Thermal transport in PDMS microfluidics

Master's Thesis, 10.6.2021

Author:

Arttu Huikuri

Supervisor:

Andreas Johansson
Jussi Hiltunen (VTT)

UNIVERSITY OF JYVÄSKYLÄ
DEPARTMENT OF PHYSICS

Abstract

Huikuri, Arttu
Thermal transport in PDMS; measurements and simulations
Master's thesis
Department of Physics, University of Jyväskylä, 2021, 72 pages.

Thermal transport in polydimethylsiloxane (PDMS), which is used to fabricate microfluidic platforms, was modelled with finite element method (FEM) simulations and the results of the simulations were compared to experimental results measured from a PDMS sample. In steady-state heating simulations all of the results were within 0.81K of each other and most of the results were within 0.2K of each other. In time-dependent heating measurements and simulations the temperature of the PDMS was found to change faster in the simulations than the experiments. FEM was then used to simulate the heating of PDMS microfluidics with two different heaters. Using a hot plate heater the temperature gradient over the PDMS microfluidic was simulated. With point heaters FEM simulations were used with the bisection method optimization algorithm to find optimal heating power values with an error tolerance of 0.1mW.

Keywords: Finite element method, negative temperature coefficient thermistor, polydimethylsiloxane

Tiivistelmä

Huikuri, Arttu
Thermal transport in PDMS; measurements and simulations
Pro gradu -tutkielma
Fysiikan laitos, Jyväskylän yliopisto, 2021, 72 sivua

Lämmön siirtymistä polydimetyylisiloksaanissa (PDMS), jota käytetään mikrofluidisten alustojen valmistuksessa, mallinnettiin elementtimallinnussimulaatioilla (FEM) ja simulaatioiden tuloksia verrattiin kokeellisiin tuloksiin, jotka mitattiin PDMS -näytteestä. Pysyvän tilan kuumennussimulaatioissa kakki tulokset olivat 0.81K sisällä toisistaan ja useimmat tulokset olivat 0.2K sisällä toisistaan. Ajasta riippuvissa kuumennusmittauksissa ja -simulaatiossa PDMS:n lämpötila muuttui nopeammin simulaatioissa kuin kokeissa. Elementtisimulaatioita käytettiin myös PDMS mikrofluidiikkojen simulaatioon kahdella kuumentimella. Lämpölevykuumennusta käyttämällä simuloitiin lämpögradientti PDMS mikrofluidiikan yli. Pistelämmönlähteiden kanssa FEM simulaatioita käytettiin optimisaatioalgoritmin bisektiometodin kanssa optimaalisen lämmitystehon löytämiseksi 0.1mW virhetoleranssilla.

Avainsanat: Elementtimenetelmä, termistori, polydimetyylisiloksaani

Preface

The measurements and simulations shown in this work were done at VTT technical research center of Finland. This thesis was written under the supervision of Andreas Johansson at the university of Jyväskylä. I would like to acknowledge for their contributions my thesis advisor Andreas Johansson, Jussi Hiltunen for supervising my work at VTT, Sanna Aikio for aid with Comsol and designing a 3D printed plastic frame for the experiments, Sari Pohjonen for aid in PDMS moulding, Rami Aikio for designing an interface circuit for the NTC temperature measurements, Eero Hietala for designing the LabView program used in the measurements and Jarno Petäjä for aiding in running the steady-state simulations.

Jyväskylä June 10, 2021

Arttu Huikuri

Contents

1 Introduction

In the process of developing solutions to physical engineering problems, computational modelling and simulations are often used as tools to help with design. In this work, finite element method (FEM) simulations are used to aid in designing heaters for VTT:s roll-to-roll printed microfluidic platforms. [1] These platforms are used in nucleic acid amplification reactions such as the polymerace chain reaction (PCR) and loop-mediated isothermal amplification (LAMP). LAMP requires a specific temperature [2] and in PCR there are multiple reaction steps that require different temperatures [3].

