Implementation of a method for the inverse calculation of the heat flux during the impact of a drop chain onto a heated surface

Unkelbach, Wilhelm

(2020)

DOI (TUprints): https://doi.org/10.25534/tuprints-00014185

Publication type: Bachelor Thesis

Division: 16 Department of Mechanical Engineering

Original source: https://tuprints.ulb.tu-darmstadt.de/14185

Implementation of a method for the inverse calculation of the heat flux during the impact of a drop chain onto a heated surface

Implementierung eines Verfahrens zur inversen Berechnung der Wärmestromdichte beim Aufprall einer Tropfenkette auf eine beheizte Oberfläche

Bachelor thesis by Wilhelm Unkelbach (Student ID: 2819914)

Date of submission: 14.10.2020

1. Review: Prof. Dr.-Ing. Jeanette Hussong
2. Review: Apl. Prof. Dr. Ilia Roisman
3. Review: J. Benedikt Schmidt, M.Sc.
Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

Mechanical Engineering Department
Institute for Fluid Mechanics and Aerodynamics

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Wilhelm Unkelbach, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 14.10.2020

Wilhelm Unkelbach

„Implementierung eines Verfahrens zur inversen Berechnung der Wärmestromdichte beim Aufprall einer Tropfenkette auf eine beheizte Oberfläche“

„Implementation of a method for the inverse calculation of the heat flux during the impact of a drop chain onto a heated surface“

Aufgabenstellung zur Bachelor Thesis von Wilhelm Unkelbach (2819914)
Betreuer: Prof. J. Hussong; Prof. I.V. Roisman; J.B. Schmidt, M.Sc.
Beginn der Arbeit: 27.04.2020

Problemstellung

In vielen technischen Anwendungen wie beispielsweise bei der Sprühkühlung, in Verbrennungsmotoren und Gasturbinen kommen Sprays in Verbindung mit heißen Oberflächen zum Einsatz. In Abhängigkeit von verschiedenen Prozessparametern wie der Oberflächentemperatur, Fluid- und Stoffeigenschaften treten verschiedene hydrodynamische und thermodynamische Phänomene auf, welche in Wechselwirkung miteinander die Performance der Anwendungsprozesse wesentlich beeinflussen. Einer dieser Effekte ist der Leidenfrost Effekt, welcher die Kühlleistung während der Sprühkühlung bei sehr hohen Temperaturen wesentlich reduziert.

Aufgrund der Vielzahl an Einflussfaktoren ist es notwendig, die Komplexität der nicht-isothermen Spray-Wand Interaktionen zu reduzieren, um die grundlegende Physik besser zu verstehen und modellieren zu können. Dabei konnte bereits ein fundamentales Verständnis rund um den Einzeltropfenaufprall, als zentrales Element des Sprayaufpralls, auf beheizte Oberflächen erlangt werden. Darüber hinaus wurde ein experimenteller Aufbau entwickelt, um den Aufprall von Tropfenketten und in Folge dessen die Abkühlung der Oberfläche zu untersuchen.

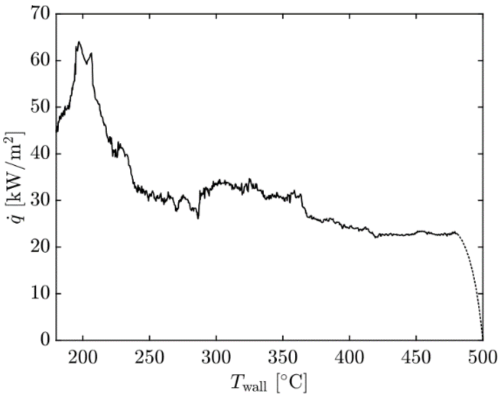

Abbildung 1: Wärmestromdichte beim Abkühlen einer heißen Oberfläche durch den Aufprall einer Tropfenkette

Aufgabenstellung

Ziel dieser Arbeit ist es, eine Berechnungsmethode zu entwickeln und validieren, um den Wärmestrom in Folge des Tropfenkettenaufpralls zu berechnen. Durch den Aufprall der Tropfenkette auf eine heiße Oberfläche kühlt diese ab, woraus ein dreidimensionaler Temperaturgradient im Material resultiert. Mithilfe des bereits bestehenden experimentellen Aufbaus ist es gelungen den Temperaturgradienten zweidimensional zu messen. Anhand dieser Messwerte soll nun eine Berechnung des inversen Wärmeleitungsproblems über eine Lösung der instationären fourierschen Wärmeleitungsgleichung erfolgen. Hierfür wurde am Fachgebiet bereits ein Lösungsansatz entwickelt, welcher noch in Matlab zu implementieren und testen ist. Darüber hinaus sind weitere bereits existierende Lösungsansätze in der Literatur zu recherchieren und für die vorliegenden Randbedingungen zu bewerten. Aus diesen ist ein geeignetes Lösungsverfahren auszuwählen und in Matlab zu implementieren. Anhand vorhandener Temperaturmessungen soll der Wärmestrom berechnet werden und der am SLA entwickelten Lösungsansatz mit dem gewählten Lösungsansatz aus der Literatur verglichen und validiert werden.
Die Aufgabe gliedert sich in folgende Teilaufgaben:

1. **Einarbeitung in die vorhandene Literatur**
2. **Literaturrecherche zu Lösungsansätzen des inversen, instationären Wärmeleitungsproblem.**
3. **Bewertung unterschiedlicher Lösungsansätze in Bezug auf den vorhandenen Messaufbau**
4. **Implementierung des am SLA entwickelten Lösungsansatzes, sowie des Lösungsansatzes aus der Literatur in Matlab und Anpassung an die vorhandenen Randbedingungen und Messungen**
5. **Berechnung des Wärmestroms und Validierung des Lösungsansatzes**
6. **Dokumentation der Arbeit**

Prof. Dr.-Ing. J. Hussong

Abstract

Spray cooling is one of the most efficient cooling methods, can achieve a very high, nearly uniform heat flux and thus reduce thermal loads. Spray-wall interaction is encountered in existing and emerging technologies and the underlying phenomenon associated with spray cooling. Spray cooling exploits the latent heat of vaporization and surpasses other conventional cooling methods. Drop impact is a punctual aspect of a monodisperse droplet chain, which reduces a spray to one dimension. The submitted thesis deals with the development of heat flux calculation models as means of solving the inverse heat conduction problem of drop impact on a heated surface.

In this work different existing models for solving the inverse heat conduction problem are introduced and one of them is applied. This existing method is compared to the newly developed calculation methods which are presented and compared. Firstly drop impact phenomena are explained for the isothermal and non-isothermal case. Then the abstraction is made to sprays. The heat conduction equation is introduced and subsequently solved for our experimental set-up.

The experimental set-up consists of a drop generator, a heated probe and measuring equipment. The drop generation and measuring equipment are used to obtain temperature measurements within the probe. The probe body is polished stainless steel with holes at different horizontal and radial positions for temperature measurement. These measurement points are used to calculate the surface temperature of the heated probe. The heat flux from the heated probe is the same as the heat flux absorbed by a droplet upon impact. Therefore the heat flux during drop impact is calculated from the temperature measurements within the heated probe body. The heat flux calculation results are used to understand the change in dissipated heat at different temperatures. The calculations are validated using a simulated test case with a known surface temperature and heat flux before the newly developed calculation methods are applied to real measurement data.

Contents

List of Figures VI

List of Tables VII

Nomenclature VIII

1 Introduction 1

2 Theoretical Background 2

2.1 Droplet Parameters . . . 2
2.2 Isothermal Drop Impact . . . 3
2.3 Non-isothermal Drop Impact . . . 4
2.3.1 Boiling Regimes . . . 4
2.3.2 Non-isothermal Drop Impact Outcomes . . . 6
2.3.3 Heat Flux in Droplet Chains . . . 7
2.4 Spray Cooling . . . 7
2.4.1 Spray Parameter . . . 7
2.4.2 Nukiyama Curve for Sprays . . . 7
2.4.3 Heat Flux Models in Spray Cooling Regimes . . . 8
2.5 General Heat Conduction Equation . . . 9
2.6 Inverse Problems . . . 9
2.7 Well-posed and Ill-posed Problems . . . 10

3 Experimental and Analytical Methods 11

3.1 Experimental Set-up . . . 11
3.2 Generic Test Case . . . 13
3.2.1 Linear Temperature Decrease . . . 13
3.2.2 Cosinusoidal Temperature Fluctuation with linear Decrease . . . 15
3.2.3 Sinusoidal Temperature Decrease with initial Discontinuity . . . 16

4 Solving the inverse Heat Conduction Equation 18

4.1 Existing inverse Heat Flux Calculation Methods . . . 19
4.1.1 Burggraf Solution . . . 19
4.1.2 Analytical Solution using Laplace Transform . . . 19
4.1.3 Levenberg Marquardt Method . . . 21
4.2 New inverse Heat Flux Calculation Methods . . . 22
4.2.1 Heat Conduction Equation . . . 22
4.2.2 Gaussian Error Function . . . 24
4.2.3 Differential Equation . . . 25

4.2.4 Finite Difference Method 27
4.2.5 1-dimensional Finite Difference Method 27
4.2.6 2-dimensional Finite Difference Method 28

5 Results **30**
5.1 Wall Temperature and Heat Flux in the Test Case 30
5.1.1 Linear Test Case 30
5.1.2 Final Test Case 33
5.2 Wall Temperature and Heat Flux in Measurement Data 35

6 Conclusion and Outlook **38**

Appendix **i**

List of Figures

2.1 Droplet parameters 2
2.2 Outcomes for isothermal drop impact, adapted from [1] 4
2.3 Nukiyama curve for boiling in droplet chains showcasing different boiling regimes, adapted from [2] 5
2.4 Outcomes for isothermal drop impact a) drop deposition, b) drop dancing, c) thermal atomization and d) drop rebound, adapted from [2] 6
2.5 Nukiyama curve for sprays, adapted from [3] 8

3.1 Experimental set-up, adapted from [4] 12
3.2 Probe geometry, adapted from [4] 12
3.3 Temperature decline and heat flux in linear test case 14
3.4 Temperature decline and heat flux in cosinusoidal test case 15
3.5 Temperature decline in final test case 16
3.6 Heat flux in final test case 16
3.7 Temperature decline and heat flux in final test case 17

4.1 Probe with impinging droplet, adapted from [5] 18
4.2 Change in temperature over depth 22
4.3 Heat Q through change in temperature 23
4.4 Heat flux using thermoelements 1 and 2 26
4.5 Temperature measurements and approximated temperature 28

5.1 Calculated temperature in the linear test case 31
5.2 Calculated temperature in the linear test case 31
5.3 Calculated heat flux in the linear test case 32
5.4 Calculated temperature in the final test case 33
5.5 Calculated heat flux in the final test case 34
5.6 Calculated temperature in measurement data 36
5.7 Calculated temperature in measurement data 36
5.8 Calculated heat flux in measurement data 37
5.9 Calculated heat flux in measurement data 37

List of Tables

3.1 FEM parameters 13

Nomenclature

Latin Letters

Symbol	Unit	Description
a	$\mathrm{m^2\,s^{-1}}$	thermal diffusivity
A		variable in differential equation method
A_i		constant in Gaussian error function method
b_k		constant in Monde solution
B		factor in spray cooling model
B		variable in differential equation method
B_i		constant in Gaussian error function method
c_p	$\mathrm{J\,kg^{-1}\,K^{-1}}$	specific heat at constant pressure
$C_{j,k}$		constant in Monde solution
C		variable in differential equation method
D	m	drop diameter
D		difference function in Levenberg Marquardt method
D		variable in differential equation method
D_0	m	initial Drop diameter
erf		Gaussian error function
$erfc$		complementary Gaussian error function
f_n		function in Burggraf solution
F_j		function in Monde solution
G		factor in spray cooling model
g_n		function in Burggraf solution
$G_{j,k}^{(12)}$		factor in Monde solution
j_m		mean mass flux density of a spray
J		jacobi matrix
K		factor in spray cooling model
L	m	length of probe
P		parameters for cubic spline interpolation in Levenberg Marquardt method
Pr	-	Prandtl number
Re	-	Reynolds number
Q	J	heat
$\dot{Q}$	W	heat flow
$\dot{q}$	$\mathrm{W\,m^{-2}\,s^{-1}}$	heat flux
t	s	time
T	K	temperature

Symbol	Unit	Description
U_0	$\mathrm{m\,s^{-1}}$	drop impact velocity
U_f	$\mathrm{m\,s^{-1}}$	droplet velocity
We	-	Weber number
Y	K	calculated temperature in Levenberg Marquardt method

Greek Letters

Symbol	Unit	Description
α_k	-	constant in Monde solution
β	°	inclination of probe
η	-	effective substrate ratio
λ	$\mathrm{W\,m^{-1}\,K^{-1}}$	thermal conductivity
μ	$\mathrm{kg\,m^{-1}\,s^{-1}}$	dynamic viscosity
ν	$\mathrm{m^2\,s^{-1}}$	kinematic viscosity
ρ	$\mathrm{kg\,m^{-3}}$	density
σ	$\mathrm{N\,m^{-1}}$	surface tension
θ	K	temperature difference T-T_0
ζ	-	argument of erfc in Gaussian error function method

Subscripts

Symbol	Description
f	fluid
L	loss
m	measured
W	wall
wet	wetted substrate
sat	saturated

1 Introduction

Spray cooling is one of the most efficient cooling methods [5] in various sectors of mechanical engineering, ranging from vehicle and reactor cooling to metal production and aerospace applications [6]. Spray cooling can achieve a very high, nearly uniform heat flux and thus reduce thermal loads [7]. High thermal loads can massively restrict a component's functionality or damage and even destroy components. Therefore cooling and a realistic estimation of dissipated heat is of utmost importance. Drop impact phenomena can influence the heat flux absorbed by sprays, therefore drop impact interactions should be investigated. In order to investigate physical interactions in spray cooling, single drop impact needs to be investigated and then superposed according to stochastic models (as in e.g. [2]). By reducing spray to one dimension droplet chains can be investigated. They are described by droplet diameter D_0, velocity U_0 and frequency f. To understand single drop impact fully, both empirical and analytical solutions for varying parameters should be consulted. By formulating the first law of thermodynamics at the drop impact surface one can obtain the heat flux absorbed by the droplet as the heat flux emitted by the underlying material. Therefore in order to calculate the absorbed heat flux by a droplet upon impact on a heated surface, traditional methods of calculating the heat flux in a solid can be used. By placing thermoelements within a probe and solving the inverse heat conduction problem the surface temperature and heat flux of the metal body on which the drop falls can be calculated. 3-dimensional temperature gradients within the metal probe allow the calculation of the heatflux within the metal body. This heatflux at the upper surface of the metal probe is then the heat flux absorbed upon drop impact.

This thesis aims to find the heat flux from a metal probe to an impinging droplet. Due to the experimental set-up the wall temperature can't be directly measured, so an inverse heat conduction equation will be solved numerically to estimate the exact wall temperature and then the corresponding heat flux. In the following Chapter fundamentals of droplet wall interactions and non-isothermal drop impact are discussed. Then the basics of applying drop impact findings to sprays are described. The fundamentals of heat conduction are described, followed by the experimental set-up used to obtain the temperature measurements. After the experimental set-up, a generic test-case for validating surface temperature and heat flux calculations is introduced. Consequently, methods of calculating the heat flux from temperature measurements are explained and applied for our set-up. Finally, the calculated results are discussed for both the test-case as well as the measurement data. An outlook on future work in this field as well as encountered problems conclude this thesis.

2 Theoretical Background

To correctly assess the heat flux, fundamentals from both fluid mechanics and thermodynamics shall be shortly summarized. Firstly, the droplet and its hydrodynamical properties as well as conventional classification numbers are introduced. Then isothermal drop impact shall be explained and physical interactions mentioned. For heated surfaces more phenomena can be observed, which will be introduced afterwards. This includes boiling regimes for water. The concepts will be transferred to 3-dimensional spray with a short description of existing heat flux calculation models. For a heated surface and its droplet interaction the general heat conduction equation is introduced and for our purposes simplified. Finally, inverse and ill-posed mathematical problems are introduced.

2.1 Droplet Parameters

Droplets are mainly characterized by their initial diameter D_0 and impact velocity U_0. The following figure shows the droplet's velocity U_f, initial temperature T_0, initial diameter D_0 and the surface temperature T_W.

