{N_u\,[N_{mpc} - 1]}\, I_{N_u}$$

to yield almost similar values for both: sensor and actuator weighing. Finally, we need to set an initial guess of the input signals $u(t_0)$ to start the optimization (8.9). We calculate the emitted power for the initial temperature $\Theta(0) = 500$ Kelvin with Eq. (8.2) as $P_{em}(0) \approx -942$ Watt. So in the long run, we need to supply the same positive value to yield a proper balance of emitted and supplied power. We consider the same initial guess for all three input signals and so we reformulate Eq. (8.3) to yield

$$\overline{u} = \frac{\overline{P}_{in}}{\sum_{n=1}^{N_u} \left(\int_{B_{in}} b_n(x)\, dx \right)} = \frac{\tilde{P}_{em}(0)}{\sum_{n=1}^{N_u} \left(\int_{B_{in}} b_n(x)\, dx \right)} \approx 5197.$$

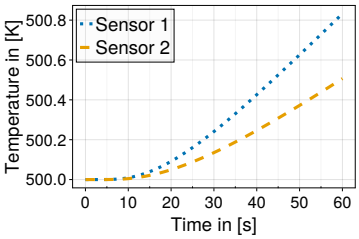
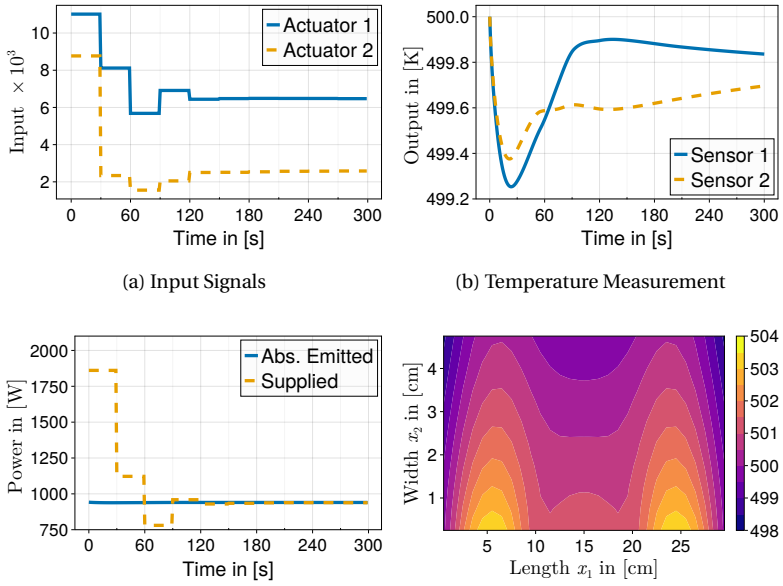


Figure 8.5: Step response of two-dim. heat conduction with $u_1(t) = u_3(t) = 6000$ and $u_2 = 3000$.



(a) Input Signals

(b) Temperature Measurement

(c) Supplied and abs. Emitted Power

(d) Temperature Distribution

Figure 8.6: Simulation results of the MPC design. The input signals in (a) and the resulting temperature measurements in (b) settle after 200 seconds, because the supplied power compensate the emissions, see (c). The emitted power in (c) is noted as absolute values, $|P_{em}|$. The temperature distribution in (d) unveils a maximum variation of 6 Kelvin between actuators and the upper left and right corners.

We implement the MPC routine with internal simulation and the external real system simulation as specified above, and we visualize the numerical results in Fig. 8.6. The input signal in Fig. 8.6 (a) starts with high values and converges in only five steps to $u_1 = u_3 \approx 6472$ and $u_2 = 2584$. This high initial input value is necessary to compensate the temperature drop in the first 60 seconds as depicted in Fig. 8.6 (b), and the measurement temperatures settle afterwards close to Θ_{des}. In Fig. 8.6 (c), we see that the supplied power compensates the emitted power after four steps precisely. We compare the temperature distribution in Fig. 8.6 (d) and in Fig. 8.2 (d) and we notice higher temperatures overall in case of the MPC design.

We summarize the findings of LQR and MPC design and we note that both approaches stabilize the measurements close to desired temperature. The linear-quadratic approach is easier to design and implement because we compute the static feedback matrix offline, but the performance close to the desired temperature is weak due to the static proportional gain. The model predictive approach requires more detailed work to specify the necessary control parameters, but it provides a good performance due to its prediction. Furthermore, we can apply the MPC design directly on our nonlinear heat conduction model as described in the next section. We find one drawback of model predictive control in case of real world applications. In such a case, we require all temperatures from the real system, e.g. with a state observer, to update the initial temperatures in the internal simulation after each step of the outer iteration, see Fig.8.2. A state observer design for a rapid thermal processing system is described in article [158].

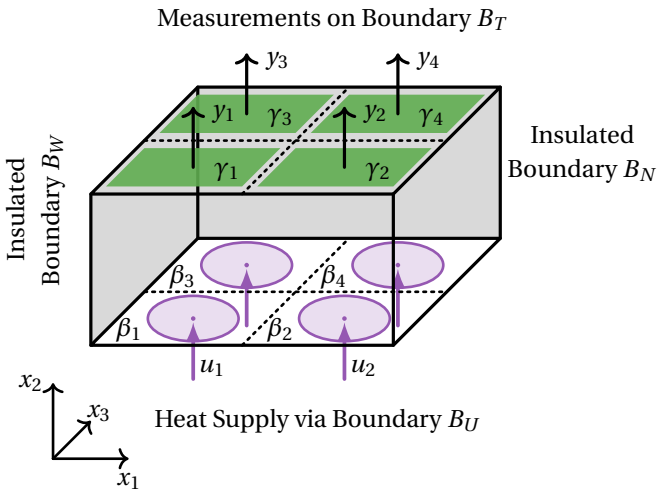


Figure 8.7: Model of a cuboid with four actuators on the underside B_U and four sensors on the topside B_T. The actuators and sensors are placed in a (2×2) checkerboard pattern. The boundary sides B_W, B_N (gray) and B_U are insulated for thermal emissions.

8.3 Simulation and Control of Heat Conduction in a Cuboid

We demonstrate the feed-forward and model predictive control design for a cuboid in this section. We portray the three-dim. model with actuators and sensors in Fig. 8.7. We consider the dimensions $L = W = 0.2$ meter, $H = 0.05$ meter and the material properties

$$\rho = 8000 \, \frac{kg}{m^3} \quad , \quad c = 400 \, \frac{J}{kgK}$$

and $\lambda(\theta) = \mathrm{diag}(\lambda_1(\theta), \lambda_1(\theta), \lambda_2(\theta))$ with

$$\lambda_1(\theta) \approx 1465 - 14.8\theta + 56.3 \cdot 10^{-3}\theta^2 - 93 \cdot 10^{-6}\theta^3 + 56.7 \cdot 10^{-9}\theta^4 \quad \text{and}$$
$$\lambda_2(\theta) \approx -2332 + 23\theta - 83 \cdot 10^{-3}\theta^2 + 133.3 \cdot 10^{-6}\theta^3 - 80 \cdot 10^{-9}\theta^4$$

as formulated in Section 7.6. The cuboid has six boundary sides and three of them are insulated for thermal emissions: B_W, B_N and B_U. The remaining boundaries, B_E, B_S and B_T, are open and we specify the thermal emissions as in Eq. (7.66). In Fig. 8.8 we depict the side view on boundary B_S. We assume four actuators on boundary B_U and four sensors on B_T, which are placed in a (2×2) checkerboard pattern. We specify the spatial characteristics of the actuators as

$$b_n(x) = \exp\left(\begin{pmatrix} 30 & \\ & 30 \end{pmatrix} [x - x_{c,n}]^4\right) \tag{8.10}$$

with central points

$$x_{c,n} \in \left\{ \begin{pmatrix} \frac{L}{4} \\ \frac{W}{4} \end{pmatrix}, \begin{pmatrix} \frac{3L}{4} \\ \frac{W}{4} \end{pmatrix}, \begin{pmatrix} \frac{L}{4} \\ \frac{3W}{4} \end{pmatrix}, \begin{pmatrix} \frac{3L}{4} \\ \frac{3W}{4} \end{pmatrix} \right\}$$

and the sensors as

$$y_n(t) = \frac{4}{L\,W} \int_{\gamma_n} \vartheta(t,x) dx$$

with sensor partitions

$$\gamma_1 = \left(0, \frac{L}{2}\right) \times \left(0, \frac{W}{2}\right) \times \{H\} \quad , \qquad \gamma_2 = \left(\frac{L}{2}, L\right) \times \left(0, \frac{W}{2}\right) \times \{H\},$$
$$\gamma_3 = \left(0, \frac{L}{2}\right) \times \left(\frac{W}{2}, W\right) \times \{H\} \quad \text{and} \qquad \gamma_4 = \left(\frac{L}{2}, L\right) \times \left(\frac{W}{2}, W\right) \times \{H\}.$$

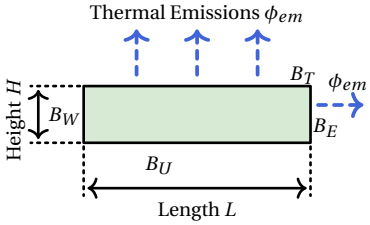


Figure 8.8: Side view of cuboid on boundary B_S with thermal emissions on boundary sides B_E and B_T.

Scenario	**Parameters**		
	p_1	p_2	p_3
Energy-based Optimization	13.065	2.070	9.076
Optimization-based Design			
Actuator 1 & 4	12.895	2.055	7.682
Actuator 2	12.967	2.054	8.301
Actuator 3	12.944	2.066	8.459

Table 8.1: Input Parameters for the Feed-Forward Control of the Three-Dimensional Example.

In the subsequent paragraphs, we apply concepts of feed-forward and feedback control design on the cuboid model.

Feed-forward Control

In the initial step, we apply the flatness-based control approach on the one-dim. model and we approximate the found input signal. We consider the same reference signal (7.68) and heat-up time $T_{ff} = 1200$ seconds as in Section 7.6. As we have the same reduced one-dim. model, we take the results from Section 7.6 as noted for the approximated input signal in the first row of Table 7.8. We continue with the energy-based optimization and we search for parameters p_1 (gain) and p_3 (kurtosis) such that the supplied energy E_{in} leads to a proper temperature transition. We wish to increase the temperature by 200 Kelvin and so we have a change of internal energy as

$$\Delta U = \rho\ c\ |\Omega_3| \Delta r = 1.28 \cdot 10^6 \quad \text{Joule}$$

with volume $|\Omega_3| = L\ W\ H = 2 \cdot 10^{-3}\ m^3$. The emitted thermal energy on boundaries B_E, B_S and B_T is approximated according to Eq. (7.62) as

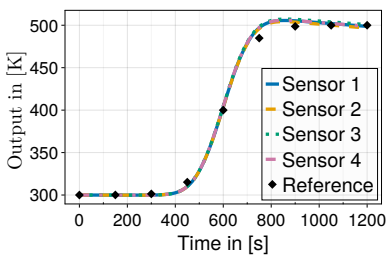
$$\tilde{E}_{em} \approx -85.70 \cdot 10^3 \quad \text{Joule.}$$

Summing up both quantities, we formulate and solve the optimization problem (7.69) with objective function

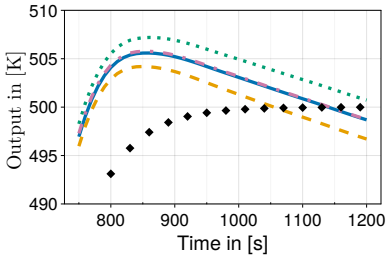
$$J(p_1, p_3) := \left[\Delta U - \tilde{E}_{em} - E_{oc}(p) \sum_{n=1}^{4} \left(\int_{\beta_n} b_n(x) dx \right) \right]^2$$

and parameterized input energy E_{oc} as in Eq. (7.58). The computed parameters p_1 and p_3 are listed in Table 8.1. We demonstrate the forced temperature transition with the found input parameters in Fig. 8.9. Here, we find in Fig. 8.9 (b) that the measured temperatures are noticeable above the reference values and we need to reduce these temperatures to match the desired reference.