The platforms are modelled in the Comsol Multiphysics simulation program. In the finite element method a geometry is modelled as a grid of elements and their connections called a mesh. A system of partial differential equations is used to calculate the values of the dependent variables in each grid node. The simulation is therefore a discrete approximation of a continuous geometry. The accuracy of the approximation depends on many factors, such as the size of the mesh elements, boundary and initial conditions, material properties and choice of solver.

The main material of the microfluidics is polydimethylsiloxane (PDMS). In order to validate the use of FEM simulations for heat transport in PDMS a set of experimental measurements on a physical PDMS sample were made and their results were compared to FEM simulations done with a model made to closely approximate the physical sample. The sample was made with negative temperature coefficient thermistors (NTCs) embedded inside the PDMS. The NTCs were calibrated using a thermocouple and the Hoge-2 equation [4] in order to use them to measure temperatures inside the PDMS.

The temperature gradient across the sample was measured with the NTCs as the sample was heated with a hot plate. The heating was simulated and the results of the experiments and simulations were compared to verify the accuracy of FEM simulations for PDMS samples. Several steady-state measurements and simulations were done with different hot plate temperatures. Time-dependent measurements and simulations were also done both during the heating and cooling of the sample.

In the simulation model, the heads of the NTCs were modelled inside PDMS in positions measured from the physical sample. Simplifying approximations such as leaving the wires of the NTCs out of the simulated model were made. The top of the PDMS was modelled with a natural convection heat transfer boundary condition. The temperature of the bottom of the model was set to a temperature measured with an NTC from between the sample and the heater during the experiments. The results of the simulation were the volume averages of the temperatures of the NTC heads.

In the steady-state cases the simulation method was found to produce results close to the experimental results. In the time-dependent cases the simulations had the temperature of the PDMS to change faster than the measured data. In the time-dependent case the end of the simulations corresponded closely to the steady state case, showing a similar accuracy to the steady-state simulation. The difference is therefore in the thermal diffusivity of the PDMS with different values in the simulation and experiments. Therefore as the FEM simulation method is used for simulations of the heatings of the microfluidic platforms this discrepancy should be taken into account when using the results.

After the simulation method was validated two different heater designs for the microfluidic platforms were simulated. In one the entire platform was heated from below with a hot plate. These simulations were done with both steady-state and time-dependent simulations. From the steady-state simulations the temperature gradient across the platform was simulated. From the time-dependent simulations the time it takes for the fluid to reach a desired temperature for the nucleic acid amplification reactions was determined. Taking into account the discrepancy between the results of the verification measurements and simulations the heating of an experimental sample should be expectedted to take longer than the simulated time. The heating time is a constraint on the amplification reactions because the reaction does not start before the reaction temperature is reached.

In the other simulated heater design the platform was heated with point heaters below the microfluidic wells were the amplification reactions occur. A series of steady-state simulations was done and the bisection method optimization algorithm was used in order to find a heater power that would raise the temperature of the fluid to the desired temperature. This power value can then be used to design a heater circuit where the heating power of the point heaters heats the fluid in the

wells.

The simulations represent a step in the design process and additional work and revisions are required to produce a working heater solution. The results of the FEM simulations can be used to inform initial heater designs but the heaters will need to be tested experimentally to ensure that the temperature of the fluid is optimal for the nucleic acid amplification reactions.

2 Theory

In sections 2.1-2.3 the theory of thermal transport is discussed. The equations that govern thermal transport are used in the finite element simulations for which theory is discussed in section 2.5. Section 2.4 discusses negative temperature coefficient thermistors, their calibration and error propagation in temperature measurements with them.