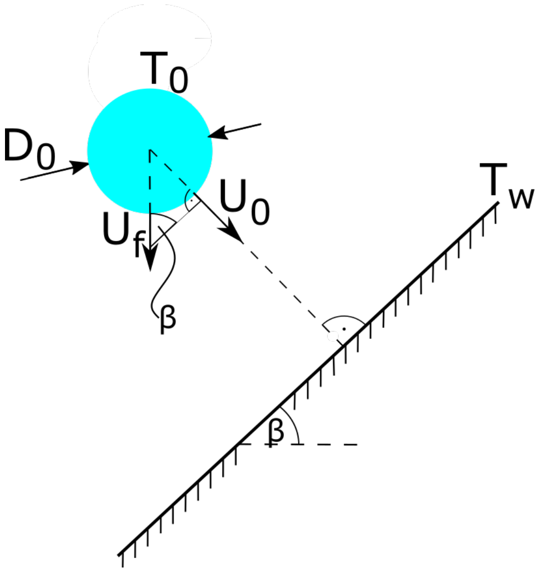

Figure 2.1: Droplet parameters

The impact velocity U_0 can be calculated using $U_0 = sin(\beta)U_f$ with inclination of the surface β. The surface is tilted relative to the droplet chain, so fluid can leave the impact area and a drop does not influence the subsequent one. The Reynolds number

$$Re = \frac{\rho_f D_0 U_0}{\mu} \tag{2.1}$$

where ρ_f and μ are the droplet's density and dynamic viscosity is usually used in order to compare droplets. The Reynolds number describes the ratio of inertial to viscous tensions in a fluid. The Weber number We relates inertial to surface tension of a fluid and is defined as

$$We = \frac{\rho_f D_0 U_0^2}{\sigma} \tag{2.2}$$

where σ is the droplet's surface tension. The Prandtl number Pr of a fluid is defined as

$$Pr = \frac{\nu}{a} \tag{2.3}$$

using the kinematic viscosity $\nu = \mu/\rho_f$ and thermal diffusivity a. An interpretation for the Prandtl number is the ratio of momentum diffusivity to thermal diffusivity.

2.2 Isothermal Drop Impact

Before one investigates drop impact on a heated surface, it is helpful to gain some insight in the isothermal case. Drop impact outcomes are mainly governed by the surface texture (i.e. wettability and roughness) and the hydrodynamics of the droplet [1]. The six different impact outcomes are depicted in Fig. 2.2. The deposition outcome is characterized by two stages. First the kinematic and then the actual deposition [8]. Hydrodynamic properties of the droplet and the physical parameters of the surface do not affect the droplet in the kinematic stage. Later the droplet spreads up to five times its inital diameter depending on its physical properties [1]. Empirical correlations can be found to describe the spreading to inital diameter ratio.

With increasing velocity the droplet's morphology changes and develops the prompt splash scenario where secondary droplets form due to the roughness of the wall. As surface tensions decrease, the lamella of the splash can detatch from the wall, therefore creating the corona splash outcome. Corona splash is also affected by the surrounding pressure and can be suppressed [**?**].

Furthermore, wettability and viscosity of the droplet influence its impact outcome. For low-viscosity fluids at high velocity the lamella overshoots the corona and starts to recede. Receding break-up is characteristic for nonwettable surfaces and the "fingers" of a lamella can further break-up due to capillary instability. If the kinetic energy is large enough, the receding break-up does not take up all the droplets energy and a secondary rebound can form. This is common for strictly nonwettable surfaces. An arising liquid column can partly or completely detach. Parts forming smaller secondary droplets as a partial rebound outcome and complete detachment in unison is shown in the last line of Fig. 2.2 denoted as complete rebound.

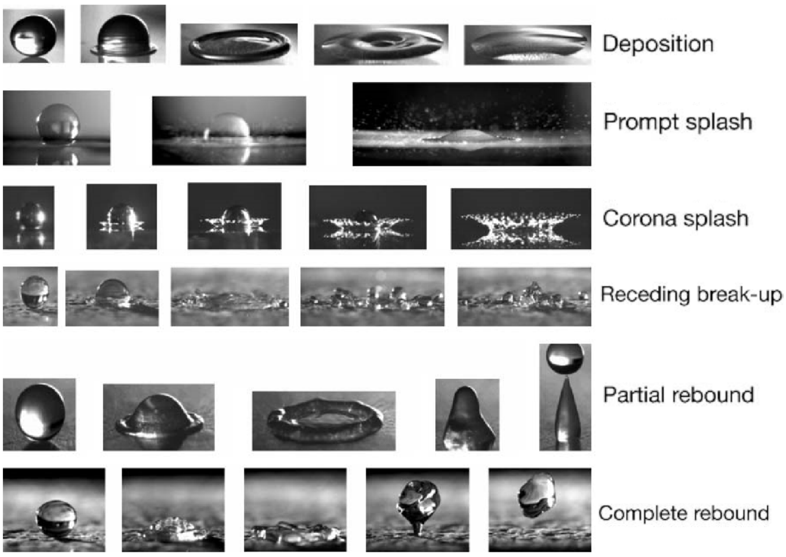

Figure 2.2: Outcomes for isothermal drop impact, adapted from [1]

2.3 Non-isothermal Drop Impact

For an initial temperature not equal to the wall temperature, the drop impact is considered non-isothermal. Non-isothermal drop impact can be classified as with and without boiling. The first relies completely on heat conduction as a means of heat transfer [2]. Different boiling regimes as well as varying impact velocity U_0 can lead to various droplet-wall interactions. Before investigating non-isothermal drop impact outcomes, it is helpful to look at the different boiling regimes and new physical phenomena which do not occur for isothermal drop impact.

2.3.1 Boiling Regimes

Single drop impact outcomes can be categorized in the heat transfer regimes of single-phase cooling, nucleate boiling, transition boiling and film boiling. These regimes are bounded by the critical heat flux and the minimum heat flux at the Leidenfrost Temperature as depicted in Fig. 2.3. The Leidenfrost effect leads to minimal specific heat conduction and should be fully understood in order to avoid this inefficient heat transfer point in cooling applications.

The Leidenfrost effect was first observed and described by J.G. Leidenfrost as a phenomenon where a water droplet was suspended over a heated surface in 1756. The droplet is suspended by its own vapour layer which forms between the surface and the underside of the droplet in the film boiling regime [9]. The droplet slowly evaporates due to heat conduction across its vapour cushion, heat radiation from the surface and molecular diffusion over the top of the droplet. As calculated numerically [9], experimental data only agrees with the analytical solution when radiation and diffusion are taken into account. The heat flux is at a local minimum for the Leidenfrost point, so conduction is relatively low in comparison with other saturation temperatures. Therefore, this does not contradict the general assumption of mainly heat conduction for the heat flux calculation.

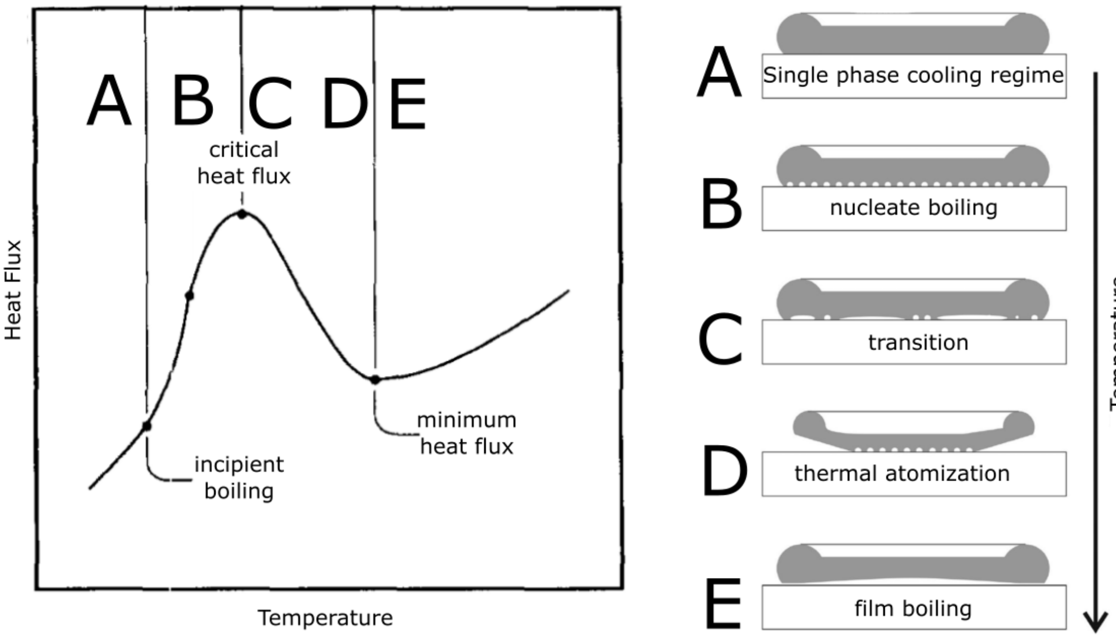

Figure 2.3: Nukiyama curve for boiling in droplet chains showcasing different boiling regimes, adapted from [2]

Fig. 2.3 separates the different parts of the boiling curve as well as the different boiling regimes in sections A to E. Section A shows the single phase cooling regime where no boiling takes place. The heat transfer is purely conductive [2] via the molecular vibrations and no bubbles form.

The incipient boiling point as depicted in Fig. 2.3 marks the beginning of the nucleate boiling regime, here denoted as B. Between surface and droplet bubbles nucleate evenly over the whole wetted area. These small spherical bubbles rise through the droplet. The end of this boiling regime is the critical heat flux which marks the highest specific heat transfer. Afterwards heat transfer declines due to vapour that partially forms in section C. The bubbles from section B unite to bigger bubbles as depicted in the right hand side of Fig. 2.3 during transition boiling. These bubbles form an insulative layer. A partial vapour film inhibits effective heat transfer and the specific heat flux declines.

Thermal atomization is a recent discovery. The droplet only wets a smaller area of the surface in comparison to the other regimes. Therefore the heat flux is significantly lower even though no insulative layer exists. An image of thermal atomization is included in Fig. 2.4.

The Leidenfrost point marks the beginning of the last boiling regime: Film boiling, here section E. Due to the rising temperature difference between droplet and surface, the specific heat flux starts to increase again. The partial vapour layer that formed in section C is fully occupying the contact area between droplet and surface. This considerably lowers the heat flux in comparison to the nucleate boiling regime. The increase in the heat flux after the minimal heat flux is because of an increasing proportional amount of radiation as means of heat transfer.

2.3.2 Non-isothermal Drop Impact Outcomes

Non-isothermal drop impact outcomes can be separated in drop deposition, drop dancing, thermal atomization and drop rebound and are depicted in Fig. 2.4. Drop deposition occurs for the isothermal case, the single-phase cooling regime as well as nucleate boiling. As depicted in Fig. 2.4, the droplet deposits after an initial lamella spreading on the surface. The temperature threshold for this outcome is approximately $150\ ^{\circ}C$ for all impact velocities $U_0 \in [0.4, 2]\ m/s$.

Drop dancing can be observed in the transition boiling regime. Here the drop produces secondary droplets upon impact which appear to “dance” on the surface due to vapour bubbles and frequently collapsing liquid layers [2]. This happens for a surface temperature between $150\ ^{\circ}C$ and $200\ ^{\circ}C$. A higher impact velocity leads to earlier thermal atomization, as shown in Fig. 2.4. Thermal atomization occurs for higher heat transfer rates. The lamella can levitate above the surface, as depicted in Figure 2.3 section D on the right hand side. This outcome happens for lower impact velocities and high temperatures. High impact velocities lead to the last outcome: Complete or partial rebound. Rebound appears at high temperatures in the film boiling regime where an instantaneous vapour layer separates the droplet and the surface. Usually the Weber number is used as a dimensionless parameter to determine the drop impact outcome for different temperatures. Here this approach is forgone for the dimensional impact velocity. This is due to the fact, that the Weber number is not sufficient to describe the impact outcome at varying temperatures since it neglects thermodynamic properties[10]. For non-isothermal drop impact not only the outcome, but also the heat flux upon impact is of interest.

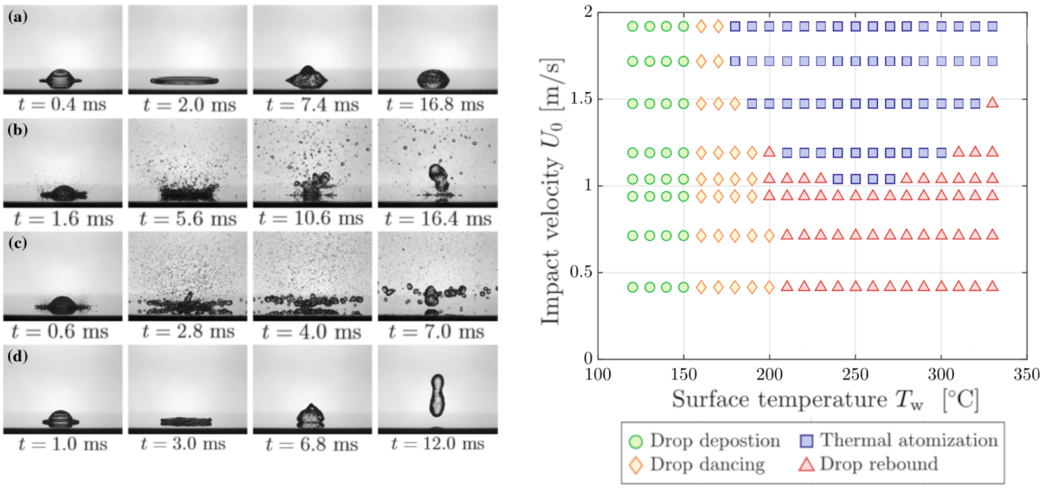

Figure 2.4: Outcomes for isothermal drop impact a) drop deposition, b) drop dancing, c) thermal atomization and d) drop rebound, adapted from [2]

2.3.3 Heat Flux in Droplet Chains

The specific heat transfer for non-isothermal drop impact can be calculated according to [11] as

$$\dot{q}(t) = \frac{\lambda_f \lambda_W (T_{w0} - T_{d0})}{\sqrt{a_f a_W \pi t}(\lambda_f/\sqrt{a_f} + \lambda_W F(Pr)/\sqrt{a_W})} \tag{2.4}$$

using thermal diffusivity a and thermal conductivity λ for fluid and surface. F(Pr) is a dimensionless function of the Prandtl number Pr which determines the degree of influence of the thermal effusivity of the wall. T_{w0} and T_{d0} are the initial temperatures of surface and drop. F(Pr) was approximated by [11] to be

$$\mathrm{F}(Pr) = 1 - \frac{0.28}{(0.27\sqrt{Pr} + 0.45)^{0.28}}. \tag{2.5}$$

Eq. 2.4 uses the surface temperature of the metal probe, which is not easily obtained. The methods used in this thesis focus on calculating the heat flux based on the metal probe and temperature measurements within the probe body. Before the thermodynamic fundamentals are introduced it is necessary to apply drop impact findings to sprays.

2.4 Spray Cooling

2.4.1 Spray Parameter

In general spray parameters are an enhancement of already existing drop or drop chain parameters. Because of fluctuations in size and velocity the mean diameter D_0 and U_0 are used. Frequency is replaced by mass flux, usually per area. This results in the mean mass flux density j_m.

2.4.2 Nukiyama Curve for Sprays

The Nukiyama curve can also be plotted for sprays while data points remain mostly experimental. Stochastic descriptions of spray parameters remain dominant and empirical correlations are used to calculate heat flux [6].

The different boiling regimes in spray cooling applications are described by [12]. In the film boiling regime spray cooling leads to a uniform cooling of the surface. In the transition boiling regime after passing the Leidenfrost point, the specific heat flux increases considerably. This can mean different boiling regimes at the same time for complex geometries which will not be discussed any further. Due to the high specific heat flux the transition boiling regime changes to the nucleate boiling quickly. Close to the critical heat flux cooling continues rapidly and reaches the single phase cooling regime. In quenching processes the mechanical properties can be determined until the critical temperature range. More details about the temperature distribution can be found in [12].