We continue with the optimization-based reference tracking to decrease the distance between measured temperatures $y(t)$ and reference function $r(t)$. In the specification of the actuator positions, we find a symmetry for the actuation in segment β_1 and β_4, see Fig. 8.7. Hence, we reduce the parameter finding problem of originally four parameter sets to three and

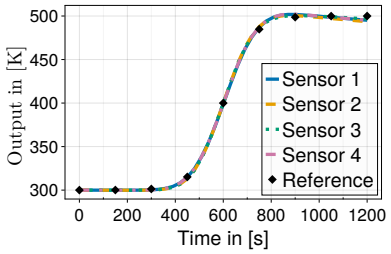


(a) Energy-based Input Design

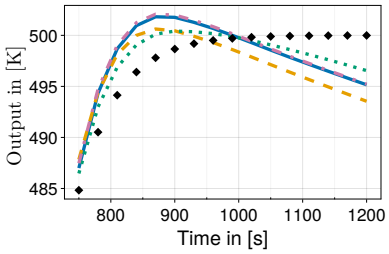


(b) Focus: Final Time

Figure 8.9: Energy-based optimization of input signals $u_{oc,n}$ for a cuboid example. The measured temperatures increase to the desired value $\Theta_{des} = 500$ Kelvin in (a), but they overshoot 500 Kelvin in (b) by more than 5 Kelvin.



(a) Optimization-based Reference Tracking



(b) Focus: Final Time

Figure 8.10: Optimization-based control for reference tracking applied on a cuboid example. The measured temperatures follow the reference values in (a) and the overshoot in (b) is reduced. However, the thermal losses lead to a temperature drop and the temperatures do not match exactly the reference.

we have the input signals

$$u(t) = \begin{pmatrix} u_{oc,1}(t,p_1) \\ u_{oc,2}(t,p_2) \\ u_{oc,3}(t,p_3) \\ u_{oc,4}(t,p_1) \end{pmatrix}$$

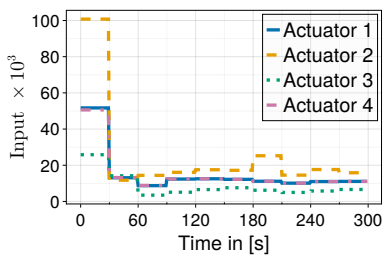
with $p_n = (p_{n,1}, p_{n,2}, p_{n,3})^\top$. We consider the error $e(t)$ between reference $r(t)$ and output $y(t)$ and we solve the minimization problem

$$(p_1^*, p_2^*, p_3^*) = \arg\min_{(p_1,p_2,p_3)} \frac{1}{T} \left\| \sum_{n=1}^{4} \mu_n \, e_n(t,p) \right\|_{L_2}^2$$

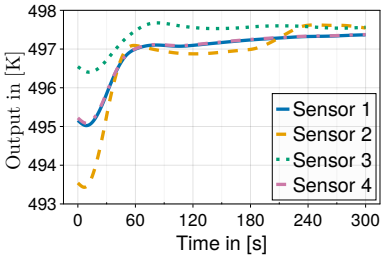
in which we assume $e_1(t,p) \equiv e_4(t,p)$ and we set $\mu_n = 1$. We note the computed parameters in Table 8.1 and we visualize the simulation results in Fig. 8.10. In Fig. 8.10 (b), we remark that the distance between the measured temperatures and the reference is reduced but the thermal losses force a temperature drop, which shall be compensated by a model predictive control approach in the next paragraph.

Feedback Control

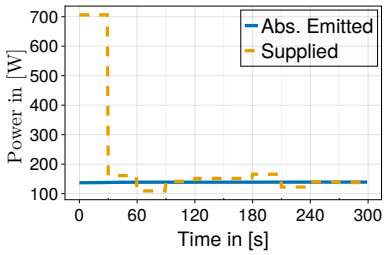
The feedback control shall stabilize the measured temperatures at the desired value $\Theta_{des} = 500$ Kelvin. We consider a model predictive control as feedback approach because it can be applied on nonlinear systems without the need of linearization. We choose a sampling time $\Delta T = 30$ seconds because it leads to proper temperature change for this scenario, see also Fig. 8.5. As the initial temperature of the feedback control is below the desired value, e.g. $y_n(T_{ff}) \approx 495$ Kelvin for $n \in \{1,2,3,4\}$, we need to consider the emitted power to stabilize the output value and an additional power to push the output temperatures closer to the reference value. We assume that entire cuboid has a temperature of 495 Kelvin and we find the change



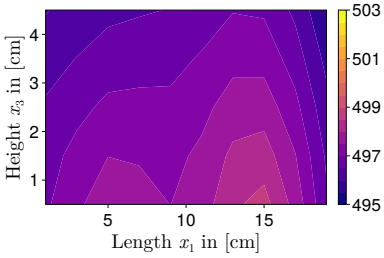
(a) Input Signals



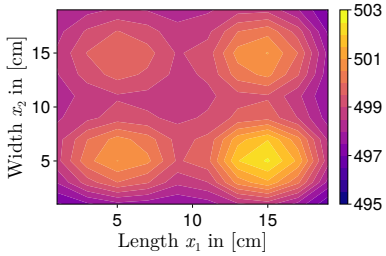
(b) Temperature Measurement



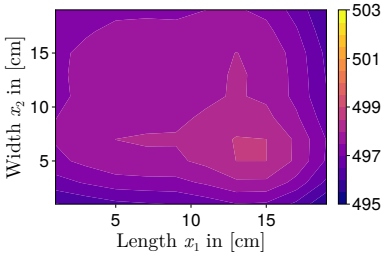
(c) Supplied and abs. Emitted Power



(d) Temp. Distribution on B_S



(e) Temp. Distribution on B_U



(f) Temp. Distribution on B_T

Figure 8.11: Model predictive control design for cuboid example. Actuator 2 has the highest input values in (a) because it is adjacent to boundaries B_E and B_S, where we have thermal emissions. Actuator 3 has the lowest input values because it is adjacent to the insulated boundaries B_W and B_N. Actuator 1 and 4 have almost equal values because their situation is symmetric. The measured temperatures increase in (b) to approx. 497.5 Kelvin and stay at this value. The supplied power in (c) starts at a high value to compensate the temperature error of five Kelvin and it approaches the amount of the absolute emitted power after few iterations. The temperature distributions at $t = T_{final}$ in (d), (e), (f) show the impact of spatial characteristics on the forced temperature evolution. The region of the highest temperatures is close to boundaries B_E and B_S.

of internal energy as

$$\Delta U = \rho\, c\, |\Omega_3|(500-495) = 32\cdot 10^3 \quad \text{Joule.}$$

We wish that this energy shall be supplied in the first iteration and accordingly, we find the additional power as $P_{add} = \frac{\Delta U}{\Delta T} \approx 1066.67$ Watt. We remark that this additional power is just an approximated value because the temperatures are not at 495 Kelvin in the whole cuboid. To stabilize the output measurements at the desired temperature, we need to compensate the thermal emission and we find the emitted power for $\Theta_{des} = 500$ Kelvin as

$$\begin{aligned}\tilde{P}_{em} &= \left[\underbrace{L\,H}_{B_S} + \underbrace{W\,H}_{B_E} + \underbrace{L\,W}_{B_T}\right]\left[-h(x)\,[\Theta_{des}-\vartheta_{amb}] - \sigma\,\varepsilon\,\Theta_{des}^4\right] \\ &\approx -141 \quad \text{Watt.}\end{aligned}$$

Hence, we need to apply input signals in the beginning of the MPC run with an average value of

$$\overline{u} = \frac{P_{add} + |\tilde{P}_{em}|}{\sum_{n=1}^{N_u}\left(\int_{B_{in}} b_n(x)dx\right)} \approx 98\cdot 10^3$$

and later this value shall converge towards the average value of

$$\overline{u} = \frac{|\tilde{P}_{em}|}{\sum_{n=1}^{N_u} \left(\int_{B_{in}} b_n(x) dx \right)} \approx 11.5 \cdot 10^3$$

We set the initial guess of the input values for the MPC optimization routine (8.9) in accordance with these ideas: in the first iteration we need to set a high input value $u_n = 98 \cdot 10^3$ and in the remaining iterations we have low input values, $u_n = 11.5 \cdot 10^3$. We design the objective function in the optimization routine (8.9) with the weighing matrices

$$Q = \frac{1}{N_y \, [N_{mpc} + 1]} I_{N_y} \quad \text{and} \quad R = \frac{10^{-8}}{N_u \, [N_{mpc} - 1]} I_{N_u}$$

to yield similar parts for the impact of measurement errors $e(t_n)$ and the input differences $\Delta u(t_n)$. We run the MPC routine for $N_t = 10$ iterations and we visualize our results in Fig. 8.11.

The input signals in Fig. 8.11 (a) start at a high level and approach after a couple of iterations almost constant values close to the expected average value $\overline{u} = 11.5 \cdot 10^3$. The supplied power acts analog to the input values and approaches the absolute value of the emitted power $|\tilde{P}_{em}| \approx 141$ Watt in Fig. 8.11 (c) in the long run. The temperature measurements in Fig. 8.11 (b) rise and they are stabilized but they do not reach the desired value $\Theta_{des} = 500$ Kelvin. So, we need to supply a higher power value to minimize this steady-state error. The temperature distributions in Fig. 8.11 (d), (e), (f) are snapshots at the final time $T_{final} = T_{ff} + 300 = 1500$ seconds and they unveil the significant influence of the actuator's spatial characteristics on the thermal treatment. Finally, we portray the temperatures inside the cuboid at the final time T_{final} in Fig. 8.12.

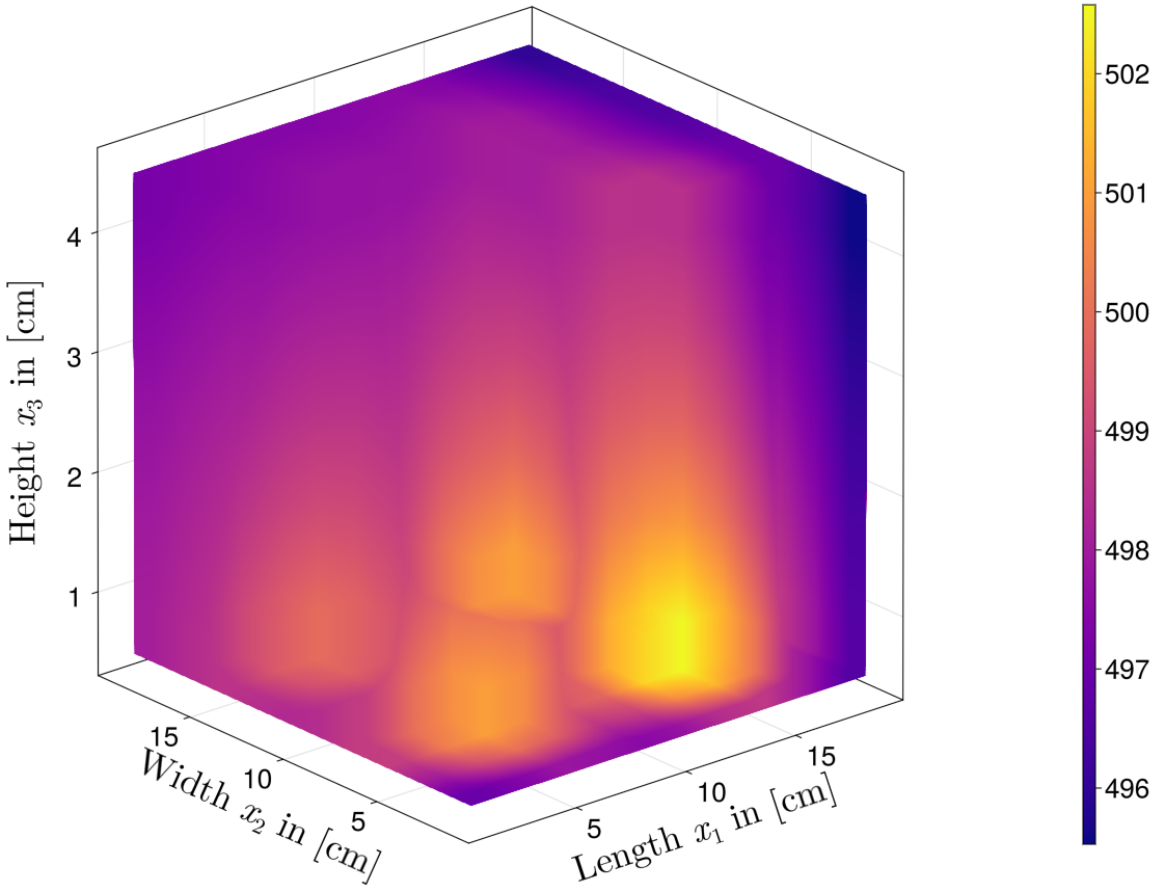


Figure 8.12: Temperatures inside the cuboid at the final time T_{final} = 1500 seconds.