2.1 Thermal conduction

In both solids and fluids Thermal conduction is the transport of heat in the material without the movement of the material itself. The conductive thermal flux q'' per unit area and unit time is given by Fourier's law [5, 6]:

$$q'' = -k\nabla T, \tag{1}$$

where k is the thermal conductivity of the material and the temperature gradient $\nabla T(r,t)$ is the vector normal of the isothermal surface. Thermal conductivity is positive and scalar for homogeneous, isotropic materials, but does vary by temperature. The negative sign ensures that the direction of the thermal flux is the direction of decreasing temperature. [5, 6]

Equation (1) can describe steady-state thermal conduction through a stationary material, but for solutions to time-dependent problems and moving materials, a more complicated heat transfer equation has to be used. [5, 6] If the thermal conductivity is constant, conduction heat transfer can be modelled with equation

$$\nabla^2 T + \frac{g}{k} = \rho c \frac{\partial T}{\partial t}, \tag{2}$$

where ρ is the density of the material, t is time, c is the specific heat, and g is the volumetric rate of internal energy generation, with a SI unit of W/m^3. [5, 6] A positive g denotes increasing internal energy and a negative g denotes a decreasing

internal energy. [6] Thermal diffusivity α is defined as [5, 6]

$$\alpha = \frac{k}{\rho c}. \tag{3}$$

Thermal diffusivity is associated with the speed of propagation on heat caused by a perturbation of temperature. [5, 6]

2.2 Thermal convection

In fluids, heat is transported both through conduction and with the movement of the fluid itself. The transport with the movement of the fluid is called convection.

2.2.1 Fluid flow

In order to characterize convection, it is important to characterize the fluid flow. In laminar flow, fluid flows in parallel layers, which do not disturb each other. [6] In turbulent flow, chaotic changes in velocity and pressure occur, causing the fluid to flow in vortices and eddies and causing fluid layers to mix. [6] Transitional flow has characteristics from both of the aforementioned flow regimes.

Several dimensionless numbers are used to help characterize fluid flow. The Reynolds number is interpreted as the ratio of inertia to viscous forces in the fluid [6]

$$Re = \frac{\rho||U||L}{\mu}, \tag{4}$$

where ρ, μ are the density, and the the dynamic viscosity of the fluid and $U = (u, v, w)$ is the vector velocity of the fluid in Cartesian coordinates and L is the characteristic length. A low Reynolds number indicates laminar flow while a high Reynolds number indicates turbulent flow. [6]

The Prandtl number can be used to characterize the fluid [6, 7]

$$Pr = \frac{\nu}{\alpha} = \frac{c_p \mu}{k}, \tag{5}$$

where α is the thermal diffusivity defined in section 2.1 and $\nu = \mu/\rho$ is the kinematic viscosity of the fluid. [6]

In the laminar regime, fluid flow can be modelled with conservation equations for mass, momentum and energy. The mass equation for a viscous, compressible fluid is

[8]

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z}. \tag{6}$$

The corresponding momentum equations are [8]

$$\begin{aligned}
\frac{\partial \rho u}{\partial t} + \rho u\frac{\partial u}{\partial x} + \rho u\frac{\partial v}{\partial y} + \rho u\frac{\partial w}{\partial z} &= -\frac{\partial p}{\partial x} + \mu\frac{\partial^2 u}{\partial x^2} + \mu\frac{\partial^2 u}{\partial y^2} + \mu\frac{\partial^2 u}{\partial z^2} \pm S'_u \\
\frac{\partial \rho v}{\partial t} + \rho v\frac{\partial u}{\partial x} + \rho v\frac{\partial v}{\partial y} + \rho v\frac{\partial w}{\partial z} &= -\frac{\partial p}{\partial x} + \mu\frac{\partial^2 v}{\partial x^2} + \mu\frac{\partial^2 v}{\partial y^2} + \mu\frac{\partial^2 v}{\partial z^2} \pm S'_v \\
\frac{\partial \rho w}{\partial t} + \rho w\frac{\partial u}{\partial x} + \rho w\frac{\partial v}{\partial y} + \rho w\frac{\partial w}{\partial z} &= -\frac{\partial p}{\partial x} + \mu\frac{\partial^2 w}{\partial x^2} + \mu\frac{\partial^2 w}{\partial y^2} + \mu\frac{\partial^2 w}{\partial z^2} \pm S'_w,
\end{aligned} \tag{7}$$