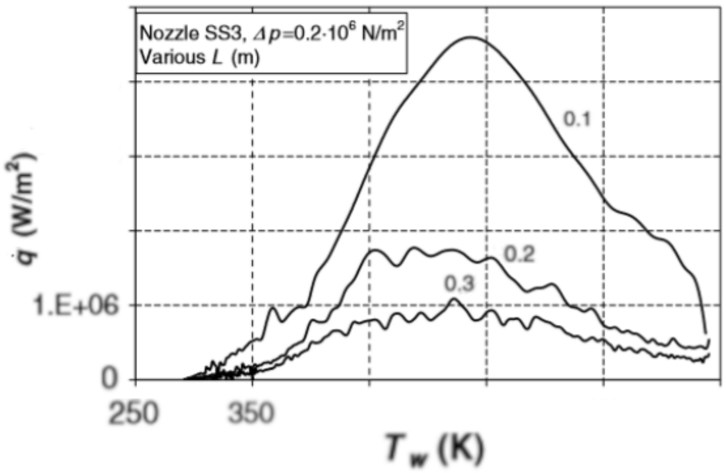

Figure 2.5: Nukiyama curve for sprays, adapted from [3]

2.4.3 Heat Flux Models in Spray Cooling Regimes

For the film boiling regime in spray cooling a heat flux calculation model already exists. The proposed model by [2] states

$$\dot{q}_{spray} = 28.32 \frac{j_m G \lambda_w (T_{w0} - T_{sat})}{\sqrt{a_w D_0 U_0} \rho_f (K + 2G)} \eta_{wet}. \tag{2.6}$$

Here, j_m is the mass flux of the spray and η_{wet} is the effective wetted substrate ratio. This is calculated using the Poisson distribution and therefore defined as

$$\eta_{wet} = \frac{1 - e^{-\lambda}}{\lambda}, \tag{2.7}$$

where the wetted area is estimated using the empirical correlation

$$\lambda = \frac{2.1 j_m}{\rho_f U_0} (1 + 0.36 We^{0.48})^2. \tag{2.8}$$

Furthermore, K and G are parameters defined as follows:

$$K = \sqrt{(B - G)^2 + \frac{4G}{\sqrt{\pi}}} - B - G \tag{2.9}$$

$$G = \frac{\sqrt{\pi} \lambda_v \rho_f L a_w}{2(T_{w0} - T_{sat}) \lambda_w^2} \tag{2.10}$$

$$B = \frac{\sqrt{5a_w}(T_{sat} - T_{d0})\lambda_f}{\sqrt{\pi a_f}(T_{w0} - T_{sat})\lambda_w}. \tag{2.11}$$

2.5 General Heat Conduction Equation

Heat transfer between fluid and wall is mainly conduction [13] and therefore governed by the general heat conduction equation

$$\frac{d}{dt}T - a\Delta T = \frac{\dot{Q}}{c_p\rho} \tag{2.12}$$

where T is the temperature function with respect to time t and location, either in carthesian or polar coordinates, $\dot{Q}$ is the heat generated within the solid, c_p is the probe's specific heat capacity at constant pressure and ρ is its density. In our current set-up no heat is generated after the probe has reached its target temperature, from which the cooling experiment starts, therefore Eq. 2.12 can be simplified to

$$\frac{d}{dt}T - a\Delta T = 0. \tag{2.13}$$

$$\frac{d}{dt}T - a(\frac{d^2}{dx^2}T + \frac{d^2}{dz^2}T) = 0 \tag{2.14}$$

with boundary conditions $T = T_0$ at $t = 0$, $\frac{d}{dx}T = 0$ at $z = 0$.

2.6 Inverse Problems

Classical problems of heat conduction theory are assumed to be known and can help to distinguish different classes of inverse problems. Each direct problem which uses a mathematical model can be compared with a certain set of inverse problems. All problem statements of heat transfer between a solid body or between a surface and an impinging droplet are considered from a "cause-effect" standpoint [14]. According to the mathematical model, boundary conditions, initial conditions, thermophysical properties, internal sources of heat and conductivities and the geometry can be classified as causal characteristics of heat transfer. The effect is a heat state which is determined by the temperature field of the substrate.

Direct heat transfer problems specify the cause-and-effect relationship. They calculate the temperature distribution for a given set of causal characteristics. Inverse problems try to recover the causal characteristics of a temperature field. In an actual experiment these characteristics cannot be reproduced since it is impossible to reverse the cause-and-effect relationship. Mathematically it is possible to reverse the problem solving process. Inverse problems can be divided in

- problems in diagnostics and identification of physical processes;
- problems in the design of engineering products;
- problems in controlling processes and systems.

A problem where the data is given and the right-hand side of the equation is unknown is considered "inverse". Inverse problems of the first class are usually experimental research. Here it is necessary to reconstruct causal characteristics on the basis of measured "output" effect characteristics. These problems are connected with the construction of the mathematical models and determination of different characteristics of the models. Design problems consist of determining design characteristics of an engineering unit on the basis of given quality indices with certain limits. The required characteristics are causal with respect to these indices and limits. Control influences create the causal characteristics which creates the control action expressed by the effect (the system state).

2.7 Well-posed and Ill-posed Problems

A problem is considered well-posed if three criteria are fulfilled [15]. In our case Eq. 2.12 can be used to explain the three requirements:

- Eq. 2.12 has a solution for any right hand side $\dot{Q}$;
- the solution to Eq. 2.12 is unique;
- the dependence of the solution on $\dot{Q}$ is continous, i.e. when the error in $\dot{Q}$ tends to zero, the error in the solution also tends to zero.

Else the problem is considered ill-posed. The three requirements can be summarized as existence, uniqueness and continuity of a solution. Continuity is related to stability. It is necessary for a solution to be continuous in order to be stable, but not all solutions, that are continuous, are stable [16]. A well-posed problem can also be unstable. All classical problems in mathematical physics are well-posed, including the general heat conduction problem 2.12 [16]. In our case, if the temperature (e.g. at given data points) is known through measurements, the unknown is then the heat flux $\dot{Q}$. To calculate $\dot{Q}$ in Eq. 2.12, we need to solve the inverse heat conduction equation. This is also the case for no heat generation within the probe. Inverse problems are usually ill-posed [16].

3 Experimental and Analytical Methods

This chapter describes the experimental set-up that was used to obtain the data points used in the heat flux calculation. Furthermore the heat conduction equation is simplified for our present example. Then a generic test case is described which was used to validate the new inverse heat flux calculation methods described in the next chapter.

3.1 Experimental Set-up

The experimental set-up is depicted in Figure 3.1 and consists of a drop generator, an observation and evaluation unit as well as the heated probe. Drops varying in diameter, velocity and frequency are generated by a hollow needle of size G 27. The needle is blunt and has a hydrophobic coating. Its outer diameter is 0.4 mm. The inner needle diameter determines the droplet diameter D_0. The needle is placed vertically above the probe. In order to increase the impact velocity U_0, the height of the needle can be adjusted. A Bartels micropump mp 6 provides the mass flux and regulates the droplet frequency with increasing mass flux.

The observation and analysis unit consists of a Phantom v12 high speed camera, a LED lamp for illumination at the time of impact, 11 thermoelements of type J class 1 in the probe and a NI DAQ device to save the thermoelements' recorded data. Images and temperature measurements are then transmitted to a PC via USB. The horizontal thermoelements' temperature measurements are called T_1 to T_7, while the radial thermoelements' temperature measurements are called T_{r1} to T_{r4}.

The procedure of the cooling experiment starts with heating the probe to an initial temperature T_0. Then the heater is turned off and the droplet chain cools down the probe. The camera is used to determine the droplet's impact velocity U_0 and initial diameter D_0. The camera also determines the impact regimes which are described in section 2.3. The droplets are recorded using shadowgraphy with a frame rate of 6000 fps and a resolution of 16 $\mu m/$pixel. The drop impact on the heated surface causes a 3-dimensional temperature gradient which can be measured with 3 radially and 8 horizontally placed thermoelements.

The probe is a polished stainless steel 1.4841 cylinder with thermoelements and a heating spiral inside, as depicted in Fig. 3.2.

The probe's density, specific heat and thermal conductivity are $\rho = 7900\frac{kg}{m^3}, c_p = 542\frac{J}{kgK}$ and $\lambda = 16\frac{W}{mK}$ respectively. Specific heat at constant pressure c_p and thermal conductivity λ are averaged from $100°C$ to $500°C$. A more detailed description of the experimental set-up can be found in the Thesis from Quell [4].

The impinging droplets cause a measurable 3-dimensional temperature gradient within the probe. Making use of the horizontal and radial thermoelements the temperature within the probe can be obtained. This allows the calculation of the surface temperature and furthermore the heat flux. To obtain surface temperature and heat flux for validation purposes, it is helpful to discuss a finite element simulation.

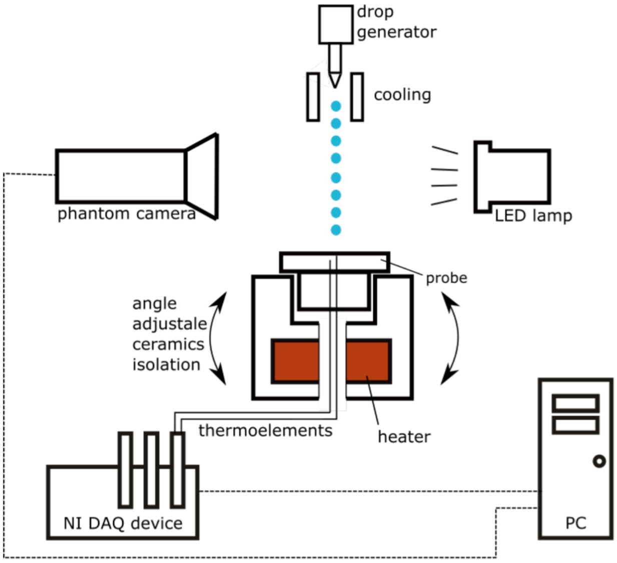

Figure 3.1: Experimental set-up, adapted from [4]

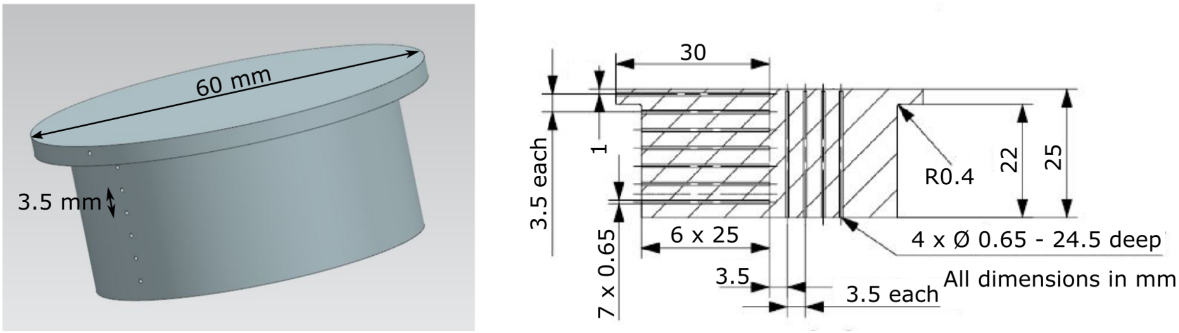

Figure 3.2: Probe geometry, adapted from [4]

3.2 Generic Test Case

In order to validate the new methods described in Chapter 4 and also to compare them to existing methods of solving the heat conduction equation as listed in Chapter 3 a test case was developed. The advantage of this test case in comparison to the real measurement data is, that the wall temperature as well as the heat flux is known and can be compared to the results of the respective calculation method. The test case was modeled as similarly as possible to our real probe. The probe body is represented by a cylinder of height $L = 25\ mm$ and radius $R = 25\ mm$ A temperature distribution on the surface was given as a boundary condition as well as adiabatic walls. By using a finite element method the temperatures at all thermoelements' location could be calculated. The parameters are delineated in Table 3.1.

parameter	value	description
ρ	7900 kg m^{-3}	density
λ	16 W m$^{-1}K^{-1}$	thermal conductivity
c$_p$	542 J kg$^{-1}K^{-1}$	specific heat
h$_{mesh}$	0.5 mm	grid spacing
Δt	0.0025 s	time step
t$_{final}$	60 s	duration

Table 3.1: FEM parameters

3.2.1 Linear Temperature Decrease

The simplest test case is depicted in Fig. 3.3 and consists of merely a linear temperature decrease. Due to computing time the test cases were only simulated for $t_{final} = 60\ s$. As expected, the temperature at each thermoelement's depth T_i follows the temperature decline with the increasing depth more strongly dampening the temperature drop. While the first thermoelement T_1 at depth $z = 1\ mm$ follows the wall temperature T_W closely, the deeper thermoelements T_2 and T_7 are affected by the temperature drop only after a delay. The temperature gradient seems to reach T_7 only after $20\ s$. The heat flux $\dot{q}$ is negative, because heat is emitted. The difference between T_W and T_1 slightly increases over the course of the experiment. This explains, why the absolute value of the heat flux also increases. The thermoelements are enumerated horizontally as depicted in 3.2.

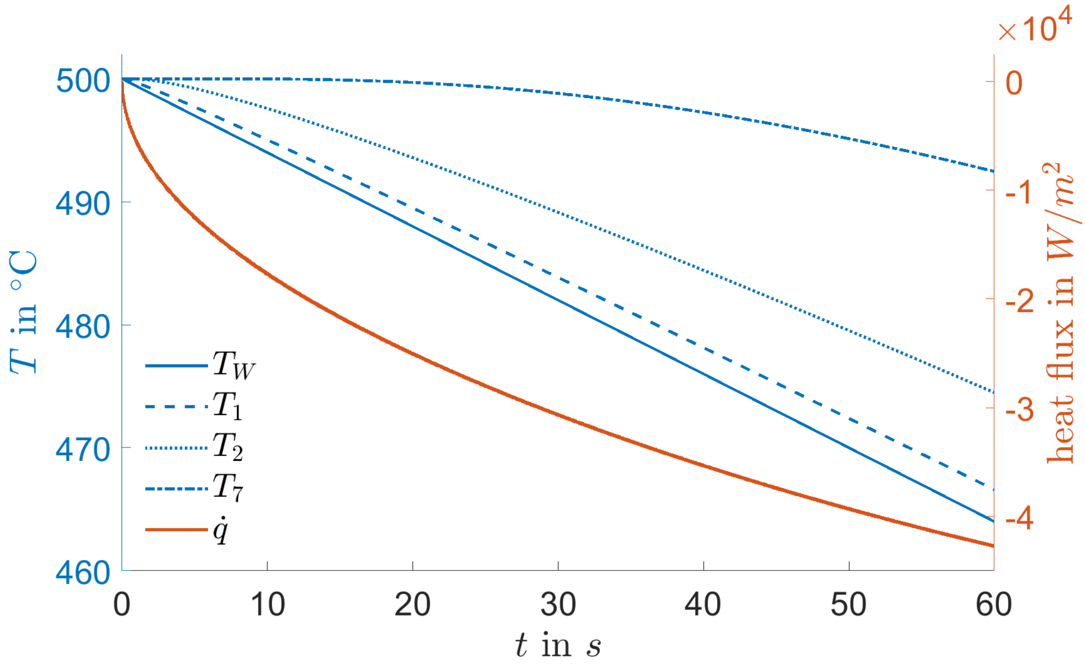

Figure 3.3: Temperature decline and heat flux in linear test case

3.2.2 Cosinusoidal Temperature Fluctuation with linear Decrease

The second test case integrates a characteristic feature of the measurement data. Each drop impact causes a visible decline in temperature for T_1 which then rises again due to reheating of the probe. The subsequent drop is visible and the cosinusoidal graph has a decreasing mean because of cooling. The advantage of using cosine instead of sine is, that there is no discontinuity at the beginning. Similarly to the linear case, Fig. 3.4 shows the temperature profile in the first $10\ s$ and the last $10\ s$ of the simulation.

The temperature at the depth of thermoelement 1 is not exactly in the same phase as the wall temperature. This can be explained through a finite information velocity through the probe. The temperature decline over $1\ mm$ is visibly influenced by the probe's depth. Due to the high frequency of drop impact we shall first look at Fig. 3.4, since individual drop impact can be clearly distinguished for this interval. At thermoelement 2 which is only $4.5\ mm$ distanced from the surface almost no oscillation can be observed. For deeper thermoelements the same behavior can be observed, their depiction is omitted for clarity. Thermoelement 7 sees no temperature drop over the course of $12\ s$. The global trend shows a drop of $\Delta T = 40°C$ for the surface and $\Delta T = 10°C$ at the depth of thermoelement 7. The heat flux is continously oscillating and has a constant mean.