Epilogue

9

Conclusion and Future Work

In this theses, we developed a mathematical framework, which connects the modeling and simulation of heat conduction with the control design via multiple actuators and sensors. In each part of this work we note connections to related topics in order to give an idea about the wide field of research on heat transfer problems. In this chapter, we present a selection of related topics and we discuss how they can improve our proposed heat conduction framework.

In the simulation of technical systems, which are described by partial differential equations, we find the significant issue that computational costs increase by the number of spatially approximated states. In particular the size of an approximated system may grow quadratically for a two-dim. and cubically for a three-dim. geometry. To solve this issue, we can reduce the system size before the computation using model order reduction methods, e.g. proper orthogonal decomposition, and we can accelerate the matrix-vector operations during the simulation with parallel computing, e.g. using graphics processing units (GPU). The scientific field of model order reduction provides a wide range of well-established approaches for PDE and common state space models. These approaches are described in the literature, see the book [159] and they are implemented as software libraries, see e.g. [160, 161]. In case of simple geometries like rectangles or cuboids, these approaches may perform very well, but we need to take care about the boundary sides to maintain the spatial characteristics of actuators and sensors with a minimum loss of information. This issue is crucial to yield a proper evaluation of supplied heat and temperature measurements. When we concern the hardware, we have a fast development of GPU, which comes along with recent needs in the domain of computer graphics and artificial intelligence. GPU-based computational methods are also applied on problems in scientific computing to solve PDE, see e.g. [162, 163], and additionally we find applications in model predictive control, see [164]. One major advantage of GPU approaches is the fast operation of linear algebra methods on large matrices. Hence, we may apply GPU methods to solve the linear spatially approximated heat conduction problems in Section 3.4 and Chapter 4.

The recent developments in artificial intelligence also enforce new connections between scientific computing and machine learning. One of these branches is known as *Scientific Machine Learning* (SciML), which focuses on computational methods to improve scientific models with data-based approaches and machine learning techniques. In particular, real data from lab experiments can enhance SciML models dramatically. We refer to the website [165] for an introduction and we find related SciML software libraries for the Julia programming language on the website [166]. Next, we briefly present two SciML approaches, Physics-Informed Neural Networks and Dynamic Mode Decomposition, which provide powerful tools to improve the modeling of processes.

Neural networks are popular techniques in machine learning for classification and regression purposes. They are extended for the modeling and simulation of physical systems as Physics-Informed Neural Networks (PINN)[1], see [60–63]. The input layer receives spatial coordinates, e.g. (x_1, x_2, x_3), and time t to compute the states, e.g. temperature $\vartheta(t,x)$, in the output layer. The objective function of this neural network type contains the considered PDE, including initial and boundary conditions, and possibly the evaluation of errors between computed states and experimental data. The PDE derivatives are realized with algorithmic differentiation, see [148,149]. This PINN approach might be very helpful in scenario where we have a good model of the actual process but additional uncertainties like unknown parameters or external influences on the process. In the sense of our thermal dynamics, we assume to have a perfect geometry state several assumptions regarding a perfect geometry, known material properties and thermal emissions. Furthermore, we neglect close or adjacent objects in the object's surrounding. In real experiments we cannot assure these assumptions and so we may improve the thermal model with a PINN approach and experimental data. However, one drawback of PINN and neural networks in general is the large size of the network architecturere. This situation leads to high computational costs[2] and a weak understanding of learning process.[3]

[1] We mentioned PINN in the beginning of Chapter 3 as an alternative to the proposed finite volume approach.

[2] Here we refer to the parallel computing approaches mentioned above.

[3] This issue is discussed in the field of *Explainable Artificial Intelligence*.

Another vibrant field of SciML was established in computational fluid dynamics: Dynamic Mode Decomposition (DMD), see the article [167]. This approach is used to compute a time-discrete mapping $f : \mathbb{R}^N \to \mathbb{R}^N$ with known data snapshots z as in

$$z(t_{n+1}) = f\left(z(t_n)\right).$$

The basic concept was proposed for linear systems where standard methods from linear algebra, e.g. singular value decomposition, are applied to recover a linear mapping $f(z) = Az$. When we transfer this concept to our framework, then we find the linear operator as the system matrix A_{N_d} and the states as temperature Θ, see Section 3.4 and Chapter 4. Hence, we are able to reconstruct the diffusive behavior from known temperature data via DMD. This basic DMD approach was extended in many directions, e.g. for systems with inputs in [168] and physics-informed DMD in [169], and so we find promising interfaces to our heat conduction framework. Recent contributions of the author focus on DMD approaches for systems with structured system matrix as described in Section 3.4, see [42,43].

In a nutshell, the described heat conduction framework provides links to various modern and auspicious fields of research. In future work, the extensions should improve the computation of the thermal dynamics and they should include data-based approaches to enhance the practicality for real-world scenarios like industrial applications.

A

Mathematical Fundamentals

A.1 Analytical Solution of the Heat Equation

In this section, we derive an analytical solution of the one-dimensional heat equation. Firstly, we describe the case of Neumann boundary conditions and secondly, we discuss briefly the case of Dirichlet boundary data. We utilize a separation of variables approach, which is a well known technique in the literature, see e.g. [4, p. 75], [170, p. 124] and [171]. We assume the one-dim. heat equation with length $L > 0$ constant material properties: $\lambda > 0$, $c > 0$, $\rho > 0$, and we note them as diffusivity $\alpha = \frac{\lambda}{c\,\rho}$. Accordingly, we consider the linear heat equation (2.21) as

$$\frac{\partial}{\partial t}\vartheta(t,x) = \alpha \frac{\partial^2}{\partial x^2}\vartheta(t,x) \tag{A.1}$$

for $(t,x) \in (0,T) \times (0,L)$ with initial condition

$$\vartheta(0,x) = \vartheta_0(x) := p\; x\; (L-x) \tag{A.2}$$

and scaling $p > 0$. We assume thermally insulated boundary sides and note the boundary conditions as

$$\frac{\partial}{\partial x}\vartheta(\cdot,x)\Big|_{x=0} \cdot \vec{n}(x) = 0 \quad \text{and} \quad \frac{\partial}{\partial x}\vartheta(\cdot,x)\Big|_{x=L} \cdot \vec{n}(x) = 0 \tag{A.3}$$

with outer normal vector $\vec{n}(0) = -1$ on the left and $\vec{n}(L) = +1$ on the right boundary. We assume a separation of variables as

$$\vartheta(t,x) = f(t)\; g(x)$$

in Eq. (A.1) to separate the temporal and spatial dynamics as

$$\frac{d}{dt}f(t)g(x) = \alpha f(t)\frac{d^2}{d^2x}g(x)$$

or equivalently

$$\frac{\dot{f}(t)}{\alpha f(t)} = \frac{g''(x)}{g(x)} = \mu.$$

We find the solution of the first-order differential equation $\dot{f}(t) = \mu\; \alpha f(t)$ as

$$f(t) = \exp(\mu\; \alpha\; t)\; f(0), \tag{A.4}$$

where we set $f(0) = 1$. In the next step, we solve the second-order differential equation

$$\frac{d^2}{d^2x} g(x) = \mu\, g(x) \tag{A.5}$$

and we notice that μ determines the solution $g(x)$. Hence, we have to discuss three cases $\mu \equiv 0$, $\mu > 0$ and $\mu < 0$.

1. If $\mu \equiv 0$, then Eq. (A.5) is simplified as $\frac{d^2}{d^2x} g(x) = 0$ and we yield

$$g(x) = c_1 x + c_0.$$

We evaluate the boundary condition (A.3) as $\frac{d}{dx} g(x) = 0$ for $x = 0$ and $x = L$ and we find $c_1 \equiv 0$. Thus, the solution of Eq. (A.5) for $\mu \equiv 0$ is $g(x) = c_0$, but this is not possible due to the initial conditions. Therefore, $\mu \equiv 0$ is not a possible value.

2. If $\mu > 0$, then we may assume the solution[1]

$$g(x) = c_1 \exp(\sqrt{\mu}x) + c_2 \exp(-\sqrt{\mu}x)$$

and its first derivative

$$\frac{d}{dx} g(x) = \sqrt{\mu} c_1 \exp(\sqrt{\mu}x) - \sqrt{\mu} c_2 \exp(-\sqrt{\mu}x).$$

We find the boundary conditions (A.3) the linear system of equations

$$\frac{d}{dx} \begin{pmatrix} g(x)\big|_{x=0} \\ g(x)\big|_{x=L} \end{pmatrix} = \begin{pmatrix} \sqrt{\mu} & -\sqrt{\mu} \\ \sqrt{\mu}\, e^{\sqrt{\mu}L} & -\sqrt{\mu}\, e^{-\sqrt{\mu}L} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

which is solved with the inverse matrix as

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \frac{1}{\mu\,(-e^{-\sqrt{\mu}L} + e^{\sqrt{\mu}L})} \begin{pmatrix} -\sqrt{\mu}\, e^{-\sqrt{\mu}L} & \sqrt{\mu} \\ -\sqrt{\mu}\, e^{\sqrt{\mu}L} & \sqrt{\mu} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{A.6}$$

and so we find $c_1 = c_2 \equiv 0$. We note that the expression

$$\frac{1}{\mu\,(-e^{-\sqrt{\mu}L} + e^{\sqrt{\mu}L})} \neq 0$$

for all μ and L. As we cannot find any $\mu > 0$, which solves Eq. (A.6), we need to exclude this approach as a candidate solution. Additionally, we also ignore the approach

$$g(x) = c_1 \exp\left(\sqrt{-\mu}x\right) + c_2 \exp\left(-\sqrt{-\mu}x\right)$$

because it leads to the wrong second-order differential equation

$$\frac{d^2}{dx^2} g(x) = -\mu g(x).$$

3. In the last case $\mu < 0$, we assume the solution of Eq. (A.5) as

$$g(x) = c_1 \sin(\sqrt{-\mu}x) + c_2 \cos(\sqrt{-\mu}x) \tag{A.7}$$

and we calculate with the first derivative

$$\frac{d}{dx} g(x) = \sqrt{-\mu}\, \left[c_1 \cos(\sqrt{-\mu}x) - c_2 \sin(\sqrt{-\mu}x)\right].$$

[1] If $c_1 = c_2$ then we may write $g(x) = c_1 \cosh(\sqrt{\mu}x)$ and if $c_1 = -c_2$ then we have $g(x) = c_1 \sinh(\sqrt{\mu}x)$.

We yield the boundary conditions as

$$\frac{d}{dx}g(x=0) = -\sqrt{-\mu}\, c_1 = 0,$$

which implies $c_1 = 0$, and

$$\frac{d}{dx}g(x=L) = -c_2\sqrt{-\mu}\sin(\sqrt{-\mu}L) = 0.$$

We notice that this approach offers a suitable solution if the expression

$$-\sqrt{-\mu}\sin(\sqrt{-\mu}L) = 0$$

is guaranteed for certain values of μ. We find these roots as $\mu = -\left[\frac{n\pi}{L}\right]^2$ with $n \in \{0, 1, \ldots, \infty\}$, see Fig. A.1.