where the terms S'_u, S'_v and S'_w are the source or sink terms. [8] The energy equation is [8]

$$\begin{aligned}
\frac{\partial(\rho H)}{\partial t} + \frac{\partial(\rho u H)}{\partial x} + \frac{\partial(\rho v H)}{\partial y} + \frac{\partial(\rho w H)}{\partial z} =& \frac{\partial}{\partial x}\left[k\frac{\partial T}{\partial x}\right] + \frac{\partial}{\partial y}\left[k\frac{\partial T}{\partial y}\right] + \frac{\partial}{\partial z}\left[k\frac{\partial T}{\partial z}\right] \\
&+ \frac{\partial p}{\partial t} + \Phi + S_T
\end{aligned} \tag{8}$$

where H is enthalpy, k is the thermal conductivity of the fluid, S_T is the heat source or sink term and Φ is the dissipation function [8]

$$\begin{aligned}
\Phi =& \frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(u\tau_{zx})}{\partial z} + \frac{\partial(v\tau_{xy})}{\partial x} + \frac{\partial(v\tau_{yy})}{\partial y} + \frac{\partial(v\tau_{zy})}{\partial z} \\
&+ \frac{\partial(w\tau_{xz})}{\partial x} + \frac{\partial(w\tau_{yz})}{\partial y} + \frac{\partial(w\tau_{zz})}{\partial z},
\end{aligned} \tag{9}$$

where τ are the normal and tangential viscous stresses. [8]

2.2.2 Convective cooling

Newton's law of convective cooling states that the heat power transferred between a solid surface and fluid is [6]

$$\frac{Q}{A} = h\Delta T, \tag{10}$$

where Q is the thermal power transported through a surface with area A, h is the convective heat transfer coefficient and ΔT is the difference in temperature between the fluid and the solid.

In order to characterize h, the geometry of the surface needs to be known. The Nusselt number is defined as [6, 7]

$$Nu_L = \frac{hL}{k} \tag{11}$$

where L is the characteristic length that depends on the surface geometry and k is the thermal conductivity of the fluid.

The Rayleigh number is defined as [6, 7]

$$Ra_L = \frac{g\beta\Delta T L^3}{\alpha\nu}, \tag{12}$$

where α is the thermal diffusivity defined in section 2.1, g is the acceleration of gravity, ΔT is the temperature difference between the surface and the surrounding fluid, ν is the kinematic viscosity and β is the coefficient of thermal expansion. [7]

The magnitude of the convective heat transfer coefficient also depends on the characteristics of the fluid flow. In natural convection the fluid flows past the surface driven by buoyancy and gravitation effects that are caused by internal differences in the temperature of the fluid. [6] In forced convection the fluid moves past the surface driven by an external force such as a fan. [6]

For a horizontal hot plate, the characteristic length L is defined as the area of the plate divided by the perimeter of the plate. [7] In the case natural convection from a horizontal hot plate facing upward with turbulence taken into account, there is a relationship between the Nusselt and Rayleigh numbers if the Prandtl number is greater than 0.5: [7]

$$\overline{Nu_L} = \begin{cases} 0.54Ra_L^{1/4}, 10^4 < Ra_L < 10^7 \\ 0.15Ra_L^{1/3}, 10^7 < Ra_L < 10^9, \end{cases} \tag{13}$$

where $\overline{Nu_L}$ is the time-averaged Nusselt number. Time-averaging is used to describe turbulent flows. [6, 7] By solving h from equation (11) and substituting equations (13) and (12) into it, The heat transfer coefficient for the horizontal hot plate can be solved. This heat transfer coefficient can then be substituted into equation (10) to make a Robin boundary condition described in equation (16) between a hot, horizontal solid and naturally convecting fluid on top of it.