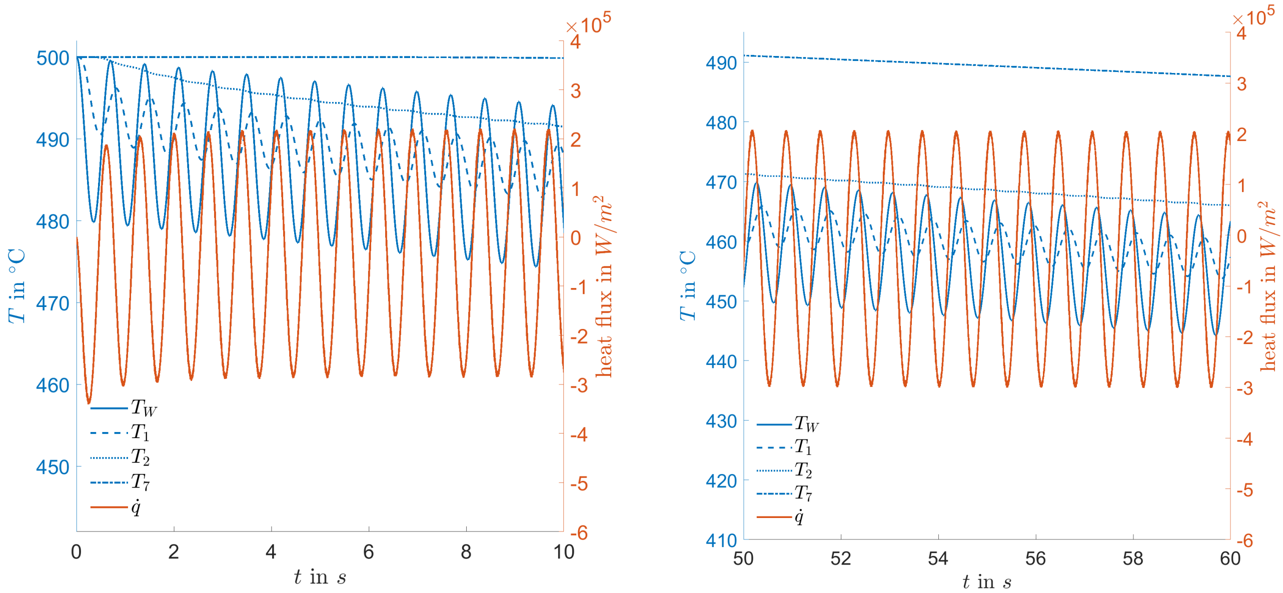

Figure 3.4: Temperature decline and heat flux in cosinusoidal test case

3.2.3 Sinusoidal Temperature Decrease with initial Discontinuity

The final test case uses a sine instead of cosine which leads to an initial temperature gradient. Also, a delay in the response of the thermoelements is taken into account. Similarly to the cosine case, Fig. 3.5 shows a distinguishable trend for the temperature and the heat flux is depicted in Fig. 3.6. The highest absolute value of the heat flux is $\dot{q}_{max} \approx -2 \cdot 10^{-6}\ W/m^2$. This is also omitted in Fig. 3.7 in favour of clarity.

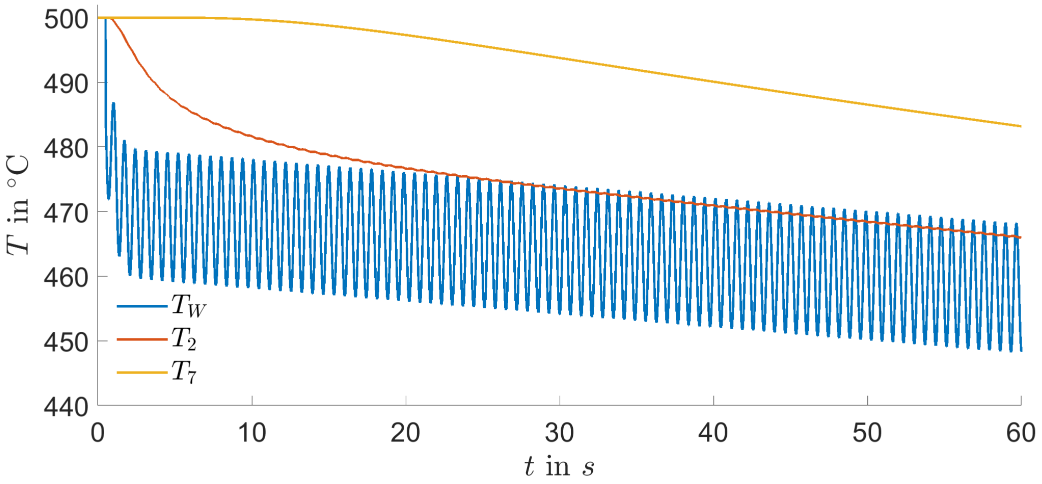

Figure 3.5: Temperature decline in final test case

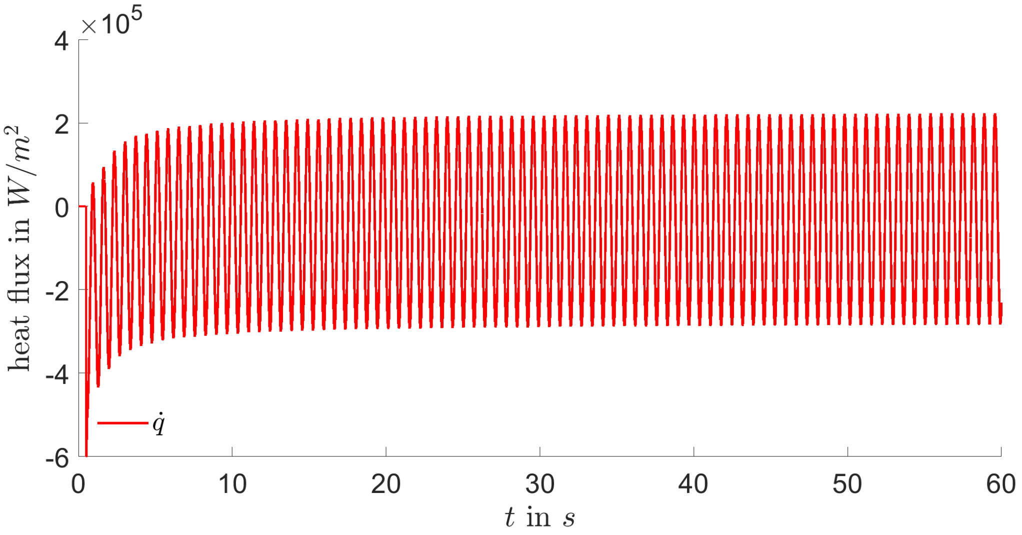

Figure 3.6: Heat flux in final test case

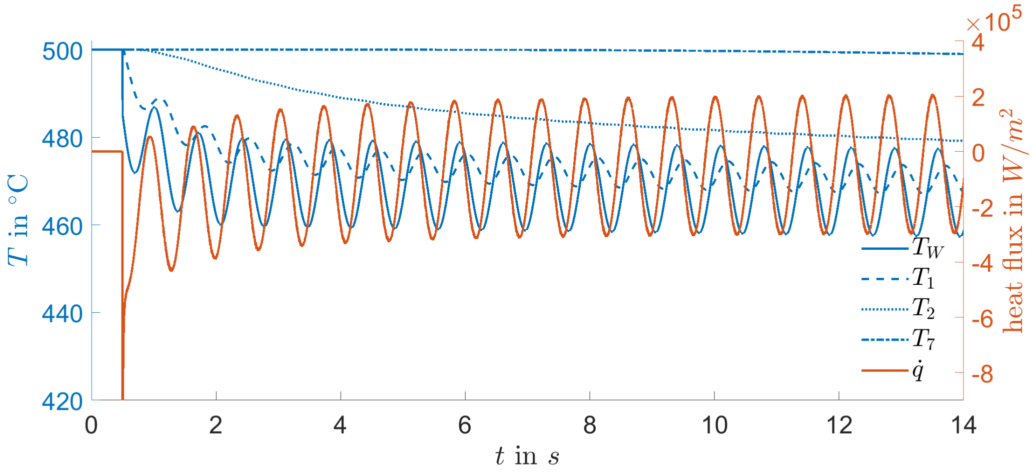

Figure 3.7: Temperature decline and heat flux in final test case

4 Solving the inverse Heat Conduction Equation

Various methods exist to solve the heat conduction equation. The heat conduction equation can be well- or ill-posed depending on the existence of a unique solution for any heat flux (i.e. right hand side) with stability for perturbations of the right hand side of the equation. The basic idea of some of the solving methods shall be briefly described in the following subsections. The coordinate system used within the substrate is depicted in Fig. 4.1. The application of methods introduced in this chapter are reported in Chapter 5 for the earlier introduced test case as well as real measurement data from the experimental set-up.

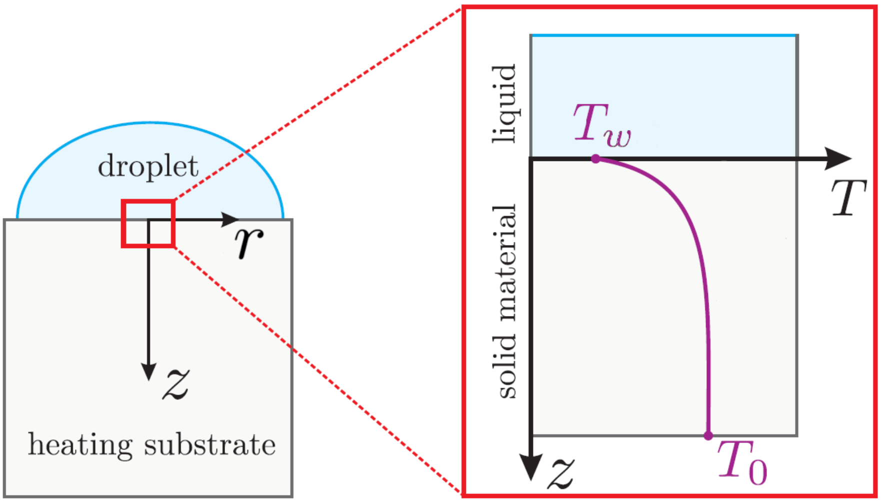

Figure 4.1: Probe with impinging droplet, adapted from [5]

4.1 Existing inverse Heat Flux Calculation Methods

The following section provides a brief introduction to already existing inverse solving methods for the heat conduction equation. The solution methods were firstly postulated by Burggraf [14], Monde [17] and Levenberg Marquardt [15].

4.1.1 Burggraf Solution

For the 1-dimensional case Burggraf obtained an exact solution using an infinite sum of functions f_n and g_n whose second derivatives are either 0 or the slope of the previous function. Then these functions are combined with a measurement of temperature and heat flux within the solid at one single point. No initial condition is necessary and this solution can be cut off for a precise approximation. The series expansion of the following form satisfies the heat conduction equation 2.14.

$$T(r,t) = \sum_{n=0}^{\infty} f_n(r) \frac{d^n t_0}{dt^n} - \frac{1}{a} \sum_{n=0}^{\infty} g_n(r) \frac{d^n q_0}{dt^n} \tag{4.1}$$

as long as the boundary conditions are met for f_n:

$$\Delta f_0 = 0, \Delta f_n = \frac{1}{a} f_{n-1} \tag{4.2}$$

for n = 1,2,3,... .

The solution for the solid cylinder is given in Eq. 4.3.

$$T(r,t) = T_0(t) + \sum_{n=0}^{\infty} \frac{r^{2n}}{2^{2n}(n!)^2 a^n} \frac{d^n T_0}{dt^n} \tag{4.3}$$

4.1.2 Analytical Solution using Laplace Transform

In the 2-dimensional case an analytical solution can be obtained using the Laplace transform. Inverting the Laplace-transformed function with respect to time-space variables can be difficult and therefore approximations are used. The following procedure provides a solution to the inverse heat conduction problem as in [**?**] for the 2-dimensional case in carthesian coordinates. The solution to Eq. 2.14 can be given using a half-power polynomial function $F_j(t)$ for $\theta = T - T_0$:

$$\theta = T - T_0 = \sum_{j=0}^{J} F_j(t) cos(j\pi x/X) \tag{4.4}$$

The half-power polynomial function $F_j(t)$ is multiplied by the Gamma-function for an easier Laplace-transform.

$$F_j(t) = \sum_{k=0}^{K} P_{j,k} t^{k/2} / \Gamma(1 + k/2) \tag{4.5}$$

The Laplace-transform of θ yields

$$\theta = \sum_{j=0}^{J} cos(j\pi x/X) F_j(s) \frac{sinh(\sqrt{s/a + (j\pi)^2/X^2}(z_2 - z))}{sinh(\sqrt{s/a + (j\pi)^2/X^2}(z_2 - z_1))} \tag{4.6}$$

with z-coordinates z_1 and z_2 for the temperature sensors' depth in the probe. This can be inversely Laplace-transformed using tables to get the solution to Eq. 2.14.

$$T(r,t) - T_0 = \sum_{j=0}^{J} cos(j\pi x/X) \left(\sum_{k=-1}^{K} G_{j,k}^{(12)} t^{k/2} / \Gamma(1 + k/2) \right) \tag{4.7}$$

The coefficients $G_{j,k}^{(12)}$ can be determined as follows:

$$\sum_{k=-1}^{K} G_{j,k}^{(12)} / s^{(k+2)/2} = \left(\sum_{p=0}^{K} P_{j,p}^{(1)} / s^{(p+2)/2} \right) \left(\sum_{q=0}^{(K+2)/2} C_{j,q}^{(2)} s^q \right) \tag{4.8}$$

$$C_{j,0} = \alpha_0 / b_0 \tag{4.9}$$

$$C_{j,k} = \frac{1}{b_0} \left(\alpha_k - \sum_{q=0}^{k-1} C_{j,q} b_{k-q} \right) \tag{4.10}$$

$$x_p = \left(\frac{1}{p!} \frac{d^p}{ds^p} (sinh(\sqrt{s/a + (j\pi)^2/X^2}(z_2 - Z))) \right) |_{s=0}. \tag{4.11}$$

for $x = \alpha$: $Z = z$ and $x = b$: $Z = z_1$. Further simplifications for e.g. Eq. 4.11 can be found in [?].

4.1.3 Levenberg Marquardt Method

The Levenberg Marquardt method is an iterative numerical procedure that successively minimizes the distance between measured and calculated temperature. The function describing the squared distance between measured and calculated temperature D(P) can be defined as follows according to [18]:

$$D(P) = \sum_{i=1}^{I} [T_m(t_i) - Y(P, t_i)]^2 \tag{4.12}$$

with I measurements for time steps $t_i, T_m(t_i)$ measured temperatures, $Y(P, t_i)$ calculated temperatures using the numerical solution and P parameters to determine $\dot{q}$. The heat flux can be approximated as a cubic spline. The resulting parameters for the interpolation of the cubic spline are P. This method was originally developed to solve ill-posed problems, but can be applied for inverse problems using the Jacobi matrix of T(P). The Jacobi matrix partially differentiates the splines of $T_i(P)$ with respect to the parameters P_j. J is the Jacobi matrix containing all cells of $J_{ij} = \frac{\partial T_i}{\partial P_j}$. According to [19] the solution to the matrix P can be calculated using

$$P = (J^T J)^{-1} J^T T_m \tag{4.13}$$

The solution to $Y(P, t_i)$ can be iteratively found using the Taylor approximation:

$$Y(P) = Y(P^k) + J^k (P - P^k) \tag{4.14}$$

for k iterations. Substituting 4.13 in 4.14 yields

$$P^{k+1} = P^k + [(J^k)^T J^k]^{-1} (J^k)^T [T_m - Y(P^k)]. \tag{4.15}$$

4.2 New inverse Heat Flux Calculation Methods

This section deals with new methods developed by Apl. Prof. Dr. Ilia Roisman. The following subsections introduce the theoretical derivation of each method. The first two methods simplify Fourier's Heat Conduction Equation and solve it using Taylor-Polynomials to calculate the unknown wall temperature T_W. From the solution of the inverse wall temperature problem the specific heat flux $\dot{q}$ can be obtained. The last method uses the analytical solution to the Heat Conduction Equation, ergo the complimentary Gaussian Error Function.