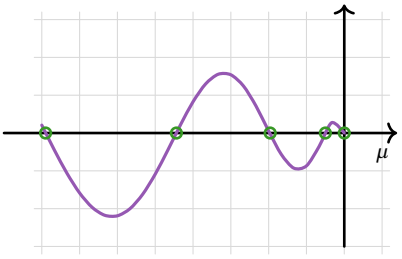


Figure A.1: Eigenvalues of analytical heat equation. Function graph of $-\sqrt{-\mu}\sin(\sqrt{-\mu}L)$ with roots $\mu = -\left[\frac{n\pi}{L}\right]^2$ for $n \in \{0, 1, \ldots, 4\}$.

We conclude from these calculations that $\alpha\mu_n = -\alpha\left[\frac{n\pi}{L}\right]^2$ are the eigenvalues and

$$\varphi_n(x) = c_{2,n}\cos(\sqrt{-\mu_n}x) = c_{2,n}\cos\left(n\,\pi\,\frac{x}{L}\right)$$

are the eigenvectors[2] of the linear heat equation (A.1). In the next steps, we find the coefficients $c_{2,n}$. We know that eigenvectors span an orthonormal basis and thus we calculate the inner product of the function space $L^2((0,L),\mathbb{R})$ as

$$\langle\varphi_n, \varphi_m\rangle := \int_0^L c_{2,n}\cos\left(n\,\pi\,\frac{x}{L}\right)\, c_{2,m}\cos\left(m\,\pi\,\frac{x}{L}\right)dx \;=\; \delta_{i,j}$$

with $\delta_{m,n} = \begin{cases} 1 & \text{for} \quad n = m, \\ 0 & \text{otherwise.} \end{cases}$ In case of equality $n = m$, we find

$$\langle\varphi_n, \varphi_n\rangle \;=\; c_{2,n}^2 \int_0^L \cos\left(n\,\pi\,\frac{x}{L}\right)^2 dx = c_{2,n}^2\,\frac{L}{2} \stackrel{!}{=} 1$$

for $n > 0$ and thus we have $c_{2,n} = \sqrt{\frac{2}{L}}$, and for $n = 0$ we find $c_{2,0} = \frac{1}{\sqrt{L}}$. Otherwise $n \neq m$ the integral vanishes as

$$\int_0^L \cos\left(n\,\pi\,\frac{x}{L}\right)\,\cos\left(m\,\pi\,\frac{x}{L}\right)dx \equiv 0$$

for all $x \in [0, L]$ and so we note the orthonormal eigenvectors

$$\varphi_0(x) = \frac{1}{\sqrt{L}} \quad \text{and} \quad \varphi_n(x) = \sqrt{\frac{2}{L}}\cos\left(n\,\pi\,\frac{x}{L}\right) \text{ for } n > 0. \tag{A.8}$$

The principle of superposition provides us a general solution of the heat equation (A.1) as

$$\begin{aligned}\vartheta(t,x) &= \tilde{c}_0 + \sum_{n=1}^{\infty} \tilde{c}_n\, f_n(t)\, \varphi_n(x) \\ &= \tilde{c}_0 + \sum_{n=1}^{\infty} \tilde{c}_n\, \exp\left(\alpha\,\mu_n\,t\right)\sqrt{\frac{2}{L}}\cos\left(n\pi\frac{x}{L}\right)\end{aligned} \tag{A.9}$$

where the coefficients $\tilde{c}_0$ and $\tilde{c}_n$ describe the behavior at the initial time, see Eq. (A.2). These coefficients are computed in the following steps. At

[2] They are also known as eigenfunctions.

the initial time $t = 0$ we know that $f_n(0) = 1$ for all $n > 0$ in Eq. (A.9) and so we have

$$\vartheta(0,x) = \vartheta_0(x) = \tilde{c}_0 + \sum_{n=1}^{\infty} \tilde{c}_n \, \varphi_n(x). \tag{A.10}$$

We multiply Eq. (A.10) with the eigenvector φ_n on both sides and apply the inner product as

$$\langle \vartheta_0, \varphi_n \rangle = \tilde{c}_0 \underbrace{\langle 1, \varphi_n \rangle}_{n=0} + \tilde{c}_n \underbrace{\langle \varphi_n, \varphi_n \rangle}_{n>0}, \tag{A.11}$$

which is distinguished as

$$\langle \vartheta_0, \varphi_0 \rangle = \tilde{c}_0 \langle 1, \varphi_0 \rangle \qquad \text{for } n = 0 \text{ and} \tag{A.12}$$

$$\langle \vartheta_0, \varphi_n \rangle = \tilde{c}_n \langle \varphi_n, \varphi_n \rangle \qquad \text{for } n > 0\,. \tag{A.13}$$

We find the left-hand side of Eq. (A.12) as

$$\begin{aligned} \langle \vartheta_0, \varphi_0 \rangle &= \varphi_0 \int_0^L \vartheta_0(x)\,dx = \frac{1}{\sqrt{L}} \int_0^L p x (L-x)\,dx \\ &= \frac{1}{\sqrt{L}}\, p \left[\frac{L}{2} x^2 - \frac{1}{3} x^3 \right]_0^L = \frac{1}{\sqrt{L}}\, p \frac{L^3}{6} \end{aligned}$$

and the right-hand side of Eq. (A.12) as

$$\tilde{c}_0 \langle 1, \varphi_0 \rangle = \tilde{c}_0 \, \varphi_0 \int_0^L 1\,dx = \tilde{c}_0 \frac{1}{\sqrt{L}} L.$$

Hence, we compute the coefficient $\tilde{c}_0 = p\frac{L^2}{6}$. In case of the second equation (A.13), we know that $\langle \varphi_n, \varphi_n \rangle = 1$ and we reduce our calculations as

$$\begin{aligned} \tilde{c}_n = \langle \vartheta_0, \varphi_n \rangle &= \int_0^L \varphi_n(x)\, \vartheta_0(x)\,dx \\ &= \sqrt{\frac{2}{L}} \int_0^L \left[p\, x\, (L-x) \right] \cos\left(n\pi \frac{x}{L} \right) dx \\ &= -\sqrt{\frac{2}{L}}\, p\, L \left(\frac{L}{n\pi} \right)^2 \left[(-1)^n + 1 \right]. \end{aligned}$$

We see that

$$(-1)^n + 1 = \begin{cases} 2 & \text{if} \quad n \text{ is even,} \\ 0 & \text{if} \quad n \text{ is odd} \end{cases}$$

and we specify the coefficients

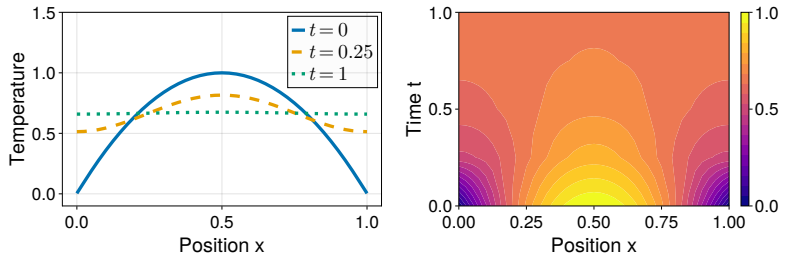
$$\tilde{c}_n = \begin{cases} -2\sqrt{\frac{2}{L}}\, p\, L \left(\frac{L}{n\pi} \right)^2 & \text{if} \quad n \text{ is even,} \\ 0 & \text{if} \quad n \text{ is odd.} \end{cases}$$

We identify the coefficients $\tilde{c}_0$ and $\tilde{c}_n$ with $n > 0$ in Eq. (A.9) and we yield

$$\vartheta(t,x) = p\frac{L^2}{6} - 2p\, \frac{L^2}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{n^2} \left[(-1)^i + 1 \right] \exp\left(-\alpha \left(\frac{n\pi}{L} \right)^2 t \right) \cos\left(n\pi \frac{x}{L} \right).$$

Finally, we consider only even indices as $n = 2k$ and we note the analytical solution of the one-dim. linear heat equation with initial temperature in Eq. (A.2) as

$$\vartheta(t,x) = p\frac{L^2}{6} - p\, \frac{L^2}{\pi^2} \sum_{k=1}^{\infty} \frac{1}{k^2} \exp\left(-\alpha\, 4 \left[\frac{k\pi}{L} \right]^2 t \right) \cos\left(2\, k\, \pi \frac{x}{L} \right). \tag{A.14}$$



(a) Temperature Distribution

(b) Temperature Evolution

Figure A.2: Simulation of the heat equation with Neumann boundary conditions.The scaling of the initial temperature is $p = 4$. The computed temperature converges towards the mean value of the initial temperature distribution.

We evaluate the found solution (A.14) for an example with length $L = 1$, diffusivity $\alpha = 0.1$ and scaling $p = 4$, where we neglect physical units. We compute the solution (A.14) for $k \in \{1, \ldots, 100\}$ and we visualize the computed data in Fig. A.2. In Fig. A.2 (a) we see that the temperature is converging towards the mean value of the initial temperature distribution

$$\overline{\vartheta}_0 = \frac{1}{L}\int_0^L 4x\,[L-x]\,dx = \frac{2}{3}L^2 = \frac{2}{3}.$$

In Fig. A.2 (b), we notice the continuous transition of the temperature values and we find the temperatures rise close to the boundary sides.

Side Note: Relations to the Basel Problem

At $(t, x) = (0, 0)$ we yield for Eq. (A.14)

$$\vartheta(0,0) = \tilde{c}_0 - \hat{c}\sum_{k=1}^{\infty}\frac{1}{k^2} \tag{A.15}$$

with $\hat{c} = p\,\frac{L^2}{\pi^2}$. The series $\sum_{k=1}^{\infty}\frac{1}{k^2}$ equals the Riemann Zeta function

$$\zeta(s) = \sum_{n=1}^{\infty}\frac{1}{n^s}$$

for $s = 2$, and the exact calculation of series $\sum_{k=1}^{\infty}\frac{1}{k^2}$ is known as the "Basel problem". According to Leonhard Euler, we note the series

$$\sum_{k=1}^{\infty}\frac{1}{k^2} = \frac{\pi^2}{6}$$

and we find coefficient $\tilde{c}_0$ in Eq. (A.15) as

$$\begin{aligned}\tilde{c}_0 &= \vartheta(0,0) + \hat{c}\sum_{k=1}^{\infty}\frac{1}{k^2} = \vartheta(0,0) + \hat{c}\,\frac{\pi^2}{6}\\ &= \vartheta(0,0) + p\,\frac{L^2}{\pi^2}\,\frac{\pi^2}{6} = \vartheta(0,0) + p\,\frac{L^2}{6}.\end{aligned}$$

Heat Equation with zero-Dirichlet Boundary Conditions

In this paragraph, we compare the previous results with a heat equation, which is equipped with a Dirichlet boundary condition. This means that a temperature data - instead of a gradient - along all boundary sides is fixed. As we do not assume Dirichlet conditions in this thesis, we only state briefly the differences to our previous case with Neumann conditions. We consider the heat equation (A.1) with fixed temperatures as

$$\vartheta(\cdot,0) = 0 \quad \text{and} \quad \vartheta(\cdot,L) = 0. \tag{A.16}$$