2.3 Boundary conditions

In order to solve the temperature distribution with equation (2) initial conditions and boundary conditions must be set. [5, 6] Boundary conditions are used in simulations to model the interaction of the simulation model with its environment. Initial conditions represent the initial values of the dependent variable which in the case of thermal transport is temperature. Therefore the initial condition to conduction problems is the temperature distribution $T(r, t = 0)$ in the medium. [5, 6] Boundary conditions can be divided into several categories. In homogeneous boundary conditions, all non-zero terms contain the dependent variable or its derivative. [5] Boundary conditions of the first type are Dirichlet boundary conditions, where the dependent variable is prescribed a constant value. [5, 6] In the case of temperature as the dependent variable [5]

$$T|_{surface} = f(\hat{r}, t), \tag{14}$$

where $f(\hat{r}, t)$ is the surface temperature distribution. If $f(\hat{r}, t) = 0$, the boundary condition is homogeneous. Boundary conditions of the second type are Neumann boundary conditions, where the derivate of the dependent variable is prescribed. Here, this is represented by a prescribed thermal flux [5]

$$-k\frac{\partial T}{\partial n}|_{surface} = g(\hat{r}, t) \tag{15}$$

where $g(r, t)$ is the surface thermal flux distribution and n is the outward-drawn norm of the surface. [5] If the heat flux is zero, that surface is perfectly insulated and the boundary condition is homogeneous. Boundary conditions of the third type are Robin boundary conditions. In these, both the dependent variable and its derivative appear [5, 6]. Here, this type of boundary conditions represent heat conduction from a surface to ambient fluid

$$-k\frac{\partial T}{\partial n}|_{surface} = h\left[T|_{surface} - T_{\inf}(\hat{r}, t)\right], \tag{16}$$

where $T_{\inf}(\hat{r}, t)$ represents the ambient fluid temperature and h is the heat transfer coefficient between the surface and the fluid.

Other boundary conditions also exist. In an interface between two solids imperfect thermal contact means the materials only make physical contact in some places. Small gaps filled with air or other interfacial fluid occur between these spots of

contact. Heat transfers by conduction through the contact spots and by convection over the gaps but the thermal conductivity is generally much lower than in the solids. [5] Due to the conservation of energy, the thermal fluxes in the surfaces and between them must be equal. The boundary condition is

$$q_i'' = -k_1 \frac{\partial T}{\partial x}|_i = h_c \left(T_1 - T_2\right)_i = -k_2 \frac{\partial T}{\partial x}|_i, \tag{17}$$

where h_c is the contact conductance of the interface. [5] If there is perfect thermal contact between the materials, equations [5]

$$T|_1 = T|_2 \tag{18}$$

$$-k_1 \frac{\partial T}{\partial x}|_i = -k_2 \frac{\partial T}{\partial x}|_i \tag{19}$$

are used instead.

2.4 Thermal measurements

There are numerous ways to measure temperature and other thermodynamic variables [9, 10]. In this section the focus is temperature measurements with negative temperature coefficient thermistors which were used in the experimental portion of this study.

2.4.1 Negative temperature coefficient thermistors

Negative temperature coefficient thermistors (NTCs) are temperature detectors. In these detectors electrical resistance through the detector decreases with increasing temperature. An electrical measurement of the resistance in used to measure the temperature. [9, 10] NTCs are usually made from ceramic semiconductors, such as metal oxides. [9, 10] The NTCs used in this work are also surrounded by a glass casing used to protect the ceramics inside. [11]

Measuring the resistance of the NTC can be done by measuring the voltage over the NTC using a voltage divider circuit. The circuit is shown in figure 1. The resistance of the NTC is

$$R_{NTC} = R_{ref} \frac{V}{V_{in} - V}, \tag{20}$$

where R_{ref} is the resistance of a reference resistor, V is the measured voltage over

the NTC and V_{in} is the input voltage of the circuit.