4.2.1 Heat Conduction Equation

The underlying equation for this method is the general heat conduction equation 2.12 as introduced in Chapter 2. For the 1-dimensional case without heat generation inside the solid body Eq. 2.12 simplifies to Eq. 4.16.

$$\frac{d}{dt}T - a\frac{d^2}{dz^2}T = 0 \tag{4.16}$$

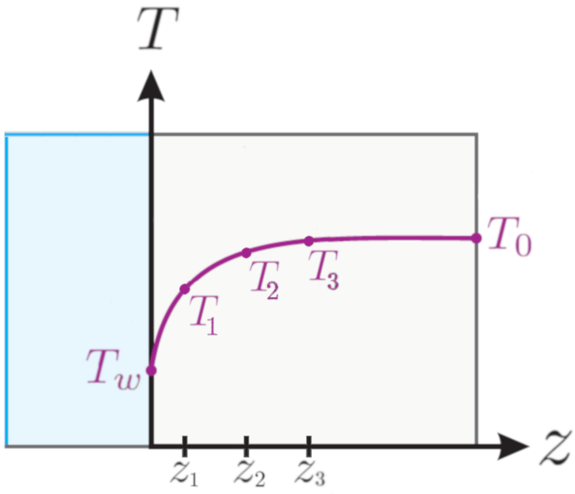

Figure 4.2: Change in temperature over depth

This can be evaluated at the depth of the first thermoelement $z = z_1 = 0.1\ mm$, therefore we get Eq. 4.17

$$\frac{d}{dt}T|_{z=z1} - a\frac{d^2}{dz^2}T|_{z=z1} = 0 \tag{4.17}$$

with help of a Taylor expansion at the surface and the depth of Thermoelement 2

$$T_W = T_1 - T'z_1 + \frac{T''}{2}{z_1}^2 \tag{4.18}$$

$$T_2 = T_1 + T'(z_1 - z_2) + \frac{T''}{2}(z_1 - z_2)^2 \tag{4.19}$$

where T' denotes the spatial differentiation dT/dz. Due to the fact that both difference quotients are evaluated at $z = z_1$ we can solve for T'' and get Eq. 4.20

$$T'' = \frac{2[T_2 z_1 - T_1 z_2 + T_w(z_2 - z_1)]}{(z_2 - z_1)^2 z_2}. \tag{4.20}$$

This can be inserted in Eq. 4.17 and solved for the unknown wall temperature T_W

$$T_W = \frac{\frac{1}{2a}\frac{d}{dt}T|_{z=z1}(z_2 - z_1)^2 z_2 - T_2 z_1 + T_1 z_2}{(z_2 - z_1)}. \tag{4.21}$$

All variables $T_{1,2}$ in Eq. 4.21 are known as discrete measurements and only functions of time which also allows to numerically compute the temporal gradient $dT/dt|_{z=z1}$.

Knowing T_W allows the the calculation of Q and later $\dot{q}$ as depicted in Fig. 4.3.

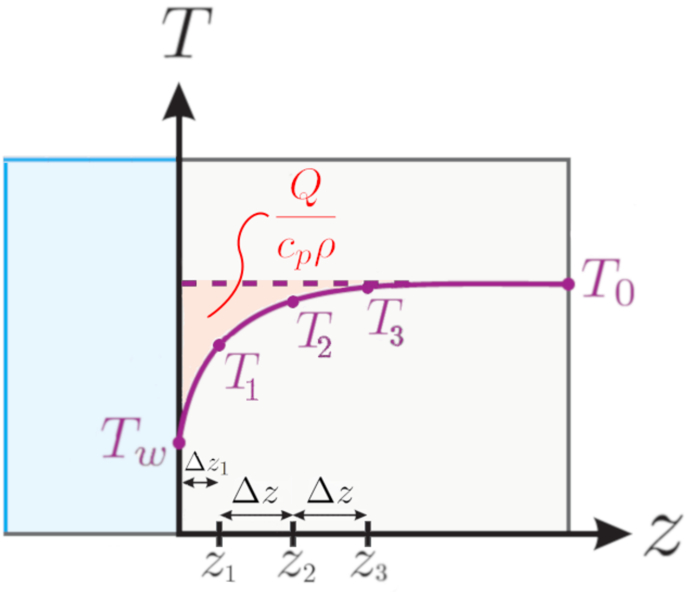

Figure 4.3: Heat Q through change in temperature

The total heat Q can be calculated according to Eq. 4.22

$$Q = c_p\rho \int_{z=0}^{\infty} T_W - T dz \approx c_p\rho[T_W L - \frac{\Delta z_1}{2}(T_W + T_1) - \frac{\Delta z}{2}(T_1 + T_2) - \frac{\Delta z}{2}(T_2 + T_3) - \ldots - \frac{\Delta z}{2}(T_6 + T_7)] \tag{4.22}$$

Now Q is a function of time and $\dot{q}$ can be calculated using Eq. 4.23

$$\dot{q} = \frac{Q(t_i) - Q(t_{i-1})}{t_i - t_{i-1}} \tag{4.23}$$

4.2.2 Gaussian Error Function

Another method is the analytical solution to the homogenous general heat conduction equation 2.13. The Gaussian Error Function $\mathrm{erf}(z)$ is defined as

$$\mathrm{erf}(z) = \frac{1}{\sqrt{\pi}} \int_0^z e^{-x^2} dx \tag{4.24}$$

over the domain $D = (-1, 1)$. The complimentary function is called $\mathrm{erfc}(z)$ and defined as $\mathrm{erfc}(z) = 1 - \mathrm{erf}(z)$.

As described in [7] the complimentary error function $\mathrm{erfc}(z)$ solves the homogenous general heat conduction equation 2.13. The wall temperature $T_W(t, z)$ can be computed using data from any given depth and time as

$$T_W(t, z) = T_0 + \int_0^t A(x)\mathrm{erfc}\left(\frac{z}{2\sqrt{a(t-x)}}\right)dx. \tag{4.25}$$

This analytical solution can be approximated as a Riemann-sum with increasing accuracy for a higher temporal resolution

$$T_W(j, \Delta t, z) = T_0 + \sum_{i=1}^{j-1} A_i \mathrm{erfc}\left(\frac{z}{2\sqrt{a\Delta t(j-i)}}\right)\Delta t. \tag{4.26}$$

Now we introduce a new variable B_i according to Eq. 4.27 in order to get an iterative solution for every consecutive time step. We also introduce the variable ζ for the argument of the error function

$$B_i := \frac{T_{W,j} - T_0}{\Delta t} \tag{4.27}$$

$$\zeta := \frac{z}{2\sqrt{a\Delta t}}. \tag{4.28}$$

With the initial value $A_1 = 0$ according to Eq. 4.26 we can calculate the subsequent B_i explicitly according to Eq. 4.29

$$B_{j+1} = \sum_{i=1}^{j} A_i \mathrm{erfc}\left(\frac{\zeta}{\sqrt{(j-i+1)}}\right) = \sum_{i=1}^{j-1} A_i \mathrm{erfc}\left(\frac{\zeta}{\sqrt{(j-i+1)}}\right) + A_j \left(\frac{\zeta}{1}\right). \tag{4.29}$$

Now we can finally solve for the unknown A_j and calculate every B_{j+1} iteratively according to Eq. 4.30:

$$A_j = \frac{B_{j+1} - \sum_{i=1}^{j-1} A_i \mathrm{erfc}\left(\frac{\zeta}{\sqrt{(j-i+1)}}\right)}{\mathrm{erfc}(\zeta)}. \tag{4.30}$$

This yields the wall temperature $T_{W,j}$for every iteration j according to Eq. 4.31

$$T_{W,j} = T_0 + \Delta t \sum_{i=1}^{j-1} A_i. \tag{4.31}$$

The specific heat flux $\dot{q}$ can be calculated as described in Eq. 4.22 and 4.23.

4.2.3 Differential Equation

The next method follows a similar approach as the heat conduction equation with the distinction of a different model to calculate the heat flux. The idea is to formulate a differential equation with constant terms and factors for the wall temperature. The first model used is depicted in Fig. 4.4 and uses Eq. 4.32

$$\frac{Q}{c_p\rho} = \int_{z=0}^{\infty} T_W - T dz \approx T_W L - \frac{\Delta z_1}{2}(T_W + T_1) - \frac{\Delta z}{2}(T_1 + T_2) - \frac{\Delta z}{2}(T_2 + T_3) - ... - \frac{\Delta z}{2}(T_6 + T_7). \quad (4.32)$$

This equation can be rearranged and leads to

$$\frac{Q}{c_p\rho} \approx -\frac{\Delta z_1}{2}T_W + T_0 L - \frac{\Delta z_1}{2}T_1 - \Delta z(\frac{T_1}{2} + T_2 + ... + T_6 + \frac{T_7}{2}) =: \frac{A}{c_p\rho}T_W + \frac{B}{c_p\rho} \quad (4.33)$$

with the introduction of the two factors A and B

$$A = \frac{\rho c_p \Delta z_1}{2} \quad (4.34)$$

$$B = -\rho c_p T_0 L + \rho c_p \left(\frac{T_1 \Delta z_1}{2} + \Delta z \left(\frac{T_1}{2} + T_2 + ... + T_6 + \frac{T_7}{2} \right) \right) \quad (4.35)$$

which shall be used later. Furthermore for the calculation of the wall temperature we use Fourier's law of heat conduction

$$\vec{q} = \lambda \nabla T. \quad (4.36)$$

which can be simplified to Eq. 4.37

$$\dot{q} = \lambda \frac{dT}{dz} \quad (4.37)$$

for the 1-dimensional case. Now the spatial temperature gradient can be approximated using Taylor polynomials for the first and second thermoelement since the wall temperature is unknown

$$T_1 = T_W - T' \Delta z_1 + \frac{T''}{2} {\Delta z_1}^2 \quad (4.38)$$

$$T_2 = T_W - T' \Delta z_2 + \frac{T''}{2} {\Delta z_2}^2 \quad (4.39)$$

Now we eliminate T'' by multiplying Eq. 4.38 with ${\Delta z_2}^2$, Eq. 4.39 with ${\Delta z_1}^2$ and subtracting Eq. 4.39 from Eq. 4.38. This leads to Eq. 4.40

$$T_1 {\Delta z_2}^2 - T_2 {\Delta z_1}^2 = T_W({\Delta z_2}^2 - {\Delta z_1}^2) + T'(\Delta z_1 {\Delta z_2}^2 - \Delta z_2 {\Delta z_1}^2) \quad (4.40)$$

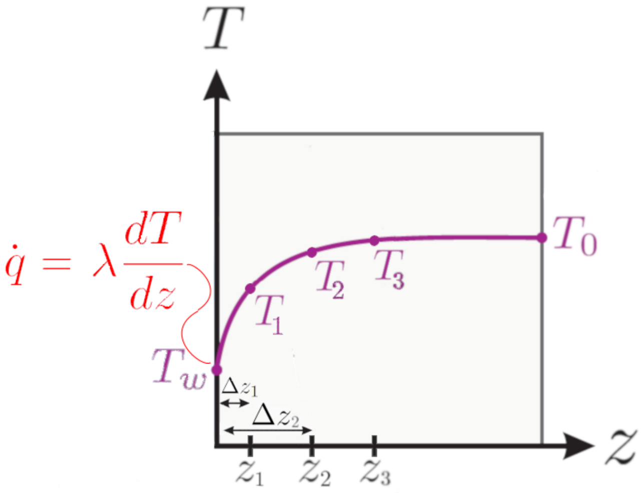

Figure 4.4: Heat flux using thermoelements 1 and 2

which can be solved for T' in order to get the spatial temperature gradient needed

$$T' = \frac{dT}{dz} = \frac{T_1 {\Delta z_2}^2 - T_2 {\Delta z_1}^2 - T_W({\Delta z_2}^2 - {\Delta z_1}^2)}{\Delta z_1 {\Delta z_2}^2 - \Delta z_2} {\Delta z_1}^2. \tag{4.41}$$

Eq. 4.41 yields the second equation needed to formulate a differential equation by differentiating Eq. 4.33. First we introduce two factors C and D as

$$\frac{dT}{dz} = \frac{C}{\lambda} T_W + \frac{D}{\lambda} \tag{4.42}$$

with

$$C = -\lambda \frac{\Delta z_1 + \Delta z_2}{\Delta z_1 {\Delta z_2}^2 - \Delta z_2 {\Delta z_1}^2} \tag{4.43}$$

and

$$D = \frac{T_1 {\Delta z_2}^2 - T_2 {\Delta z_1}^2}{\Delta z_1 {\Delta z_2}^2 - \Delta z_2 {\Delta z_1}^2}. \tag{4.44}$$

The temporal differentiation of Q from Eq. 4.33 is equal to $\dot{q}$ from Eq. 4.37 plus the heat flux lost on the underside of the probe. The sides of the probe are well insulate and thus assumed adiabatic. This is stated in Eq. 4.45

$$\dot{q} = \frac{dQ}{dt} + \dot{q}_L. \tag{4.45}$$

The heat flux lost on the underside of the probe is relatively small. Therefore the temperature gradient can be simply approximated by a finite difference

$$\dot{q}_L \approx \lambda \frac{T_7 - T_6}{\Delta z}. \tag{4.46}$$

Inserting Eq. 4.33, 4.42 and 4.46 in Eq. 4.45 yields the following differential equation for T_W with known variables A, B, C, D and $\dot{q_L}$

$$A\dot{T_W} = CT_W - \dot{B} + D - \dot{q}_L. \tag{4.47}$$

This inhomogenous differential equation of the first order has an analytical solution which is

$$T_W(t) = e^{\frac{Ct}{A}} \left(T_0 + \int_0^t \frac{e^{\frac{-Cx}{A}} (D(x) - \dot{q}_L - \dot{B}(x))}{A} dx \right). \tag{4.48}$$

This wall temperature can then be used to calculate the heatflux, for example by using Eq. 4.37 or the method described in 4.22.

4.2.4 Finite Difference Method

The finite difference method solves the heat conduction equation for a 1-dimensional case with temperature measurements along one axis (e.g. the z-axis) or a 2-dimensional grid with temperature measurements along a z-axis as well as a r-axis. Firstly, the 1-dimensional case shall be described and then the extension to two dimensions is made.

4.2.5 1-dimensional Finite Difference Method

The 1-dimensional finite difference method starts by assuming a cubic spline for the temperature function of the form

$$T(z) = T_1 + A(z - z_1) + B(z - z_1)^2 + C(z - z_1)^3. \tag{4.49}$$

Now Eq. 4.49 is evaluated at the points $z_2 = z_1 + \Delta z$ and $z_3 = z_1 + 2\Delta z$ so that

$$T_2 = T_1 + A\Delta z + B\Delta z^2 + C\Delta z^3 \tag{4.50}$$

$$T_2 = T_1 + 2A\Delta z + B(2\Delta z)^2 + C(2\Delta z)^3. \tag{4.51}$$

Eq. 4.50 and 4.51 can be used to eliminate the factors A and B

$$T(z) = T_1 - \frac{(-4C\Delta z^3 + 3T_1 - 4T_2 + T_3)(z - z_1)}{2\Delta z} - \frac{(6C\Delta z^3 - T_1 + 2T_2 - T_3)(z - z_1)^2}{2\Delta z^2} + C(z - z_1)^3 \tag{4.52}$$

so only the factor C remains. We can use Eq. 2.13 evaluated at $z = z_1$ to solve for C. This yields

$$C = -\frac{c_p \frac{dT}{dt} \rho}{6\Delta z \lambda} + \frac{T_1 - 2T_2 + T_3}{6\Delta z^3}. \tag{4.53}$$

Now Eq. 4.52 can be evaluated for $z \to 0$

$$T_W = T_1 + \frac{(-4C\Delta z^3 + 3T_1 - 4T_2 + T_3)z_1}{2\Delta z} + \frac{(6C\Delta z^3 - T_1 + 2T_2 - T_3)z_1^2}{2\Delta z^2} - Cz_1^3 \tag{4.54}$$

which also yields the heat flux $\dot{q}$

$$\dot{q} = \lambda \frac{dT}{dz}\bigg|_{z\to 0} = \lambda\left(\frac{-4C\Delta z^3 + 3T_1 - 4T_2 + T_3}{2\Delta z} + \frac{(6C\Delta z^3 - T_1 + 2T_2 - T_3)z_1}{\Delta z^2} - 3Cz_1^2\right). \tag{4.55}$$

4.2.6 2-dimensional Finite Difference Method

The 2-dimensional method also uses a cubic spline to approximate the temperature distribution along the z-axis. Analogously, Eq. 4.49 is simplified to Eq. 4.52. The heat conduction equation can be discretized using the notation depicted in Fig. 4.5. Therefore we have the discrete heat conduction equation 4.56