We split again the temporal and spatial terms, $f(t)$ and $g(x)$, we note the temporal term as in Eq. (A.4) and we consider the spatial term as in Eq. (A.7). Here, we apply the Dirichlet boundary conditions and we calculate

$$g(x=0) = c_2 = 0$$

on the left side and

$$g(x=L) = c_1 \, \sin\left(\sqrt{-\mu}L\right) = 0 \tag{A.17}$$

on the right side. As we assume $c_1 \neq 0$, we know that

$$\mu = -\left[\frac{n\,\pi}{L}\right]^2$$

fulfills Eq. (A.17). Hence, we find the eigenvectors as

$$\varphi_n(x) = c_{1,n} \, \sin\left(n\pi\frac{x}{L}\right),$$

where we have $\varphi_0(x) = 0$. We evaluate the inner product

$$\langle \varphi_n, \varphi_m \rangle := \int_0^L c_{1,n} \sin\left(n\,\pi\,\frac{x}{L}\right) \, c_{1,m} \sin\left(m\,\pi\,\frac{x}{L}\right) dx \, = \, \delta_{i,j}$$

and we yield the coefficients $c_{1,n} = \sqrt{\frac{2}{L}}$ for $n = m$. We formulate a preliminary version of the analytical solution as

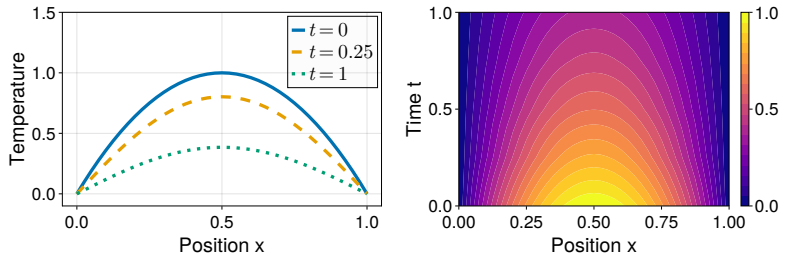
$$\begin{aligned}
\vartheta(t,x) &= \sum_{n=1}^{\infty} \tilde{c}_n \, \varphi_n(x) \, \exp\left(-\alpha\left[\frac{n\pi}{L}\right]^2 \, t\right) \\
&= \sum_{n=1}^{\infty} \tilde{c}_n \sqrt{\frac{2}{L}} \, \sin\left(n\pi\frac{x}{L}\right) \, \exp\left(-\alpha\left[\frac{n\pi}{L}\right]^2 \, t\right)
\end{aligned} \tag{A.18}$$

in which we need to determine the coefficients $\tilde{c}_n$ via the initial temperature distribution in Eq. (A.2). For this purpose, we need to solve Eq. (A.13) as

$$\begin{aligned}
\tilde{c}_n = \langle \vartheta_0, \varphi_n \rangle &= \int_0^L \varphi_n(x) \, \vartheta_0(x) dx \\
&= \sqrt{\frac{2}{L}} \int_0^L \left[p \, x \, (L-x)\right] \, \sin\left(n\pi\frac{x}{L}\right) dx \\
&= \sqrt{\frac{2}{L}} \, 2p \, \left(\frac{L}{n\pi}\right)^3 \left[1-(-1)^n\right].
\end{aligned}$$

and we have

$$1-(-1)^n = \begin{cases} 0 & \text{if} \quad n \text{ is even,} \\ 2 & \text{if} \quad n \text{ is odd.} \end{cases}$$



(a) Temperature Distribution

(b) Temperature Evolution

Figure A.3: Simulation of the heat equation with Dirichlet boundary conditions. The scaling of the initial temperature is $p = 4$. The computed temperature decreases in time towards zero.

Thus, we note the coefficients as

$$\tilde{c}_n = \begin{cases} 0 & \text{if} \quad n \text{ is even,} \\ 4p\sqrt{\frac{2}{L}}\left(\frac{L}{n\pi}\right)^3 & \text{if} \quad n \text{ is odd.} \end{cases}$$

We insert the coefficients $\tilde{c}_n$ in the solution (A.18), we define the new index $k = 2n-1$ and we obtain the solution

$$\vartheta(t,x) = 8p\frac{L^2}{\pi^3}\sum_{n=1}^{\infty}\frac{1}{(2k-1)^3}\sin\left([2k-1]\,\pi\frac{x}{L}\right)\exp\left(-\alpha\left[\frac{[2k-1]\,\pi}{L}\right]^2 t\right). \tag{A.19}$$

We evaluate the solution of the Dirichlet problem (A.19) for the same example as above with length $L = 1$, diffusivity $\alpha = 0.1$, scaling $p = 4$, and $k \in \{1,\ldots,100\}$. We portray the resulting temperatures in Fig. A.3, where we see that the temperatures are decreasing towards zero because the data on both boundary side is fixed at zero.

A.2 Riccati Equation

In Section 8.1, we apply the linear-quadratic regulator approach on the heat conduction problem to find a stabilizing feedback law. Here, we derive the feedback law (8.6) and the algebraic Riccati equation (8.7). The subsequent ideas are based on [172, p. 296]. Further information about solving the linear-quadratic problem may be found in [153, page 120], [173, p. 363] and [174, p. 218].

We consider the quadratic optimal control problem

$$\min\left\{J(u) = z(T_f)^\top S z(T_f) + \int_0^{T_f} z(t)^\top Q z(t) + u(t)^\top R u(t)\,dt\right\}$$

with subject to the state space system

$$\frac{d}{dt}z(t) = A\,z(t) + B\,u(t) \quad \text{with} \quad z(0) = z_0. \tag{A.20}$$

We have the states $z : [0, T_f] \to \mathbb{R}^N$, the input signals $u : [0, T_f) \to \mathbb{R}^{N_u}$ and the matrices $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times N_u}$, $S, Q \in \mathbb{R}^{N\times N}$ and $R \in \mathbb{R}^{N_u\times N_u}$. We note the Hamiltonian

$$H(z,u,v) = \frac{1}{2}\left[z^\top Q z + u^\top R u\right] + v^\top\left[A\,z + B\,u\right]$$

with costate $v : [0, T_f] \to \mathbb{R}^N$ and we derive necessary (first-order) optimal-

ity conditions

$$\frac{\partial}{\partial v} H(z,u,v) = A\,z + B\,u = \frac{d}{dt} z(t), \tag{A.21a}$$

$$\frac{\partial}{\partial u} H(z,u,v) = Ru + B^\top v = 0, \tag{A.21b}$$

$$\frac{\partial}{\partial z} H(z,u,v) = Qz + A^\top v = -\frac{d}{dt} v(t) \tag{A.21c}$$

with the terminal value

$$v(T_f) = \frac{d}{dz(T_f)} \left[z(T_f)^\top S z(T_f) \right] = S z(T_f).$$

We obtain from Eq. (A.21b) the optimal input signal

$$u^*(t) = -R^{-1} B^\top v(t) \tag{A.22}$$

and we insert u^* in the state-space system (A.20) to yield the closed-loop system

$$\frac{d}{dt} z(t) = A\,z + B\,u = A\,z(t) - BR^{-1}B^\top v(t). \tag{A.23}$$

We summarize the differential equations (A.21c) and (A.23) as

$$\begin{pmatrix} \dot{z}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{pmatrix} A & -BR^{-1}B^\top \\ Q & A^\top \end{pmatrix} \begin{pmatrix} z(t) \\ v(t) \end{pmatrix} \tag{A.24}$$

with the initial value $z(0) = z_0$ and the terminal value $v(T_f) = Sz(T_f)$. As we have a system of linear ODEs in Eq. (A.24), we can consider a linear state-to-costate mapping

$$v(t) = P(t) z(t) \tag{A.25}$$

with $P : [0, T_f \rightarrow \mathbb{R}^{N \times N}]$. In the end of this section, we note one way to proceed from Eq. (A.24) to Eq. (A.25).

The mapping (A.25) is inserted in Eq. (A.22) and we find the feedback law

$$u^*(t) = -R^{-1} B^\top\, P(t) z(t).$$

In the next steps we derive the Riccati equation to find P. We differentiate the mapping (A.25) as

$$\begin{aligned} \frac{d}{dt} v(t) &= \dot{P}(t) z(t) + P(t) \dot{z}(t) \\ &= \dot{P}(t) z(t) + P(t) \left[A\,z(t) - BR^{-1}B^\top P(t) z(t) \right] \\ &= \left[\dot{P}(t) + P(t)A - P(t)BR^{-1}B^\top P(t) \right] z(t). \end{aligned} \tag{A.26}$$

In the third optimality condition (A.21c), we specify the the derivative $\frac{d}{dt} v(t)$ as

$$\begin{aligned} \frac{d}{dt} v(t) &= -Qz(t) - A^\top v(t) = -Qz(t) - A^\top P(t) z(t) \\ &= \left[-Q - A^\top P(t) \right] z(t) \end{aligned} \tag{A.27}$$

We summarize Eq. (A.26, A.27) and we obtain the *Riccati differential equation*

$$\dot{P}(t) + Q + A^\top\, P(t) + P(t)\, A - P(t)\, B\, R^{-1}\, B^\top\, P(t) = 0.$$

If we consider an infinite time horizon, $T_f \to \infty$, the terminal costs vanish as

$$z(T_f)^\top \, S \, z(T_f) = 0$$

and P is constant, $\dot{P} \equiv 0$. Hence, we yield the *algebraic Riccati equation*

$$0 = Q + P\,A + A^\top P - P\,B\,R^{-1}\,B^\top P.$$

Approach to find Equation (A.25)

Next, we propose a naive approach to calculate the state-to-costate mapping (A.25). Firstly, we introduce

$$w(t) := \begin{pmatrix} z(t) \\ v(t) \end{pmatrix} \quad \text{and} \quad M := \begin{pmatrix} A & -BR^{-1}B^\top \\ Q & A^\top \end{pmatrix}$$

such that $\dot{w}(t) = Mw(t)$ expresses the differential equation (A.24). We solve this ODE from any time $t \in [0, T_f)$ towards the final time T_f as

$$w(T_f) = \underbrace{\exp\left(\int_t^{T_f} M\,d\tau\right)}_{=:\Omega(t)} w(t) \tag{A.28}$$

in which we have the 2×2-matrix

$$\Omega = \begin{pmatrix} \Omega_{1,1} & \Omega_{1,2} \\ \Omega_{2,1} & \Omega_{2,2} \end{pmatrix}.$$

We formulate the solution (A.28) in terms of the original states $z(t)$ and $v(t)$ with identity $v(T_f) = Sz(T_f)$ as

$$\begin{pmatrix} z(T_f) \\ Sz(T_f) \end{pmatrix} = \begin{pmatrix} \Omega_{1,1}(t) & \Omega_{1,2}(t) \\ \Omega_{2,1}(t) & \Omega_{2,2}(t) \end{pmatrix} \begin{pmatrix} z(t) \\ v(t) \end{pmatrix} \tag{A.29}$$

Now, we solve the linear equations (A.29) and we note the state-to-costate mapping as

$$\begin{aligned} v(t) &= \left[\Omega_{2,2}(t) - S\Omega_{1,2}(t)\right]^{-1} \left[\Omega_{2,1}(t) - S\Omega_{1,1}(t)\right] \, x(t) \\ &= P(t)\, x(t). \end{aligned}$$

B

Implementation of Simulations

In this thesis, we present several simulation results to exemplify and visualize the proposed concepts and methods. These numerical experiments are implemented with JULIA programming language on a basis of the software library *Hestia.jl* [44]. The provided functions of *Hestia.jl* are explained in the online documentation [175]. The simulations are stored online on GitHub and Zenodo in the project *ThesisSimulations.jl* [176, 177].

We need to specify several coefficients to setup a heat conduction simulation with *Hestia.jl*. First of all, we define the dimensions: length L, width W, height H, and their corresponding number of finite volume cells: N_j, N_m, N_k. In the next step, we set the material properties with the (anisotropic) thermal conductivity λ, density ρ and specific heat capacity c. On all boundary sides, we have a thermal emission and so we denote a heat transfer coefficient h and an emissivity ε. In case of an insulated boundary side, the values $h = \varepsilon = 0$ are set by default. If we assume actuators and sensors, then we need to specify the number of actuators N_u and sensors N_y, the corresponding boundary sides, the checkerboard patterns in case of a three-dim. problem and the spatial characteristics with the scaling m, curvature matrix M and power ν. The central point x_c is computed internally. Finally, we state the initial temperature ϑ_0 and the simulation time T_{final} or T_{ff} for the feed-forward control.