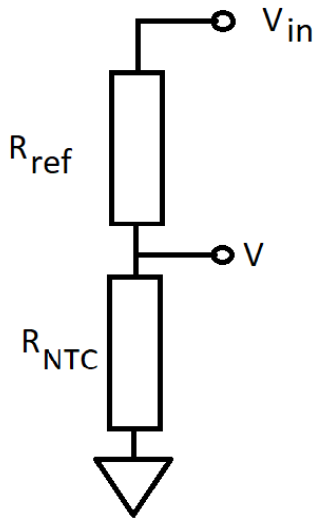

Figure 1. A simple voltage divider circuit that can be used to measure the resistance of the NTC by measuring the input voltage and the voltage over the NTC.

In order to convert the measured resistance into an accurate temperature value the NTC needs to be calibrated. [9, 10, 12] To minimize errors, the NTC should be calibrated with its interface circuit. [10] The interface circuit measures the resistance of the NTC. Calibration is done by placing the NTC in several known temperatures and the resistance values of the NTC are measured in these temperatures [10]. These calibration points are then used to characterize the transfer function of the NTC by fitting the transfer function numerically to the calibration points. [10]

The calibration should be done over the entire measurement temperature range and for each NTC seperately. [10] Outside the calibration range the uncertainty of the calibration increases significantly. [13] The transfer functions for NTCs are discussed in section 2.4.2.

2.4.2 NTC transfer functions and calibration

A simple transfer function for NTCs is [9, 10, 12]

$$R_T = R_{T0} e^{\beta\left(\frac{1}{T} - \frac{1}{T_0}\right)}, \tag{21}$$

where R_T is the NTCs resistance in temperature T, with T_0 being a specific reference temperature, usually 25^oC [12] and β is the calibration coefficient.

Equation (21) gives the resistance of the NTC as a function of the measured temperature. In temperature measurement, the temperature is measured as a function of the resistance and therefore a transfer function which represents this relationship is needed. There are several functions that can be used for this purpose [12–14] such as the functions proposed by Hoge. [4] Liu et al. find the Hoge-2 equation to be the best for high-accuracy temperature measurement. [12, 13] The Hoge-2 equation is [4]

$$\frac{1}{T} = A_1 + A_2 \ln(R_T) + A_3 \left(\ln(R_T)\right)^2 + A_4 \left(\ln(R_T)\right)^3, \tag{22}$$

where R_T is the resistance of the NTC at temperature T and A_1, A_2, A_3 and A_4 are the calibration coefficients.

White [15] suggests using a slightly modified verison of the Hoge equations for calibration where R_T is divided by one of the calibration points, R_{T_0}. The modified Hoge-2 equation is

$$\frac{1}{T} = A_1 + A_2 \ln\left(\frac{R_T}{R_{T_0}}\right) + A_3 \left(\ln\left(\frac{R_T}{R_{T_0}}\right)\right)^2 + A_4 \left(\ln\left(\frac{R_T}{R_{T_0}}\right)\right)^3. \tag{23}$$

The values of the calibration coefficients are solved by least-squares fitting [16] equation (23) to calibration points that are resistance-temperature value pairs. At least as many calibration points as calibration coefficients are needed, but the accuracy of the calibration increases significantly with a higher amount of calibration points. [13]

After the calibration is done equation (23) can be solved for temperature.

$$T = \frac{1}{A_1 + A_2 \ln\left(\frac{R}{R_{T_0}}\right) + A_3 \left(\ln\left(\frac{R}{R_{T_0}}\right)\right)^2 + A_4 \left(\ln\left(\frac{R}{R_{T_0}}\right)\right)^3}, \tag{24}$$

Where R is the measured NTC resistance and T is the temperature value measured

by the NTC. By substituting R and R_{T_0} in equation (24)with equation (20) the temperature measured by the NTC in a voltage divider circuit shown in figure 1 is

$$T = \frac{1}{A_1 + A_2 \ln\left(\frac{V(V_{in_0}-V_0)}{V_0(V_{in}-V)}\right) + A_3\left(\ln\left(\frac{V(V_{in_0}-V_0)}{V_0(V_{in}-V)}\right)\right)^2 + A_4\left(\ln\left(\frac{V(V_{in_0}-V_0)}{V_0(V_{in}-V)}\right)\right)^3}, \quad (25)$$

where V is the measured voltage over the NTC, V_{in} is the input voltage of the circuit and V_0 and V_{in0} the measured and input voltages of the calibration point R_{T_0}. This removes the effect of R_{ref} from the equation as it is the same for both R and R_{T_0}.