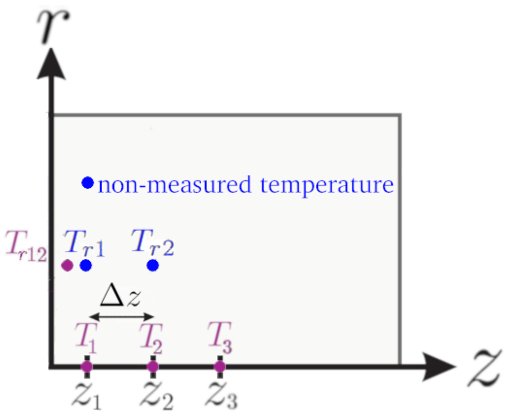

Figure 4.5: Temperature measurements and approximated temperature

$$\rho c_p \frac{dT}{dt} - \lambda \frac{T_1 - 2T_2 + T_3}{\Delta z^2} - \lambda \frac{4(T_{r2} - T_2)}{\Delta r^2} = 0. \tag{4.56}$$

As depicted in Fig. 4.5 for our grid we need the non-measured temperature T_{r2}. We can solve Eq. 4.56 for T_{r2}

$$T_{r2} = \frac{c_p \Delta r^2 \frac{dT_2}{dt} \Delta z^2 \rho - \Delta r^2 \lambda T_1 + 2\Delta r^2 \lambda T_2 + 4\Delta z^2 \lambda T_2 - -\Delta r^2 \lambda T_3}{4\Delta z^2 \lambda}. \tag{4.57}$$

Now we interpolate the temperature T_{r1} using our measured temperature T_{r12} as well as the calculated T_{r2}

$$T_{r1} = T_{r12} + \frac{T_{r2} - T_{r12}}{z_1/2 + \Delta z} z_1/2. \tag{4.58}$$

To determine C from Eq. 4.52 we use a temperature distribution of the radial coordinate r

$$T(r) = T_1 + Dr^2. \tag{4.59}$$

The Laplace operator ΔT of the temperature is for cylindrical coordinates with point symmetry defined as

$$\Delta T = \frac{d}{rdr}\left(r\frac{dT}{dr}\right) + \frac{d^2T}{dz^2}. \tag{4.60}$$

Therefore in our case this yields

$$\Delta T = \frac{6C\Delta z^3 - T_1 + 2T_2 - T_3}{\Delta z^2} + \frac{4(-T_1 + T_{r1})}{\Delta r^2}. \tag{4.61}$$

Now we can use the general heat conduction equation again to calculate C using T_{r1} from Eq. 4.58

$$C = \frac{-c_p \Delta r^2 \frac{T_1}{dt} \Delta z^2 \rho + \lambda(\Delta r^2 (T_1 - 2T_2 + T_3) + 4\Delta z^2(-T_1 + T_{r1}))}{6\Delta r^2 \Delta z^3 \lambda}. \tag{4.62}$$

Using this updated C, the same procedure as in the 1-dimensional finite difference method can be deployed, namely making use of Eq. 4.54 and 4.55 to calculate the wall temperature T_W and heat flux $\dot{q}$ respectively.

5 Results

This chapter shows the calculation results using the methods delineated in chapter 4. The test case yields the wall temperature as well as the heat flux. This can be compared to the calculation results of the analytical solution using the Laplace transform (here referred to as Monde) as well as the newly invented inverse heat flux calculation methods. The abbreviations for the Heat Conduction Equation Method and Finite Difference Method are HCE and FD1D respectively. The cosine test case was used to continuously increase the complexity of the heat flux calculation algorithms. All features of the cosine test case can also be observed in the final test case, therefore the cosine case shall not be discussed in the following. The methods Gaussian Error Function and Differential Equation proved not to be applicable to our measurement data, because they needed an unproportionally high amount of pretreatment. Given the measurement oscillations in the temperature measurements as depicted in the calculation results in Fig. 5.7 this data proved to be too strongly fluctuating for the iterative processes of the Gaussian Error Function and Differential Equation methods. The test case was designed to be 1-dimensional and did not provide radial temperature data. Therefore FD2D was not used for the test case and consequently not used for the measurement data since it could not be validated. The source code of the implemented calculation methods from Chapter 4 is attached in the appendix.

5.1 Wall Temperature and Heat Flux in the Test Case

5.1.1 Linear Test Case

The simplest form of the test case is linear temperature decrease. All in Fig. 5.1 depicted calculation models approximate the unknown wall temperature T_W on a global level very well. Over the course of the whole simulation the Monde solution, the Heat Conduction Equation and the Finite Difference Method in the 1-dimensional form agree with the simulation very well.

Upon closer inspection, a small difference between test case and the Heat Conduction Equation can be observed. This is depicted in Fig. 5.2 and can be attributed to the fact, that the Heat Conduction Equation does not take the heat flux on the bottom of the probe into account. The detailed depiction of the temperature is continued in the form that is presented in 5.2. The temperature decrease is continued and matched by the calculation methods Finite Differences and Monde until the end of the simulation.

Using the wall temperature T_W the heat flux $\dot{q}$ on the surface of the probe can be calculated. The graphs for the same calculation methods as before are depicted in Fig. 5.3. Similarly to the temperature trend the heat flux is calculated very accurately. The temperature gradient is proportional to the heat flux therefore the graph is not constant but increases at a decreasing rate resembling a root-function. The rising temperature gradient is not apparent in the temperature graph, but can be logically explained. The cooled surface has a higher temperature difference to the inside of the probe, therefore the heat flux rises at the beginning.

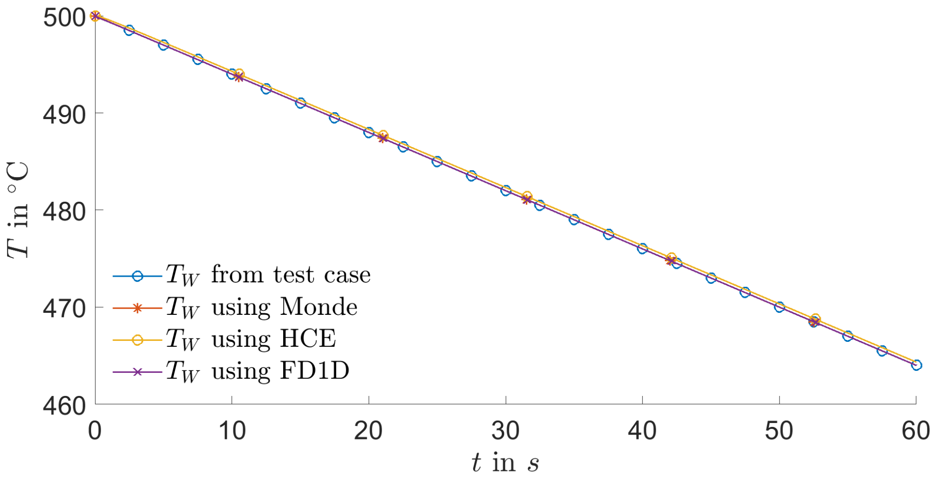

Figure 5.1: Calculated temperature in the linear test case

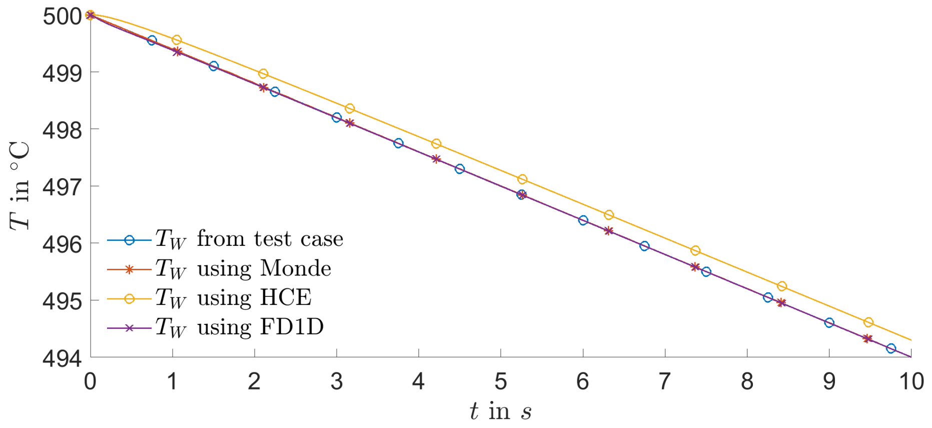

Figure 5.2: Calculated temperature in the linear test case

After a delay the cooling effect also affects most of the probe body and the heat flux rise slows down. The heat flux has a negative sign since it is emmitted heat by the probe. The absorbed heat by the droplet would be positive.

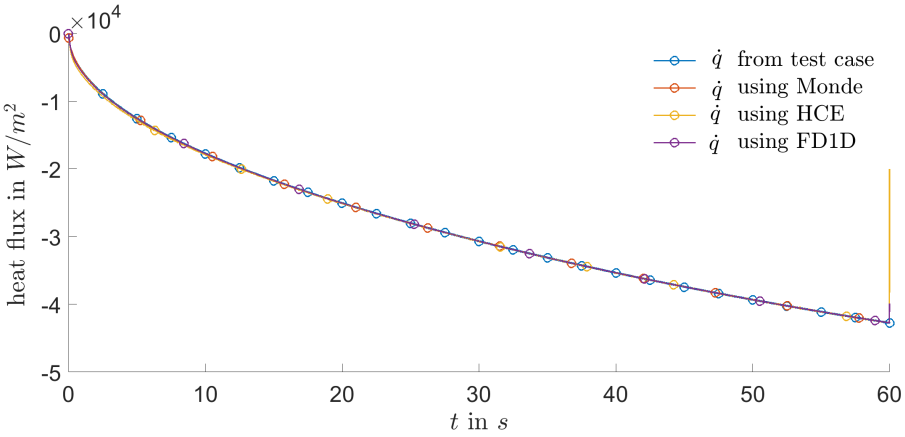

Figure 5.3: Calculated heat flux in the linear test case

5.1.2 Final Test Case

All features of the cosine test case can be observed in the final test case which combines a linear temperature decrease which resembles a cooling effect with a sinusoidal temperature course. The sinusoidal course approximates a temperature measurement of drop impact. The drop lands on the heated body, cools it down and evaporates. Heating from the underlying probe body increases the temperature again. This characteristic temperature trend can be seen in the measurement data (see Fig. 5.7). The beginning of the simulation features a response delay of the thermoelement measurement. Often singularities are observed when the drop first touches the surface and a contact temperature results. This singularity is dampened, because the temperature changes faster than the thermoelement can record data. The initial discontinuity is described by a step function (see Fig. 5.4). The temperature trend continues in the same form until the end of the simulation. An exemplary discussion of the first $10\ s$ covers all observed phenomena.

In contrast to the linear test case the test case data does only overlap with the Monde solution. All models differ in amplitude and the Heat Conduction Equation method even in phase shift. The phase shift is explicable due to the time the temperature gradient needs to travel through the material. This information delay is used to calculate the total heat. The model does not contain a function which reverses the phase shift. This could be implemented, but would lead to more data preprocessing.

The Monde solution does exactly match the test case temperature in amplitude and phase shift. A higher calculation time correlates with higher accuracy in comparison to the other calculation methods.

The Finite Difference Method in its 1-dimensional version has no phase shift. The temperature oscillation is overestimated by approximately 10 percent.

The mean of all calculations fits the mean of the real temperature. The Monde solution calculated the temperature trend most exactly, the Finite Difference Method also calculated mean and phase accurately, though overestimated the temperature amplitude by a small amount. The Heat Conduction Equation is a rough estimate for amplitude and phase.

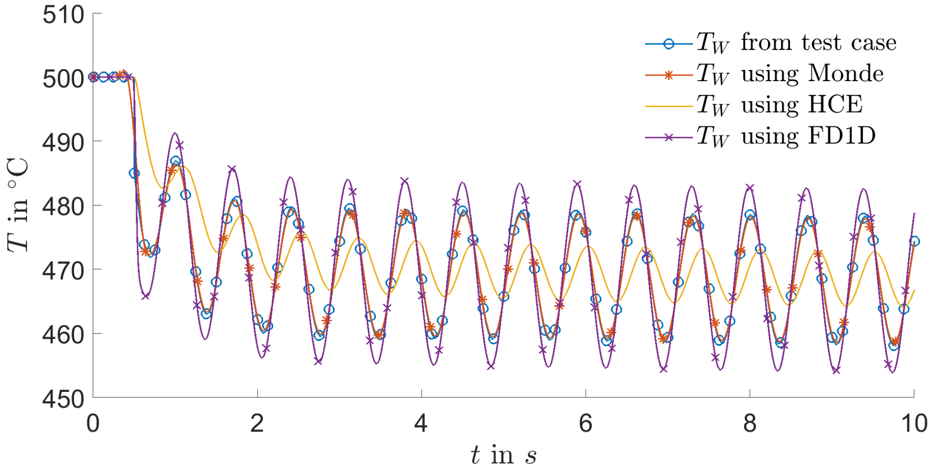

Figure 5.4: Calculated temperature in the final test case

The first $10\ s$ also suffice to describe the characteristics of the heat flux curve. The following $50\ s$ which are not depicted in Fig. 5.5 show the exact same trend until the end of the simulation.

Surprisingly the Monde solution which calculated the temperature best underestimated the heat flux by a big factor. The real heat flux from the test case is roughly 4 times higher in amplitude. The phase shift is smallest for the Monde and the Finite Difference solution. The Heat Conduction Equation also calculates the phase relatively well. All three calculation methods determine the mean of the heat flux exactly. The phenomenon that Monde differs from other calculation methods for the heat flux can also be observed in the measurement data. The Monde solution seems to calculate a broader mean. Changes in heat flux are therefore not represented by the same amplitude.

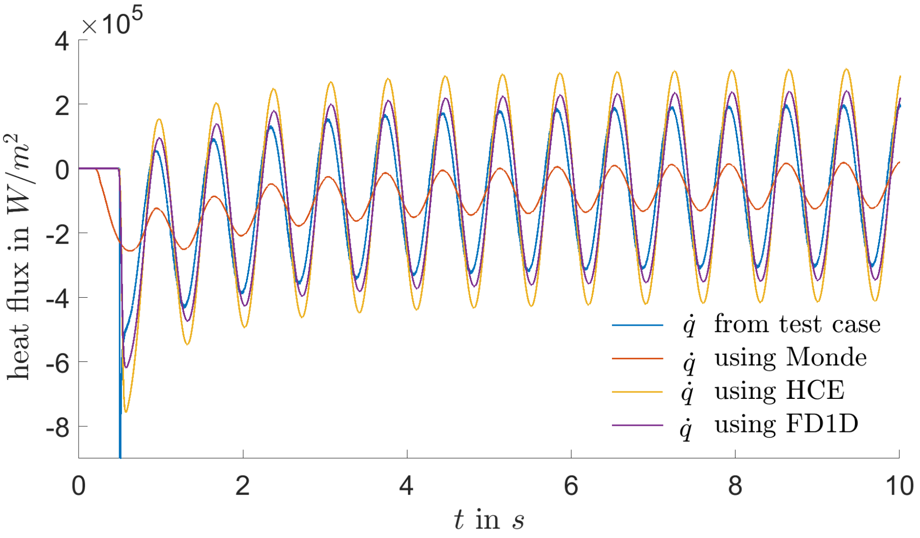

Figure 5.5: Calculated heat flux in the final test case

5.2 Wall Temperature and Heat Flux in Measurement Data

The final temperature trend for the real measurement data is depicted in Fig. 5.6. The measurement took place over the course of $2262\ s$ with a sample rate of $95\ Hz$. The algorithm provided by Monde et al. only supports 16 000 data points. Therefore the measurement data needed to be sliced in 13 data sets for individual calculation. These assembled results show an initial discontinuity at the beginning of each data set.

Allover the three calculation methods used agree on the wall temperature very well. The fluctuation in measurement is more sensible for the Finite Difference method and Monde than the Heat Conduction Equation. The global decrease is depicted in Fig. 5.6. Individual drop impacts cannot be distinguished in the global trend. The temperature drops in the domain from $T = 460\ °C$ to $T = 250\ °C$ with a constant slope. As depicted in Fig. 5.7 individual drop impacts can be made out. Consecutive drop impacts do not seem to interfere with each other in the high temperature domain $T \in [250, 460]\ °C$. Interactions between consecutive drops can occur in a lower temperature domain $T \in [110, 250]$. This was also affirmed by image data [4].