Subsequently, we list the source code files of the simulations and the corresponding figures in this thesis.

Heat Conduction

Fig. 2.3	via	`11_slow_fast_heat_conduction.jl`
Fig. 2.4	via	`12_anisotropic_heat_conduction.jl`
Fig. 2.6	via	`13_dynamic_heat_conduction.jl`
Fig. 2.7	via	`14_heat_supply_vs_emission.jl`
Fig. 2.9	via	`15_heat_transfer_heat_radiation.jl`

Folder: `src/modeling`

Approximated Linear System

Fig. 4.8	via	`21_relative_error_condition_number.jl`
Fig. 4.11	via	`22_analytical_sol_gauss_quad.jl`

Folder: `src/simulation`

Numerical Time Integration

Fig. 5.8 via `23_numerical_error_time_integration.jl`

Folder: `src/simulation`

The Control System Framework

Fig. 6.6 via `31_actuation_narrow_wide.jl`
Fig. 6.5 via `32_actuator_characteristics_2d.jl`

Folder: `src/control_feed_forward`

Feed-forward Control

Fig. 7.1 via `41_gevrey_transition_bump.jl`
Fig. 7.2 via `42_gevrey_derivatives.jl`
Fig. 7.4 via `43_gevrey_input_heat_eq_pde.jl`
Fig. 7.7 via `44_flatness_ode_1d.jl`
Fig. 7.9 via `45_flatness_ode_2d.jl.jl`
Fig. 7.10 via `46_polynomial_transition.jl`
Fig. 7.15 via `47_opt_input_approximation.jl`
Fig. 7.16 via `47_opt_input_approximation.jl.jl`
Fig. 7.18 via `48_opt_reference_tracking.jl`
Fig. 7.19 via `48_opt_reference_tracking.jl`
Fig. 7.21 via `49_opt_energy_parameter_search.jl`
Fig. 7.22 via `49_opt_energy_parameter_search.jl`

Folder: `src/control_feed_forward`

Simulation of the Feed-forward Controlled System

Fig. 7.27 via `51_ff_example_approx_input.jl`
Fig. 7.28 via `51_ff_example_approx_input.jl`
Fig. 7.29 via `52_ff_example_energy_supply.jl`
Fig. 7.30 via `52_ff_example_energy_supply.jl`
Fig. 7.31 via `52_ff_example_energy_supply.jl`
Fig. 7.32 via `53_ff_example_optimization.jl`
Fig. 7.33 via `53_ff_example_optimization.jl`
Fig. 7.34 via `53_ff_example_optimization.jl`

Folder: `src/control_feed_forward/example_2d`

Closed-Loop Control

Fig. 8.2 via `61_lqr_linear_2d.jl`
Fig. 8.4 via `62_step_impulse_response.jl`
Fig. 8.5 via `63_mpc_step_response_2d.jl`
Fig. 8.6 via `64_mpc_linear_2d.jl`

Folder: `src/control_feedback`

Simulation and Control of Heat Conduction in a Cuboid

Fig. 8.9 via `71_cuboid_energy_opt.jl`
Fig. 8.10 via `72_cuboid_opt_control.jl`
Fig. 8.11 via `73_cuboid_mpc.jl`
Fig. 8.12 via `74_cuboid_volume_plot.jl`

Folder: `src/control_feedback/example_3d`

Analytical Solution of the Heat Equation

Fig. A.2 via `81_analytical_solution_neumann_dirichlet.jl`
Fig. A.3 via `81_analytical_solution_neumann_dirichlet.jl`

Folder: `src/simulation`

List of Figures

Bibliography

Introduction

[1] BIPM, Le Système international d'unités / The International System of Units ('The SI Brochure'), Ninth. Bureau international des poids et mesures, 2019. [Online]. Available: http://www.bipm.org/en/si/si_brochure/

[2] J. B. Clarke, J. W. Hastie, L. H. E. Kihlborg, R. Metselaar, and M. M. Thackeray, "Definitions of terms relating to phase transitions of the solid state (IUPAC Recommendations 1994)," Pure and Applied Chemistry, vol. 66, no. 3, pp. 577–594, 1994, doi:10.1351/pac199466030577.

[3] L. C. Evans, Partial differential equations, vol. 19. American Mathematical Society, 2010.

[4] W. Arendt and K. Urban, Partielle Differenzialgleichungen. Springer, 2010.

[5] M. Frewin and D. Scott, "Finite element model of pulsed laser welding," WELDING JOURNAL-NEW YORK-, vol. 78, pp. 15-s, 1999.

[6] V. Petzet, C. Büskens, H. J. Pesch, A. Prikhodovsky, V. Karkhin, and V. Ploshikhin, "Elimination of hot cracking in laser beam welding," in PAMM: Proceedings in Applied Mathematics and Mechanics, 2004, vol. 4, no. 1, pp. 580–581.

[7] J. P. Bergmann et al., "Prevention of solidification cracking during pulsed laser beam welding," vol. 17-09. Ilmenau, Aug. 16, 2017. [Online]. Available: https://nbn-resolving.org/urn:nbn:de:gbv:ilm1-2017200422.

[8] R. Herzog, and D. Strelnikov, "An optimal control problem for single-spot pulsed laser welding," J. of Math. in Industry, vol. 13, no. 1, p. 4, 2023.

[9] M. Bielenin, Prozessstrategien zur Vermeidung von Heißrissen beim Schweißen von Aluminium mit pulsmodulierbaren Laserstrahlquellen. Universitätsverlag Ilmenau, Ilmenau, 2021. doi: 10.22032/dbt.47297.

[10] S. Ecklebe, T. Buchwald, P. Rüdiger, and J. Winkler, "Model predictive control of the vertical gradient freeze crystal growth process," IFAC-PapersOnLine, vol. 54, no. 6, pp. 218–225, 2021.

[11] S. Ecklebe, F. Woittennek, C. Frank-Rotsch, N. Dropka, and J. Winkler, “Toward model-based control of the vertical gradient freeze crystal growth process,” IEEE Transactions on Control Systems Technology, vol. 30, no. 1, pp. 384–391, 2021.

[12] J. Rudolph, J. Winkler, and F. Woittennek, Flatness Based Control of Distributed Parameter Systems: Examples and Computer Exercises from Various Technological Domains. Shaker, 2003.

[13] S. Ecklebe, Beiträge zur Regelung des Vertical-Gradient-Freeze-Kristallzüchtungsprozesses auf Basis verteiltparametrischer Modelle. Doctoral thesis. Technische Universität Dresden, Dresden, 2024.

[14] R. Waser, Nanotechnology: Volume 3: Information Technology I. Wiley, 2008.

[15] H. J. Levinson, Principles of lithography. SPIE press, 2005.

[16] A. Tay, K.K. Tan, S. Zhao, and T.H. Lee, “Predictive Ratio Control of Multizone Thermal Processing System in Lithography,” IFAC Proceedings Volumes, vol. 41, no. 2, pp. 10863–10868, 2008.

[17] Y.-M. Lee, Efficient Extreme Ultraviolet Mirror Design: An FDTD Approach. IOP Publishing, 2021.

[18] P. Laube, “Lithografie: Belichten und Belacken,” [Online]. Available: https://www.halbleiter.org/lithografie/belichten/. [Accessed 13-Dec-2024].

[19] S. Rizvi, Handbook of photomask manufacturing technology. CRC Press, 2018.

[20] Semiconductor Engineering, “Photomask,” [Online]. Available: https://semiengineering.com/knowledge_centers/manufacturing/lithography/photomask/. [Accessed 13-Dec-2024].

[21] J. Lee et al., “Thermal design of hot plate for 300-mm wafer heating in post-exposure bake,” Microelectronic engineering, vol. 88, no. 11, pp. 3195–3198, 2011.

[22] L. Berger et al., “Global critical dimension uniformity improvement for mask fabrication with negative-tone chemically amplified resists by zone-controlled postexposure bake,” Journal of Micro/Nanolithography, MEMS, and MOEMS, vol. 3, no. 2, pp. 203–211, 2004, doi: 10.1117/1.1683338.

[23] L. Berger, P. Dress, S.-H. Yang, and C.-H. Kuo,“Qualification of design-optimized multi-zone hotplate for 45nm node mask making,” in Photomask and Next-Generation Lithography Mask Technology XIV, 2007, vol. 6607, pp. 117–127.

[24] W. Saule, L. Berger, C. Krauss, and R. Weihing, ’Method and device for the thermal treatment of substrates’, US-7842905-B2, 2010.

[25] SPS-International, “POLOS®HOTPLATE 200,” [Online]. 2024. Available: https://www.sps-polos.com/content/net_products/675-POLOS_Hotplate_200_-_Datasheet_-_2024.pdf.

[26] Y. Geng, Real-time monitoring and control of critical dimensions in Lithography. Doctoral thesis. National University of Singapore, Singapore, 2012.

[27] K. V. Ling, W. K. Ho, B. Wu, A. G. Aribowo, Y. Feng and H. Yan, “Experimental evaluation of Multiplexed MPC for semiconductor manufacturing,” 2009 7th Asian Control Conference, Hong Kong, China, 2009, pp. 1719–1722.

[28] Y. Han, Temperature sensing and control in multi-zone semiconductor thermal processing. Doctoral thesis. National University of Singapore, Singapore, 2009.

[29] E. Joelianto and I. G. Prasetia, “Bake plate control using a robust multiplexed model predictive control (RMMPC),” 2011 2nd International Conference on Instrumentation Control and Automation, Bandung, Indonesia, 2011, pp. 175–180, doi: 10.1109/ICA.2011.6130151.

[30] X. Wang, W. K. Ho, and K. V. Ling, “Computational load comparison of multiplexed and standard model predictive control,” in Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering, 2018, pp. 17–22.

[31] S. Franssila, Introduction to microfabrication. John Wiley & Sons, 2010.

[32] Stefan Peters, Rapid Thermal Processing of Crystalline Silicon Materials and Solar Cells. Doctoral thesis. Universität Konstanz, Konstanz, 2004.

[33] E. Dassau, B. Grosman, and D. R. Lewin, “Modeling and temperature control of rapid thermal processing,” Computers & chemical engineering, vol. 30, no. 4, pp. 686–697, 2006.

Author’s Contributions

[34] S. Scholz and L. Berger, “Modeling of a multiple source heating plate,” arXiv preprint. arXiv:2011.14939, 2020.

[35] S. Scholz and L. Berger, “Hestia. jl: A Julia Library for Heat Conduction Modeling with Boundary Actuation.,” Simul. Notes Eur., vol. 33, no. 1, pp. 27–30, 2023.

[36] S. Scholz and L. Berger, “Fast computation of function composition derivatives for flatness-based control of diffusion problems,” Journal of Mathematics in Industry, vol. 14, no. 1, p. 15, 2024.

[37] S. Scholz, C. Bonenberger, N. Scheiter and L. Berger, “Simulation and Control of 2-Dimensional Anisotropic Heat Conduction,” in ARGESIM Report 46 / Tagungsband Kurzbeiträge ASIM SST 2024, 27. Symposium Simulationstechnik, München.

[38] S. Scholz, L. Berger and D. Lebiedz , “Benchmarking of Flatness-based Control of the Heat Equation,” in ARGESIM Report 47 / Tagungsband Langbeiträge ASIM SST 2024, 27. Symposium Simulationstechnik, München.

[39] S. Scholz and L. Berger, “Optimization-based Trajectory Planning for Heat Conduction,” in Proceedings of the 2024 25th International Carpathian Control Conference (ICCC), Krynica Zdrój, Poland, 2024, pp. 1-5, doi: 10.1109/ICCC62069.2024.10569610.

[40] S. Scholz and L. Berger, “Optimization-based Reference Tracking for Two-Dimensional Multiple Source Heating,” in Proceedings of the 2024 IEEE Conference on Decision and Control (CDC), Milano, Italy, 2024. (accepted, not yet published)

[41] D. Peters, S. Scholz, and L. Berger, “Dynamic Mode Decomposition for Cascaded Electrical Circuits,” in Proceedings of the 2023 6th International Conference on Mathematics and Statistics, 2023, pp. 147–151.