2.4.3 Error propagation

In an NTC temperature measurement the temperature reading is affected by the errors of the measured NTC resistance, but also by the accuracy of the calibration. Therefore the error propagation through the measurement should be considered. Liu et al. [13] proposed a method for determining the error propagation in a temperature measurement using least-squares fitting calibration and the Hoge [4] equations. [13] The uncertainty of a NTC temperature reading according to them is [13]

$$\Delta T = \sqrt{\sum_{i=1}^{m}\left(\frac{\partial T}{\partial R_i}\right)^2 \Delta R_i^2 + \sum_{i=1}^{m}\left(\frac{\partial T}{\partial T_i}\right)^2 \Delta T_i^2 + \left(\frac{\partial T}{\partial R}\right)^2 \Delta R^2}, \quad (26)$$

where (R_i,T_i) are the calibration points used, R is the measured NTC resistance, and $T\,(R, R_1,...R_m; T_1,...,T_m)$ is the temperature value measured with the NTC with m being the amount of calibration points. $\Delta T, \Delta R$,ΔT_i and ΔR_i are the uncertainties of the variables. [13]

If equations (23), (24) and (25) are used, uncertainty should be calculated from the variables that were measured. The calibration points are $(R_i(V_i, V_{in_i}); T_i)$, where V_i and V_{in_i} are the measured voltages from the voltage divider circuit in equation (20) at temperature T_i. Equation (26) becomes

$$\begin{aligned}\Delta T = \Bigg\{ &\sum_{i=1}^{m}\left(\frac{\partial T}{\partial V_i}\right)^2 \Delta V_i^2 + \sum_{i=1}^{m}\left(\frac{\partial T}{\partial V_{in_i}}\right)^2 \Delta V_{in_i}^2 + \sum_{i=1}^{m}\left(\frac{\partial T}{\partial T_i}\right)^2 \Delta T_i^2 \\ &+ \left(\frac{\partial T}{\partial V}\right)^2 \Delta V^2 + \left(\frac{\partial T}{\partial V_{in}}\right)^2 \Delta V_{in}^2 + \left(\frac{\partial T}{\partial V_0}\right)^2 \Delta V_0^2 + \left(\frac{\partial T}{\partial V in_0}\right)^2 \Delta V_{in_0}^2 \Bigg\}^{\frac{1}{2}},\end{aligned} \quad (27)$$

According to the chain rule

$$\begin{aligned} \frac{\partial T}{\partial V_i} &= \frac{\partial T}{\partial R_i}\frac{\partial R_i}{\partial V_i}, \qquad \frac{\partial T}{\partial V_{in_i}} = \frac{\partial T}{\partial R_i}\frac{\partial R_i}{\partial V_{in_i}}, \\ \frac{\partial T}{\partial V_0} &= \frac{\partial T}{\partial R_0}\frac{\partial R_0}{\partial V_0}, \qquad \frac{\partial T}{\partial V_{in0}} = \frac{\partial T}{\partial R_0}\frac{\partial R_0}{\partial V_{in0}}, \end{aligned} \tag{28}$$