Fig. 5.7 depicts an arbitrary interval of roughly $25\ s$ in length. Other intervals resemble the same characteristics qualitatively. Individual decreases in temperature can be made out due to the drop impact and reheating from the hot body underneath. The drop impact causes a rapid local temperature decline. This temperature decline stops as soon as the drop does not touch the probe surface anymore. The surrounding and underlying probe body then slowly reheats the drop impact area until the next drop impact.

The agreement between all three models can be seen in Fig. 5.7. The temperature increase after drop impact seems to be more rapid than in the Heat Conduction Equation graph, because the Monde solution and the mean of the Finite Difference Method agree and validate each other. The temperature fluctuations seem to increase in amplitude around $T = 250\ °C$. This can be attributed to wetting. For these lower temperatures the time between each individual drop impact does not suffice for the complete evaporation of the previous droplet. The wetted surface leads to a higher temperature difference between drop impact and (partial) drop evaporation. The higher temperature difference also causes a higher temperature gradient and therefore higher heat flux. This can be seen in Fig 5.8.

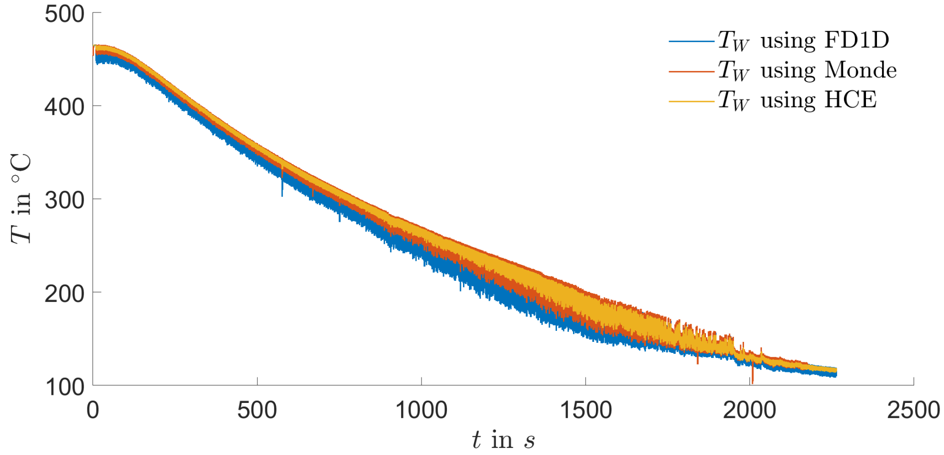

Figure 5.6: Calculated temperature in measurement data

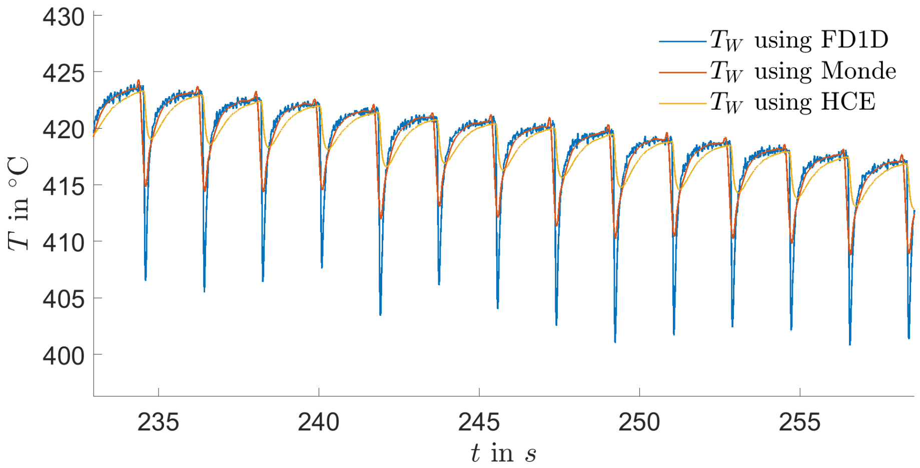

Figure 5.7: Calculated temperature in measurement data

Fig. 5.8 shows the global trend of the heat flux. For the whole time of the simulation no individual drop impacts can be recognized. Similarly to the temperature trend a section of the heat flux trend demonstrates the characteristic differences of the different calculation models used. Fig. 5.9 shows individual drop impact and allows distinguishable heat flux graphs. In contrast to the temperature graph the Heat Conduction Equation shows the highest temperature fluctuations. The Monde solution depicts an averaged heat flux which does not oscillate in the same high amplitude as the Finite Difference method or the Heat Conduction Equation. Due to the good agreement in temperature trends between the Finite Difference method and Monde, the heat fluxes of those two models are most likely accurate as well. It is to be expected that the true heat flux is slightly higher than the Monde calculation as the final test case demonstrated. The Heat Conduction Equation overestimates the heat flux due to the fact, that firstly, the heat lost on the bottom of the probe is neglected and secondly, the temperature rises more slowly (see Fig. 5.7). The 2-dimensional Finite Difference Method provides a continuous solution that agrees with the 1-dimensional version very well.

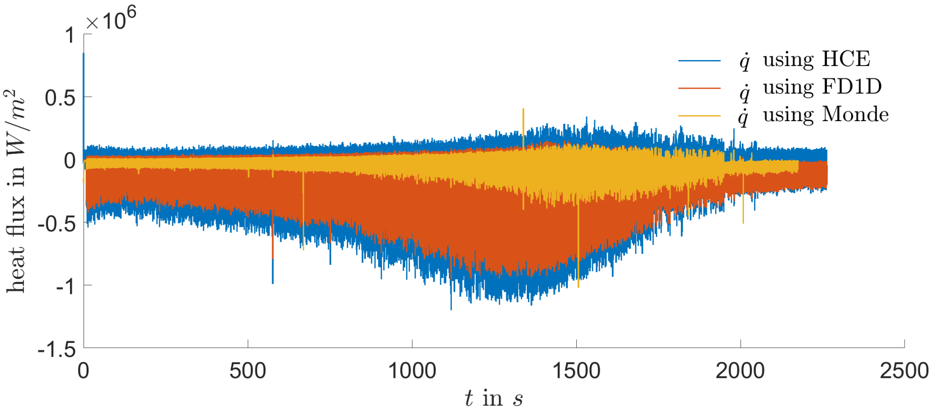

Figure 5.8: Calculated heat flux in measurement data

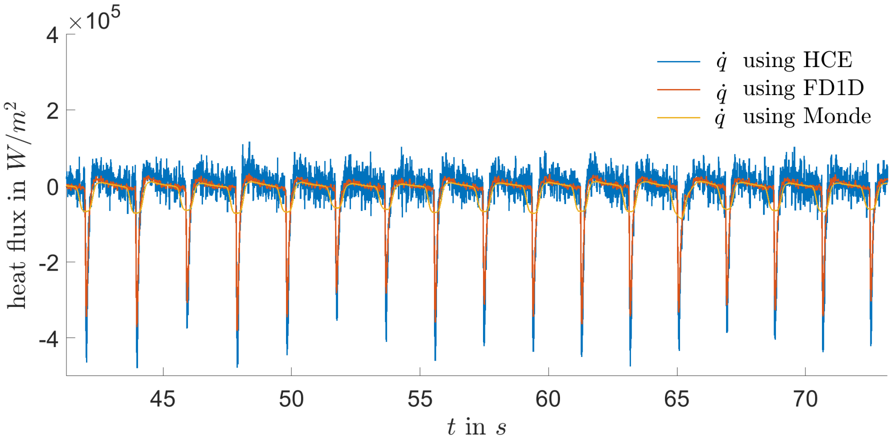

Figure 5.9: Calculated heat flux in measurement data

6 Conclusion and Outlook

Different methods for the inverse calculating of the heat flux during the impact of a drop chain on a heated surface were implemented. These methods consist on the one hand of implementing the already existing calculation method presented by Monde et al. for the present test cases and measurement data. On the other hand new calculation methods, namely the Heat Conduction Equation method, the Gaussian Error Function method, the Differential Equation method and the Finite Difference method in its 1- and 2-dimensional form, were implemented. The measurement data as well as the test case, which was modeled after the measurement data, reduced the applicable calculation methods to the Heat Conduction Equation method and the Finite Difference Method.

A test case in its most complex form showed a good agreement between the simulated surface temperature, the solution obtained by the Monde code and the Finite Difference Method for any time. The Heat Conduction Equation calculated the mean of the temperature decrease over the time of the whole simulation accurately. The heat flux was calculated precisely by the Heat Conduction Equation and the Finite Difference method. The Monde solution provided an accurate solution for the mean of the heat flux but could not depict the spikes in the heat flux.

The measurement data was evaluated using the Monde code, the Heat Conduction Equation method and the Finite Difference method. The solution for the surface temperature showed good agreement between Monde and the Finite Difference method. The Heat Conduction Equation managed to calculate the global surface temperature trend. The heat flux was calculated and all used methods agreed over their global trend. The Heat Conduction Equation method's solution oscillated to such a high degree, that post processing will be necessary, if this method should be employed.

Finally, the Finite Difference method agreed with the proven Monde code. In a special case, it even provided a more accurate depiction of the heat flux. The computing time of this new method is substantially lower than the Monde code and it is not limited to 16 000 data points.

Future work could include extensive data pretreatment to also implement the Gaussian Error Function method and the Differential Equation method. Furthermore the heat flux results of the Heat Conduction Equation method can be post processed to reduce oscillation. An agreement between these improved methods and the Finite Difference method further validates the results.

Bibliography

[1] Yarin, A.L.: Drop impact dynamics: Splashing, spreading, receding, bouncing. . . . In: *Annual Review of Fluid Mechanics* 38 (2006), Nr. 1, S. 159–192

[2] Breitenbach, J. ; Roisman, I. V. ; Tropea, C.: From drop impact physics to spray cooling models: a critical review. In: *Experiments in Fluids* 59 (2018). https://link.springer.com/article/10.1007/s00348-018-2514-3

[3] Ciofalo, M. ; Caronia, A. ; Di Liberto, M. ; Puleo, S.: The Nukiyama curve in water spray cooling: Its derivation from temperature–time histories and its dependence on the quantities that characterize drop impact. In: *International Journal of Heat and Mass Transfer* 50 (2007), Nr. 25, 4948 - 4966. http://www.sciencedirect.com/science/article/pii/S001793100700600X. – ISSN 0017–9310

[4] Quell, J. K. O.: Experimentelle Untersuchung der Leidenfrost Temperatur beim Tropfenkettenaufprall auf eine beheizte Oberfläche. (2020), Bachelor-Thesis, Darmstadt: Technische Universität Darmstadt

[5] Nayak, S. K. ; Mishra, P. C. ; Parashar, S. K. S.: Influence of spray characteristics on heat flux in dual phase spray impingement cooling of hot surface. In: *Alexandria Engineering Journal* 55 (2016), Nr. 3, S. 1995–2004

[6] Liang, G. ; Mudawar, I.: Review of spray cooling – Part 1: Single-phase and nucleate boiling regimes, and critical heat flux. In: *International Journal of Heat and Mass Transfer* 115 (2017), 1174 - 1205. http://www.sciencedirect.com/science/article/pii/S0017931017302946. – ISSN 0017–9310

[7] Tenzer, F. M. ; Roisman, I. V. ; Tropea, C.: Fast transient spray cooling of a hot thick target. In: *Journal of Fluid Mechanics* 881 (2019), S. 84–103

[8] Rioboo, R. ; Tropea, C. ; Marengo, M.: Outcomes from a drop impact on solid surfaces. In: *Atomization and Sprays* 11 (2001), Nr. 2. – ISSN 1044–5110

[9] Gottfried, B. S. ; Bell, K. J.: Film boiling of spheroidal droplets. Leidenfrost phenomenon. In: *Industrial Engineering Chemistry Fundamentals* 5 (1966), November, Nr. 4, 561-568. https://doi.org/10.1021/i160020a023. – ISSN 0196–4313, 1541–4833 (electronic)

[10] Roisman, I. V. ; Breitenbach, J. ; Tropea, C.: Thermal atomisation of a liquid drop after impact onto a hot substrate. In: *Journal of Fluid Mechanics* 842 (2018), S. 87–101

[11] Roisman, I. V.: Fast forced liquid film spreading on a substrate: Flow, heat transfer and phase transition. In: *Journal of Fluid Mechanics* 656 (2010), S. 189–204

[12] Hall, D. D. ; Mudawar, I.: Experimental and numerical study of quenching complex-shaped metallic alloys with multiple, overlapping sprays. In: *International Journal of Heat and Mass Transfer* 38 (1995), Nr. 7, 1201 - 1216. `http://www.sciencedirect.com/science/article/pii/001793109400244P`. – ISSN 0017–9310

[13] Bertola, V.: An impact regime map for water drops impacting on heated surfaces. In: *International Journal of Heat and Mass Transfer* 85 (2015), 430 - 437. `http://www.sciencedirect.com/science/article/pii/S0017931015000927`. – ISSN 0017–9310

[14] Alifanov, O. M.: *Inverse heat transfer problems*. Springer-Verlag, Berlin, 1994

[15] Daniell, P. J.: Lectures on Cauchy's problem in linear partial differential equations by J. Hadamard. In: *The Mathematical Gazette* 12 (1924), Nr. 171, S. 173–174

[16] Bertero, M. ; Poggio, T. A. ; Torre, V.: Ill-posed problems in early vision. In: *Proceedings of the IEEE* 76 (1988), Nr. 8, S. 869–889

[17] Woodfield, P.L. ; Monde, M. ; Mitsutake, Y.: Improved analytical solution for inverse heat conduction problems on thermally thick and semi-infinite solids. In: *International Journal of Heat and Mass Transfer* 49 (2006), Nr. 17, 2864 - 2876. `http://www.sciencedirect.com/science/article/pii/S0017931006001530`. – ISSN 0017–9310

[18] Alghamdi, A. S. A.: Inverse estimation of boundary heat flux for heat conduction model. (2010), Technical Report, Makkah: Umm Al-Qura University

[19] Tao, M. ; Xing, C. ; Yao, S. ; Qiang, H.: A numerical approach to solving an inverse heat conduction problem using the Levenberg-Marquardt algorithm. (2014), Technical Report, Xi'an: Xi'an University of Technology