[42] C. Bonenberger, S. Scholz and N. Scheiter, “Data-adaptive dynamic simulation via structured Dynamic Mode Decomposition,” in ARGESIM Report 46 / Tagungsband Kurzbeiträge ASIM SST 2024, 27. Symposium Simulationstechnik, München.

[43] C. Bonenberger, S. Scholz and M. Schneider, “From Data-Driven to Model-Driven Learning via Structured Dynamic Mode Decomposition,” in Proceedings of the 2024 IEEE Conference on Decision and Control (CDC), Milano, Italy, 2024. (accepted, not yet published)

Software

[44] S. Scholz, stephans3/Hestia.jl: v0.3.0. Zenodo, 2023. doi:10.5281/zenodo.10044594.

[45] S. Scholz, stephans3/BellBruno.jl: Add to Zenodo. Zenodo, 2023. doi: 10.5281/zenodo.7685927.

Heat Conduction

[46] B. Laroche, P. Martin, and P. Rouchon, “Motion planning for the heat equation,” in Int. J. of Robust and Nonlinear Control: IFAC-Affiliated J., vol. 10, no. 8, pp. 629–643, 2000.

[47] R. Katz, I. Basre, and E. Fridman, “Delayed finite-dimensional observer-based control of 1D heat equation under Neumann actuation,” in 2021 European Control Conference (ECC), 2021, pp. 2500–2505.

[48] A. Lavasani, D. Bulmash, and S. D. Sarma, “Wiedemann-Franz law and Fermi liquids,” Physical Review B, vol. 99, no. 8, p. 085104, 2019.

[49] H. D. Baehr, and K. Stephan, Wärme-und Stoffübertragung, vol. 7. Springer, 1994.

[50] W. Weißbach, Werkstoffkunde: Strukturen, Eigenschaften, Prüfung. Springer-Verlag, 2012.

[51] O. Volkova, Mathematische Modellierung und experimentelle Untersuchung der Schnellerstarrung von Stählen. Technischen Universität Bergakademie Freiberg, 2002.

[52] F. Richter, “Die physikalischen Eigenschaften der Stähle „Das 100 - Stähle - Programm“” [Online]. Available: https://www.tugraz.at/fileadmin/user_upload/Institute/IEP/Thermophysics_Group/Files/Staehle-Richter.pdf. 2011. [Accessed 10-Feb-2025].

[53] J. H. Lienhard IV, and J. H. Lienhard V, A heat transfer textbook. Phlogistron, 2020.

[54] M. W. Zemansky, and R. H. Dittman, Heat and Thermodynamics. McGraw-Hill, 1997.

[55] MacTutor, “Mikhail Vasilevich Ostrogradski,” [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Ostrogradski/. [Accessed 09-Jan-2024].

[56] F. Hoppe, Optimal Control of Quasilinear Parabolic PDEs: Theory and Numerics. Doctoral thesis. Universitäts-und Landesbibliothek Bonn, Bonn, 2022.

[57] V. Shankar, G. B. Wright, A. L. Fogelson, and R. M. Kirby, “A radial basis function (RBF) finite difference method for the simulation of reaction–diffusion equations on stationary platelets within the augmented forcing method,” International Journal for Numerical Methods in Fluids, vol. 75, no. 1, pp. 1–22, 2014.

[58] L. Su, “A radial basis function (RBF)-finite difference (FD) method for the backward heat conduction problem,” Applied Mathematics and Computation, vol. 354, pp. 232–247, 2019.

[59] D. Dutykh, “A brief introduction to pseudo-spectral methods: application to diffusion problems,” arXiv preprint arXiv:1606.05432, 2016.

[60] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” Journal of Computational physics, vol. 378, pp. 686–707, 2019.

[61] J. Berg and K. Nyström, "Data-driven discovery of PDEs in complex datasets," Journal of Computational Physics, vol. 384, pp. 239–252, 2019.

[62] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," arXiv preprint arXiv:1711.10561, 2017.

[63] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations," arXiv preprint arXiv:1711.10566, 2017.

[64] R. Eymard, R. Herbin, and T. Gallouët, "Finite volume method," Scholarpedia, vol. 5, no. 6, p. 9835, 2010, doi: 10.4249/scholarpedia.9835.

[65] T. Barth, R. Herbin, and M. Ohlberger, "Finite Volume Methods: Foundation and Analysis," Encyclopedia of Computational Mechanics Second Edition, pp. 1–60, 2018.

[66] R. J. LeVeque, Finite volume methods for hyperbolic problems, vol. 31. Cambridge university press, 2002.

[67] M. Schlottke-Lakemper, "A direct-hybrid method for aeroacoustic analysis," Doctoral thesis, RWTH Aachen University, Aachen, 2017.

[68] S. F. Nemadjieu, "Finite volume methods for advection diffusion on moving interfaces and application on surfactant driven thin film flow," Doctoral thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, 2012.

[69] I. D. Mishev, "Finite volume methods on Voronoi meshes," Numerical Methods for Partial Differential Equations: An International Journal, vol. 14, no. 2, pp. 193–212, 1998.

Approximated Linear System

[70] MacTutor, "Semyon Aranovich Gershgorin," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Gershgorin/. [Accessed 27-Apr-2024].

[71] R. J. LeVeque, Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. SIAM, 2007.

[72] MacTutor, "Otto Toeplitz," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Toeplitz/. [Accessed 02-May-2024].

[73] L. Reichel and L. N. Trefethen, "Eigenvalues and pseudo-eigenvalues of Toeplitz matrices," Linear Algebra and its Applications, vol. 162–164, pp. 153–185, 1992, doi: https://doi.org/10.1016/0024-3795(92)90374-J.

[74] D. Kulkarni, D. Schmidt, and S.-K. Tsui, "Eigenvalues of tridiagonal pseudo-Toeplitz matrices," Linear Algebra and its Applications, vol. 297, pp. 63–80, 1999.

[75] S. Noschese, L. Pasquini, and L. Reichel, "Tridiagonal Toeplitz matrices: properties and novel applications," Numerical linear algebra with applications, vol. 20, no. 2, pp. 302–326, 2013.

[76] R. M. Gray and others, "Toeplitz and circulant matrices: A review," Foundations and Trends®in Communications and Information Theory, vol. 2, no. 3, pp. 155–239, 2006.

[77] StackExchange, "How to find the eigenvalues of tridiagonal Toeplitz matrix?," [Online]. Available: https://math.stackexchange.com/questions/955168/how-to-find-the-eigenvalues-of-tridiagonal-toeplitz-matrix. [Accessed 02-May-2024].

[78] W.-C. Yueh, "Eigenvalues of several tridiagonal matrices.," Applied Mathematics E-Notes [electronic only], vol. 5, pp. 66–74, 2005.

[79] S. Kouachi, "Eigenvalues and eigenvectors of tridiagonal matrices," The Electronic Journal of Linear Algebra, vol. 15, pp. 115–133, 2006.

[80] D. A. H. Ahmed, "On the characteristic polynomial, eigenvalues for block tridiagonal matrices," Journal of Discrete Mathematical Sciences and Cryptography, vol. 25, no. 6, pp. 1745–1756, 2022, doi: 10.1080/09720529.2020.1854939.

[81] E. Isaacson and H. B. Keller, Analysis of numerical methods. Courier Corporation, 2012.

[82] P. Moin, Fundamentals of engineering numerical analysis. Cambridge University Press, 2010.

[83] M. Abramowitz and I. A. Stegun, Handbook of mathematical functions with formulas, graphs, and mathematical tables, vol. 55. US Government printing office, 1968.

[84] MacTutor, "Benjamin Olinde Rodrigues," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Rodrigues/. [Accessed 04-Jul-2024].

[85] I. Bogaert, "Iteration-free computation of Gauss–Legendre quadrature nodes and weights," SIAM Journal on Scientific Computing, vol. 36, no. 3, pp. A1008–A1026, 2014. s

Numerical Time Integration

[86] MacTutor, "Leonhard Euler," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Euler/. [Accessed 01-Jan-2025].

[87] MacTutor, "Carl David Tolmé Runge," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Runge/. [Accessed 01-Jan-2025].

[88] MacTutor, "Martin Wilhelm Kutta," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Kutta/. [Accessed 01-Jan-2025].

[89] MacTutor, "John Crank," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Crank/. [Accessed 18-Jul-2024].

[90] MacTutor, "Phyllis Nicolson," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Nicolson/. [Accessed 18-Jul-2024].

[91] G. G. Dahlquist, "A special stability problem for linear multistep methods," BIT Numerical Mathematics, vol. 3, no. 1, pp. 27–43, 1963.

[92] E. Hairer and G. Wanner, Solving ordinary differential equations II, vol. 375. Springer Berlin Heidelberg New York, 1996.

[93] B. L. Ehle, On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems. Doctoral thesis. University of Waterloo, Waterloo, Ontario, 1969.

[94] University of Auckland, "Biography of John C. Butcher," [Online]. Available: https://www.math.auckland.ac.nz/~butcher/biography.html. [Accessed 24-Jul-2024].

[95] E. Hairer, S. Nørsett, and G. Wanner, Solving Ordinary Differential Equations I Nonstiff Problems. Springer-Verlag Berlin Heidelberg, 2011.

[96] W. Kutta, "Beitrag zur näherungsweisen Integration totaler Differentialgleichungen," Teubner, 1901.

[97] C. Tsitouras, I. T. Famelis, and T. Simos, "On modified Runge–Kutta trees and methods," Computers & Mathematics with Applications, vol. 62, no. 4, pp. 2101–2111, 2011.

[98] C. A. Kennedy and M. H. Carpenter, "Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review," NASA, 2016.

[99] J. B. Jørgensen, M. R. Kristensen, and P. G. Thomsen, "A family of ESDIRK integration methods," arXiv preprint arXiv:1803.01613, 2018.

[100] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," SIAM review, vol. 59, no. 1, pp. 65–98, 2017.

[101] C. Rackauckas et al., SciML/DifferentialEquations.jl: v7.6.0. Zenodo, 2022. doi: 10.5281/zenodo.7239171.

[102] C. A. Kennedy and M. H. Carpenter, “Additive Runge–Kutta schemes for convection–diffusion–reaction equations,” Applied numerical mathematics, vol. 44, no. 1–2, pp. 139–181, 2003.

[103] A. Murua, “From Runge-Kutta Methods to Hopf Algebras of Rooted Trees,” Algebra and Applications 2: Combinatorial Algebra and Hopf Algebras, p. 179, 2021.

Control System Framework

[104] I. Wyzkiewicz et al., “Self-regulating heater for microfluidic reactors,” Sensors and Actuators B: Chemical, vol. 114, no. 2, pp. 893–896, 2006, doi: https://doi.org/10.1016/j.snb.2005.08.028.

[105] M. Wesolowski and A. Czaplicki, “Modeling of Flat Inductor - Workpiece Heating Systems for Design of Induction Heaters,” 2018 Progress in Applied Electrical Engineering (PAEE), Koscielisko, Poland, 2018, pp. 1-5, doi: 10.1109/PAEE.2018.8441121.

[106] A. Tommasi et al., “Modeling, fabrication and testing of a customizable micromachined hotplate for sensor applications,” Sensors, vol. 17, no. 1, p. 62, 2016. m

[107] BACH Resistor Ceramics GmbH, “Waferbeheizung,” [Online]. Available: `https://www.bachrc.de/products/ceramic-heating-elements/wafer-heaters/`. [Accessed 06-Jan-2025].

[108] M. Fliess, J. Lèvine, P. Martin, and P. Rouchon, “Flatness and defect of non-linear systems: introductory theory and examples,” International journal of control, vol. 61, no. 6, pp. 1327–1361, 1995.

[109] T. Utz, T. Meurer, and A. Kugi, “Motion planning for the heat equation with radiation boundary conditions based on finite difference semi-discretizations,” IFAC Proceedings Volumes, vol. 40, no. 12, pp. 306–311, 2007.

[110] T. Utz, K. Graichen, and A. Kugi, “Trajectory planning and receding horizon tracking control of a quasilinear diffusion-convection-reaction system,” IFAC Proceedings Volumes, vol. 43, no. 14, pp. 587–592, 2010.

[111] T. Utz, Control of parabolic partial differential equations based on semi-discretizations. Doctoral thesis. Technische Universität Wien, Wien, 2012.

[112] J. Levine, Analysis and control of nonlinear systems: A flatness-based approach. Springer Science & Business Media, 2009.

[113] P. Martin, R. M. Murray, and P. Rouchon, “Flat systems, equivalence and trajectory generation,” Ecole des Mines de Paris, Technical report, 2003.

[114] J. Deutscher, Zustandsregelung verteilt-parametrischer Systeme. Springer-Verlag, 2012.

[115] T. Meurer, Control of higher–dimensional PDEs: Flatness and backstepping designs. Springer Science & Business Media, 2012.

[116] F. Ollivier and A. Sedoglavic, "A generalization of flatness to nonlinear systems of partial differential equations. Application to the command of a flexible rod," IFAC Proceedings Volumes, vol. 34, no. 6, pp. 219–223, 2001.

[117] F. Tröltzsch, Optimal control of partial differential equations: theory, methods, and applications, vol. 112. American Mathematical Soc., 2010.

[118] T. Meurer and A. Kugi, "Trajectory planning for boundary controlled parabolic PDEs with varying parameters on higher-dimensional spatial domains," IEEE Transactions on Automatic Control, vol. 54, no. 8, pp. 1854–1868, 2009.

[119] J. B. Rawlings, D. Q. Mayne, M. Diehl, Model predictive control: theory, computation, and design, vol. 2. Nob Hill Publishing Madison, WI, 2017.

[120] L. Grüne and J. Pannek, Nonlinear model predictive control. Springer, 2017.

[121] N. Altmüller, Model Predictive Control for Partial Differential Equations. Doctoral thesis. Universität Bayreuth, 2014.

[122] S. Dietze and M. A. Grepl, "Reduced order model predictive control for parametrized parabolic partial differential equations," Applied Mathematics and Computation, vol. 453, p. 128044, 2023.

Open-Loop Control Design

[123] G. Teschl, Ordinary Differential Equations and Dynamical Systems, vol. 140. American Mathematical Society, 2012.

[124] F. G. Frobenius, "Ueber die Integration der linearen Differentialgleichungen durch Reihen.," Journal für die reine und angewandte Mathematik, vol. 76, p. 214, 1873.

[125] MacTutor, "Ferdinand Georg Frobenius," [Online]. Available: `https://mathshistory.st-andrews.ac.uk/Biographies/Frobenius/`. [Accessed 30-Sep-2024].

[126] MacTutor, "Lazarus Immanuel Fuchs," [Online]. Available: `https://mathshistory.st-andrews.ac.uk/Biographies/Fuchs/`. [Accessed 30-Sep-2024].

[127] MacTutor, "Karl Theodor Wilhelm Weierstrass," [Online]. Available: `https://mathshistory.st-andrews.ac.uk/Biographies/Weierstrass/`. [Accessed 30-Sep-2024].

[128] M. Gevrey, “Sur la nature analytique des solutions des équations aux dérivées partielles. Premier mémoire,” in Annales Scientifiques de l’École Normale Supérieure, 1918, vol. 35, pp. 129–190.

[129] Encyclopedia of Mathematics, “Gevrey class,” [Online]. Available: https://encyclopediaofmath.org/wiki/Gevrey_class. [Accessed 13-Dec-2024].

[130] F. Fischer, J. Gabriel, and S. Kerschbaum, coni - a Matlab toolbox facilitating the solution of control problems. Zenodo, 2023. doi: 10.5281/zenodo.10067379.

[131] P. Martin, L. Rosier and P. Rouchon,“Null controllability using flatness: a case study of a 1-D heat equation with discontinuous coefficients,” in 2015 European Control Conference (ECC), 2015, pp. 55–60.

[132] S. Dubljevic and P. D. Christofides, “Predictive output feedback control of parabolic partial differential equations (PDEs),” Industrial & engineering chemistry research, vol. 45, no. 25, pp. 8421–8429, 2006.

[133] N. Altmüller and L. Grüne, “Distributed and boundary model predictive control for the heat equation,” GAMM-Mitteilungen, vol. 35, no. 2, pp. 131–145, 2012.

[134] R. Voropai, A. Geletu, and P. Li, “Model Predictive Control of Parabolic PDE Systems under Chance Constraints,” Mathematics, vol. 11, no. 6, 2023, doi: 10.3390/math11061372.

[135] G. Rzadkowski, “Derivatives and Eulerian numbers,” The American Mathematical Monthly, vol. 115, no. 5, pp. 458–460, 2008.

[136] L. Comtet, Advanced Combinatorics: The art of finite and infinite expansions. Springer Science & Business Media, 2012.

[137] Piping-Designer, “Characteristic Length,” [Online]. Available: https://www.piping-designer.com/index.php/properties/classical-mechanics/252-characteristic-length/. [Accessed 15-Jan-2025].

[138] L. R. Glicksman and J. H. Lienhard V, Modeling and Approximation in Heat Transfer. Cambridge: Cambridge University Press, 2016.

[139] V. K. Dixit and C. Rackauckas, Optimization.jl: A Unified Optimization Package. Zenodo, 2023. doi: 10.5281/zenodo.7738525.

[140] P. Mogensen and A. Riseth, “Optim: A mathematical optimization package for Julia,” Journal of Open Source Software, vol. 3, no. 24, 2018.

[141] J. Revels, M. Lubin, and T. Papamarkou, “Forward-mode automatic differentiation in Julia,” arXiv preprint arXiv:1607.07892, 2016.

[142] W. W. Hager and H. Zhang, “Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent,” ACM Transactions on Mathematical Software (TOMS), vol. 32, no. 1, pp. 113–137, 2006.

[143] W. W. Hager and H. Zhang, "The limited memory conjugate gradient method," SIAM Journal on Optimization, vol. 23, no. 4, pp. 2150–2168, 2013.

[144] Optim.jl, "Conjugate Gradient Descent," [Online]. Available: https://julianlsolvers.github.io/Optim.jl/stable/algo/cg/. [Accessed 19-Jan-2025].

[145] J. Lu, "Gradient descent, stochastic optimization, and other tales," arXiv preprint arXiv:2205.00832, 2022.

[146] J. Nocedal and S. J. Wright, Numerical optimization. Springer, 1999.

[147] M. J. Kochenderfer and T. A. Wheeler, Algorithms for Optimization. The MIT Press, 2019.

[148] A. G. Baydin and B. A. Pearlmutter, "Automatic differentiation of algorithms for machine learning," arXiv preprint arXiv:1404.7456, 2014.

[149] A. H. Gebremedhin and A. Walther, "An introduction to algorithmic differentiation," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 10, no. 1, p. e1334, 2020.

Closed-Loop Control Design

[150] MacTutor, "Jacopo Francesco Riccati," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Riccati/. [Accessed 25-Dec-2024].

[151] MacTutor, "Rudolf Emil Kalman," [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/Kalman/. [Accessed 25-Dec-2024].

[152] K. J. Åström and R. Murray, Feedback systems: an introduction for scientists and engineers. Princeton university press, 2021.

[153] J. P. Hespanha, Linear systems theory. Princeton university press, 2018.

[154] K. A. Morris, Controller design for distributed parameter systems. Springer, 2020.

[155] MathWorks, "lqr Linear-Quadratic Regulator (LQR) design," [Online]. Available: https://www.mathworks.com/help/control/ref/lti.lqr.html. [Accessed 26-Dec-2024].

[156] MathWorks, "icare Implicit solver for continuous-time algebraic Riccati equations," [Online]. Available: https://www.mathworks.com/help/control/ref/icare.html. [Accessed 26-Dec-2024].

[157] A. Varga, D. Karrasch, B. Legat, and D. Aluthge, andreasvarga/MatrixEquations.jl: v2.4.2. Zenodo, 2024. doi: 10.5281/zenodo.10546715.

[158] A. Schaum et al., “Observer design for a nonlinear heat equation: Application to semiconductor wafer processing,” Journal of Process Control, vol. 119, pp. 34–43, 2022, doi: https://doi.org/10.1016/j.jprocont.2022.09.004.

Conclusion

[159] J. Rommes, W. H. Schilders, and H. A. van der Vorst, Model order reduction: theory, research aspects and applications. Springer, 2008.

[160] R. Milk, S. Rave, and F. Schindler, “pyMOR–generic algorithms and interfaces for model order reduction,” SIAM Journal on Scientific Computing, vol. 38, no. 5, pp. S194–S216, 2016.

[161] F. Rizzi, P. J. Blonigan, E. J. Parish, and K. T. Carlberg, “Pressio: Enabling projection-based model reduction for large-scale nonlinear dynamical systems,” arXiv preprint arXiv:2003.07798, 2020.

[162] A. Gloster, GPU methodologies for numerical partial differential equations. Doctoral thesis, Dublin, University College Dublin, 2020.

[163] T. Carlotto, R. V. da Silva, and J. M. V. Grzybowski, “GPGPU-accelerated environmental modelling based on the 2D advection-reaction-diffusion equation,” Environmental Modelling & Software, vol. 116, pp. 87–99, 2019.

[164] E. Adabag, M. Atal, W. Gerard, and B. Plancher,“MPCGPU: Real-Time Nonlinear Model Predictive Control through Preconditioned Conjugate Gradient on the GPU,” in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 9787–9794.

[165] J. Iwema, “Scientific Machine Learning,” [Online]. Available: `https://sciml.wur.nl/reviews/sciml/sciml.html`. [Accessed 03-Feb-2025].

[166] C. Rackauckas et. al., “SciML: Open Source Software for Scientific Machine Learning,” [Online]. Available: `https://sciml.ai/`. [Accessed 03-Feb-2025].

[167] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” Journal of Fluid Mechanics, vol. 656, pp. 5–28, 2010.

[168] J. L. Proctor, S. L. Brunton, and J. N. Kutz, “Dynamic mode decomposition with control,” SIAM Journal on Applied Dynamical Systems, vol. 15, no. 1, pp. 142–161, 2016.

[169] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. Brunton, “Physics-informed dynamic mode decomposition,” Proceedings of the Royal Society A, vol. 479, no. 2271, p. 20220576, 2023.

Appendix

[170] P. J. Olver, Introduction to partial differential equations, vol. 1. Springer, 2014.

[171] N. Subani, F. Jamaluddin, M. A. H. Mohamed, and A. D. H. Badrolhisam, "Analytical solution of homogeneous one-dimensional heat equation with Neumann boundary conditions," in Journal of Physics: Conference Series, 2020, vol. 1551, no. 1, p. 012002.

[172] M. Papageorgiou, M. Leibold, and M. Buss, Optimierung, vol. 4. Springer, 2015.

[173] E. D. Sontag, Mathematical control theory: deterministic finite dimensional systems, vol. 6. Springer Science & Business Media, 2013.

[174] H. Trentelman, A. Stoorvogel, M. Hautus, and L. Dewell, Control Theory for Linear Systems, Applied Mechanics Reviews, vol. 55, no. 5, Sep. 2002, doi: 10.1115/1.1497472.

[175] S. Scholz, "Hestia," [Online]. Available: `https://stephans3.github.io/Hestia.jl/dev/`. [Accessed 20-Jan-2025].

[176] S. Scholz, "ThesisSimulations.jl," [Online]. Available: `https://github.com/stephans3/ThesisSimulations.jl/`. [Accessed 11-Mar-2025].

[177] S. Scholz, stephans3/ThesisSimulations.jl: Initial Release. Zenodo, 2025. doi: 10.5281/zenodo.15005424.