Appendix

```
clear;
close all;
clc;

%% define variables

% material value stainless steel 1.4841

rho     = 7900;                 % kg/m³            density
cp      = 542;                  % K/(kg K)         specific heat at constant pressure
lambda  = 16;                   % W/(m K)          thermal conductivity
e_w     = sqrt(rho*cp*lambda);  % J/(m² K sqrt(s)) thermal effusivity
alpha   = lambda/(rho*cp);      % m/s              thermal diffusivity
temp.samplerate = 95;

%% load Data

load('20200911_000.mat')

%% Heat Conduction Equation method calculation


% determine geometric parameters

delta_z1 = -temp.Position_z1(1,1);      % m     distance from first thermoelement to↙
surface
delta_z2 = -temp.Position_z2(1,1);      % m     distance from second thermoelement to↙
surface
delta_z = delta_z2 - delta_z1;          % m     distance between thermoelements
h = 0.025;                              % m     height of probe body
delta_z6 = h + temp.Position_z6(1,1);   % m     distance from sixth thermoelement to↙
bottom
delta_z7 = h + temp.Position_z7(1,1);   % m     distance from seventh thermoelement to↙
bottom

% for test case

% T_z1 = temp.T_z1;                   % °C    temperature thermoelement z1
% T_z2 = temp.T_z2;                   % °C    temperature thermoelement z2
% T_z3 = temp.T_z3;                   % °C    temperature thermoelement z3
% T_z4 = temp.T_z4;                   % °C    temperature thermoelement z4
% T_z5 = temp.T_z5;                   % °C    temperature thermoelement z5
% T_z6 = temp.T_z6;                   % °C    temperature thermoelement z6
% T_z7 = temp.T_z7;                   % °C    temperature thermoelement z7

% for measurement data

T_z1 = temp.T_z1';                % °C    temperature thermoelement z1
T_z2 = temp.T_z2';                % °C    temperature thermoelement z2
T_z3 = temp.T_z3';                % °C    temperature thermoelement z3
T_z4 = temp.T_z4';                % °C    temperature thermoelement z4
T_z5 = temp.T_z5';                % °C    temperature thermoelement z5
T_z6 = temp.T_z6';                % °C    temperature thermoelement z6
T_z7 = temp.T_z7';                % °C    temperature thermoelement z7
```

```
dT_z1Dt = gradient(temp.T_z1,1/samplerate);        % temporal gradient thermoelement z1
dT_z7Dt = gradient(temp.T_z7,1/samplerate);        % temporal gradient thermoelement z7

t = temp.t;

% read temperature in struct with average over "meanWidth"

meanWith = 3;
T_z1 = movmean(temp.T_z1,meanWith);                   % °C    temperature thermoelement↙
z1
T_z2 = movmean(temp.T_z2,meanWith);                   % °C    temperature thermoelement↙
z2
T_z3 = movmean(temp.T_z3,meanWith);                   % °C    temperature thermoelement↙
z3
T_z4 = movmean(temp.T_z4,meanWith);                   % °C    temperature thermoelement↙
z4
T_z5 = movmean(temp.T_z5,meanWith);                   % °C    temperature thermoelement↙
z5
T_z6 = movmean(temp.T_z6,meanWith);                   % °C    temperature thermoelement↙
z6
T_z7 = movmean(temp.T_z7,meanWith);                   % °C    temperature thermoelement↙
z7
T_r1 = movmean(temp.T_r1,meanWith);                   % °C    temperature thermoelement↙
r1
T_r2 = movmean(temp.T_r2,meanWith);                   % °C    temperature thermoelement↙
r2
T_r3 = movmean(temp.T_r3,meanWith);                   % °C    temperature thermoelement↙
r3
T_r4 = movmean(temp.T_r4,meanWith);                   % °C    temperature thermoelement↙
r4

% preallocate memory

q = zeros(1,length(temp.t));
Q = zeros(1,length(temp.t));
T_z0 = zeros(1,length(temp.t));
T_z8 = zeros(1,length(temp.t));

% set first timestep

T_z0(1) = T_z1(1);                   % °C    temperature surface (z=0mm)
T_z8(1) = T_z7(1);                   % °C    temperature bottom  (z=25mm)

Q(1) = rho*cp*(+ ...
       delta_z1*(T_z0(1)+T_z1(1))/2 +...      % Q in upper interval between surface and↙
first thermoelement
       delta_z/2*(T_z1(1) + 2*T_z2(1) + 2*T_z3(1) + 2*T_z4(1) + 2*T_z5(1) + 2*T_z6(1) +↙
T_z7(1)) + ...   % delta Q in all thermoelements
       delta_z7*(T_z7(1)+T_z8(1))/2);         % Q in lower interval between bottom and↙
seventh thermoelement

for it=2:1:length(temp.t)

    % calculate wall temperature surface (z=0mm)
    T_z0(it) = T_z1(it)*(1 + delta_z1/delta_z) - T_z2(it)*delta_z1/delta_z + dT_z1Dt(it)↙
*(delta_z1*delta_z2)/(2*alpha);

    % calculate wall temperature bottom  (z=25mm)

    T_z8(it) = T_z7(it)*(1 + delta_z7/delta_z) - T_z6(it)*delta_z7/delta_z + dT_z7Dt(it)↙
*(delta_z7*delta_z6)/(2*alpha);

    % calculate heat in material

    Q(it) = rho*cp*(+ ...
           delta_z1*(T_z0(it)+T_z1(it))/2 +...       % Q in upper interval between↙
surface and first thermoelement
           delta_z/2*(T_z1(it) + 2*T_z2(it) + 2*T_z3(it) + 2*T_z4(it) + 2*T_z5(it) +↙
2*T_z6(it) + T_z7(it)) + ...   % delta Q in all thermoelements
           delta_z7*(T_z7(it)+T_z8(it))/2);           % Q in lower interval between↙
bottom and seventh thermoelement

    % calculate heat flux

    q(it) = (Q(it) - Q(it-1)) / (temp.t(it) - temp.t(it-1));

end

% save data to struct

heatHCE.measurement_number = temp.measurement_number;
heatHCE.solver = 'Heatflux Ilias Heat Conduction Equation';
heatHCE.meanWith = meanWith;
heatHCE.t = temp.t';
heatHCE.q = q;
heatHCE.T_z0 = T_z0;

%% Gaussian Error Function method

% set variables for the calculation

delta_t = 0.6;                                       % s     time difference between↙
temperature measurements
                                                     % > 1   for realistic↙
temperature results
tpoints = (temp.t(1,end)-temp.t(1,1))/delta_t;       % -     number of points, for↙
which the temperature is calculated
tstar = linspace(temp.t(1,1),temp.t(1,end),tpoints);  % s     timevector
delta_z1 = -temp.Position_z1(1,1);                   % m     distance from first↙
thermoelement to surface

zeta = delta_z1 / (2 * sqrt(alpha * delta_t));

% smoothing data for smalter delta_t

fT_z1 = fit(temp.t',temp.T_z1','smoothingspline','SmoothingParam',0.7);
T_z1star = fT_z1(tstar);

B(:) = (T_z1star(:)-T_z1star(1))/delta_t;
```

```
% loop for calculation

A = zeros(length(tstar),1);         % preallocate memory
Asum = zeros(length(tstar),1);      % preallocate memory
T_z0star = zeros(length(tstar),1);  % preallocate memory


% define variable in first time step

A(1) = 0;
Asum(1) = 0;
T_z0star(1) = T_z1star(1);


for j=2:1:(length(tstar)-1)

    Asum(j) = Asum(j-1)+A(j-1);

    A(j) = (B(j+1)-Asum(j))/(erfc(zeta));

    % surface temperature for time step j

    T_z0star(j) = T_z1star(1) + delta_t*Asum(j);

end

T_z0star(end) = T_z0star(end-1);
Tstar = T_z0star;

% calculates heat flux for T_z0star

x = [1 4.5 8 11.5 15 18.5 22] / 1000 ;  %depth of measurement points H1 to H7 in Meter
delta_z1 = -temp.Position_z1(1,1);      % m     distance from first thermoelement to↙
surface
delta_z2 = -temp.Position_z2(1,1);      % m     distance from second thermoelement to↙
surface
delta_z = delta_z2 - delta_z1;          % m     distance between thermoelements

T_z1 = temp.T_z1';                % °C    temperature thermoelement z1
T_z2 = temp.T_z2';                % °C    temperature thermoelement z2
T_z3 = temp.T_z3';                % °C    temperature thermoelement z3
T_z4 = temp.T_z4';                % °C    temperature thermoelement z4
T_z5 = temp.T_z5';                % °C    temperature thermoelement z5
T_z6 = temp.T_z6';                % °C    temperature thermoelement z6
T_z7 = temp.T_z7';                % °C    temperature thermoelement z7

% first time step

T_z0(1) = T_z1(1);
T_z02(1) = T_z1(1);
T_z8(1) = T_z7(1);
Q(1) = rho * cp * temp.T_z1(1) * x(end) - rho * cp * x(1) * T_z0(1) / 2 - rho * cp *↙
delta_z * ...
```

```
           (temp.T_z1(1) * x(1) / (2 * delta_z) + temp.T_z1(1) / 2 + temp.T_z2(1) +↙
temp.T_z3(1) + ...
           temp.T_z4(1) + temp.T_z5(1) + temp.T_z6(1) + temp.T_z7(1) / 2);

% loop for remaining time steps

for t = 2:length(length(tstar))

    dT_z7Dt = gradient(temp.T_z7);        % temporal gradient of thermoelement 7
    T = round(tstar(t)*length(temp.t)) / delta_t;
    Q(t) = rho*cp*(+ ...
           delta_z1*(T_z0star(t)+temp.T_z1(T))/2 +...       % Q in upper interval↙
between surface and first thermoelement
           delta_z/2*(temp.T_z1(T) + 2*temp.T_z2(T) + 2*temp.T_z3(T) + 2*temp.T_z4(T) +↙
2*temp.T_z5(T) + 2*temp.T_z6(T) + temp.T_z7(T)) + ...   % delta Q in all thermoelements
           delta_z7*(T_z7(T)+T_z8(T))/2);           % Q in lower interval between bottom↙
and seventh thermoelement

       % calculate heat flux

    qstar(t) = -(Q(t) - Q(t-1)) / (temp.t(1,T) - temp.t(1,T-1));

end

%% Differential Equation method

% define geometry and inital values

x = [1 4.5 8 11.5 15 18.5 22] / 1000 ;  % depth of measurement points H1 to H7 in Meter

delx1 = x(1) - 0;
delx2 = x(2) - 0;
delx = x(2) - x(1);

k = lambda;
T0 = temp.T_z1(1,1);
L = x(end);

%% coefficients for analytical solution

c = rho * cp * delx1 / 2;
A = -k * (delx1 + delx2) / (delx1 * delx2);

D = zeros(1,length(temp.t));
B = zeros(1,length(temp.t));
qL = zeros(1,length(temp.t));

for t = 1:length(temp.t)

    T1 = temp.T_z1(1,t);
    T2 = temp.T_z2(1,t);
    T3 = temp.T_z3(1,t);
    T4 = temp.T_z4(1,t);
    T5 = temp.T_z5(1,t);
    T6 = temp.T_z6(1,t);
    T7 = temp.T_z7(1,t);
    D(t) = -rho * cp * T0 * L + rho * cp * delx * ...
           (T1 * delx1 / (2 * delx) + T1 / 2 + T2 + T3 + T4 + T5 + T6 + T7 / 2);

    B(t) = k * (T1 * delx2^2 - T2 * delx1^2) / (delx1 * delx2 * (delx1 + delx2));
    qL(t) = k * (T7 - T6) / delx;
end

% calculate analytical solution with Integral K(t)

Ddot = zeros(1,length(temp.t));
K = zeros(1,length(temp.t));
T_w = zeros(1,length(temp.t));
delt = temp.t(1,2) - temp.t(1,1);
c1 = T0 ;

for t = 2:length(temp.t)

    Ddot(t) = (D(t) - D(t-1)) / delt;
    t1 = temp.t(1,t-1);
    t2 = temp.t(1,t);
    K(t) = K(t-1) + delt * (exp( -A * t1 / c) * (B(t-1) - qL(t-1) - Ddot(t-1)) + ...
           exp( -A * t2 / c) * (B(t) - qL(t) - Ddot(t))) /2;
    T_w(t) = exp(A * t2 / c) * (c1 + K(t));

end

%% Finite Difference method 1-dimensional calculation

% geometric parameters

delta_z1 = -temp.Position_z1(1,1);      % m     distance from first thermoelement to↙
surface
delta_z2 = -temp.Position_z2(1,1);      % m     distance from second thermoelement to↙
surface
delta_z = delta_z2 - delta_z1;          % m     distance between thermoelements

t = temp.t;                             % s     time vector

% read temperature in struct with average over "meanWidth"

meanWith = 3;
T_z1 = movmean(temp.T_z1,meanWith);     % °C    temperature thermoelement z1
T_z2 = movmean(temp.T_z2,meanWith);     % °C    temperature thermoelement z2
T_z3 = movmean(temp.T_z3,meanWith);     % °C    temperature thermoelement z3

dT_z1Dt = gradient(T_z1,(1/temp.samplerate));     % temporal gradient thermoelement z1

% Preallocate Memory

c    = zeros(length(temp.t),1);
q_z0 = zeros(length(temp.t),1);
T_z0 = zeros(length(temp.t),1);

% iteration for every time step
```

```
for it=1:1:length(temp.t)

    % calculate factor c of cubix approximation

    c(it) = -(cp*dT_z1Dt(it)*rho)/(6*delta_z*lambda) + (T_z1(it)-2*T_z2(it)+T_z3(it))/↙
(6*delta_z^3);

    % calculate upper surface temperature (z=0mm)

    T_z0(it) = T_z1(it) + ((-4*c(it)*delta_z^3+3*T_z1(it)-4*T_z2(it)+T_z3(it))*delta_z1)↙
/(2*delta_z) - ((6*c(it)*delta_z^3-T_z1(it)+2*T_z2(it)-T_z3(it))*delta_z1^2)/↙
(2*delta_z^2) - c(it)*delta_z1^3;

    % calculate inverse heat flux

    q_z0(it) = lambda*(-(-4*c(it)*delta_z^3+3*T_z1(it)-4*T_z2(it)+T_z3(it))/(2*delta_z)↙
+ ((6*c(it)*delta_z^3-T_z1(it)+2*T_z2(it)-T_z3(it))*delta_z1)/(delta_z^2) + 3*c(it)↙
*delta_z1^2);

end

% save data to struct

heatFD1D.measurement_number = temp.measurement_number;
heatFD1D.solver      = 'heatFluxes_FD 1D (Ilia)';
heatFD1D.meanWith    = meanWith;
heatFD1D.t           = temp.t';
heatFD1D.q_z0        = q_z0;
heatFD1D.T_z0        = T_z0;

%% Finite Difference method 2-dimensional calculation

% geometric parameters

delta_z1 = -temp.Position_z1(1,1);      % m     distance from first thermoelement to↙
surface
delta_z2 = -temp.Position_z2(1,1);      % m     distance from second thermoelement to↙
surface
delta_z = delta_z2 - delta_z1;          % m     distance between thermoelements in axial↙
direction
delta_r = temp.Position_r1(1,2) - temp.Position_z1(1,2); % m     distance between↙
thermoelements in radial direction


t = temp.t;                             % s     time vector

% read temperature in struct with average over "meanWidth"
% meanWith = 2 * 95/0.5; % width definied by samplerate and drop frequency times 2

meanWith = 1;
T_z1 = movmean(temp.T_z1,meanWith);     % °C    temperature thermoelement z1
T_z2 = movmean(temp.T_z2,meanWith);     % °C    temperature thermoelement z2
T_z3 = movmean(temp.T_z3,meanWith);     % °C    temperature thermoelement z3
T_r1 = movmean(temp.T_r1,meanWith);     % °C    temperature thermoelement r1
```

```
T_r2 = movmean(temp.T_r2,meanWith);     % °C    temperature thermoelement r2
T_r3 = movmean(temp.T_r3,meanWith);     % °C    temperature thermoelement r3
T_r4 = movmean(temp.T_r4,meanWith);     % °C    temperature thermoelement r4

dT_z1Dt = gradient(T_z1,(1/temp.samplerate));      % temporal gradient thermoelement z1
dT_z2Dt = gradient(T_z2,(1/temp.samplerate));      % temporal gradient thermoelement z2

% Preallocate Memory

c     = zeros(length(temp.t),1);
q_z0  = zeros(length(temp.t),1);
T_z0  = zeros(length(temp.t),1);
T_z1r = zeros(length(temp.t),1);
T_z2r = zeros(length(temp.t),1);


for it=1:1:length(temp.t)

    % calculate temperature at position (z=4.5mm; r=3.5mm)

    T_z2r(it) = (cp*rho*delta_r^2*delta_z^2*dT_z2Dt(it) - delta_r^2*lambda*T_z1(it) +↙
2*delta_r^2*lambda*T_z2(it) + 4*delta_z^2*lambda*T_z2(it) - delta_r^2*lambda*T_z3(it))/↙
(4*delta_z^2*lambda);

    % calculate temperature at position (z=1mm; r=3.5mm)

    T_z1r(it) = T_r1(it) + (T_z2r(it)-T_r1(it))/(0.5E-3+3.5E-3)*0.5E-3;

    % calculate factor c of cubix approximation

    c(it) = -(cp*dT_z1Dt(it)*rho)/(6*delta_z*lambda) + (T_z1(it)-2*T_z2(it)+T_z3(it))/↙
(6*delta_z^3) - 2*(T_z1(it)-T_z1r(it))/(3*delta_r^2*delta_z);

    % calculate upper surface temperature (z=0mm)

    T_z0(it) = T_z1(it) + ((-4*c(it)*delta_z^3+3*T_z1(it)-4*T_z2(it)+T_z3(it))*delta_z1↙
/(2*delta_z) - ((6*c(it)*delta_z^3-T_z1(it)+2*T_z2(it)-T_z3(it))*delta_z1^2)/↙
(2*delta_z^2) - c(it)*delta_z1^3;

    % calculate inverse heat flux

    q_z0(it) = lambda*(-(-4*c(it)*delta_z^3+3*T_z1(it)-4*T_z2(it)+T_z3(it))/(2*delta_z)↙
+ ((6*c(it)*delta_z^3-T_z1(it)+2*T_z2(it)-T_z3(it))*delta_z1)/(delta_z^2) + 3*c(it)↙
*delta_z1^2);

end

% save data to struct

heatFD2D.measurement_number = temp.measurement_number;
heatFD2D.solver     = 'heatFluxes_FD 2D (Ilia)';
heatFD2D.meanWith   = meanWith;
heatFD2D.t          = temp.t';
heatFD2D.q_z0       = q_z0;
heatFD2D.T_z0       = T_z0;
```