and therefore equation (27) can be written as

$$\begin{aligned} \Delta T = &\Bigg\{ \sum_{i=1}^{m} \left(\frac{\partial T}{\partial R_i}\frac{\partial R_i}{\partial V_i} \right)^2 \Delta V_i^2 + \sum_{i=1}^{m} \left(\frac{\partial T}{\partial R_i}\frac{\partial R_i}{\partial V_{in_i}} \right)^2 \Delta V_{in_i}^2 + \sum_{i=1}^{m} \left(\frac{\partial T}{\partial T_i} \right)^2 \Delta T_i^2 \\ &+ \left(\frac{\partial T}{\partial R}\frac{\partial R}{\partial V} \right)^2 \Delta V^2 + \left(\frac{\partial T}{\partial R}\frac{\partial R}{\partial V_{in}} \right)^2 \Delta V_{in}^2 \\ &+ \left(\frac{\partial T}{\partial R_0}\frac{\partial R_0}{\partial V_0} \right)^2 \Delta V_0^2 + \left(\frac{\partial T}{\partial R_0}\frac{\partial R_0}{\partial V_{in0}} \right)^2 \Delta V_{in0}^2 \Bigg\}^{\frac{1}{2}}, \end{aligned} \tag{29}$$

The partial derivatives of T in equation (29) are [13]

$$\begin{aligned} \frac{\partial T}{\partial R_i} &= \sum_{j=1}^{n} \left[BH^{-1} \right]_{i,j} \frac{\partial T}{\partial A_j}, \\ \frac{\partial T}{\partial T_i} &= \sum_{j=1}^{n} \left[CH^{-1} \right]_{i,j} \frac{\partial T}{\partial A_j} \end{aligned} \tag{30}$$

where A_j are the calibration coefficients and $[M]_{i,j}$ is the (i,j)th element of matrix M. [13]

The matrices B and C in equations (30) are (m x n) matrices, where m is the amount of calibration points used in the least-squares fit and n is the amount of calibration coefficients A_j in the fitting equation. H is a square (n x n) matrix. The elements of these matrices are [13]

$$B_{i,j} = -\left(\frac{\partial T}{\partial R}\frac{\partial T}{\partial A_j} \right)\Big|_{R=R_i} + \left(T_i - T(R_i) \right) \frac{\partial^2 T}{\partial R \partial A_j}\Big|_{R=R_i}, \tag{31}$$

$$C_{i,j} = \frac{\partial T}{\partial A_j}\Big|_{R=R_i} \tag{32}$$

and

$$H_{i,j} = \sum_{k=1}^{m} \left[\left(\frac{\partial T}{\partial A_i}\frac{\partial T}{\partial A_j} \right) - \left(T_k - T(R_k) \right) \frac{\partial^2 T}{\partial A_i \partial A_j}\Big|_{R=R_i} \right]. \tag{33}$$

The calibration equation used in this work is the Hoge-2 equation (22) which has 4 calibration coefficients. Equation (24) is the Hoge-2 equation solved for T and can also be written with a sum notation

$$T = \frac{1}{\sum_{k=1}^{n} A_k \left(\ln(\frac{R}{R0})\right)^{k-1}}. \tag{34}$$

Now, the first-order partial derivatives of T are

$$\frac{\partial T}{\partial A_j} = -\frac{\left(\ln(\frac{R}{R0})\right)^{j-1}}{\left(\sum_{k=1}^{n} A_k \left(\ln(\frac{R}{R0})\right)^{k-1}\right)^2}, \tag{35}$$

$$\frac{\partial T}{\partial R} = -\frac{\sum_{k=1}^{n} A_k(k-1) \left(\ln(\frac{R}{R0})\right)^{k-2}}{R\left(\sum_{k=1}^{n} A_k \left(\ln(\frac{R}{R0})\right)^{k-1}\right)^2} \tag{36}$$

and

$$\frac{\partial T}{\partial R_0} = \frac{\sum_{k=1}^{n} A_k(k-1) \left(\ln(\frac{R}{R0})\right)^{k-2}}{R_0\left(\sum_{k=1}^{n} A_k \left(\ln(\frac{R}{R0})\right)^{k-1}\right)^2} \tag{37}$$

while the second-order partial derivatives or T are

$$\frac{\partial^2 T}{\partial A_i \partial A_j} = 2\frac{\left(\ln(\frac{R}{R0})\right)^{i-1} \left(\ln(\frac{R}{R0})\right)^{j-1}}{\left(\sum_{k=1}^{n} A_k \left(\ln(\frac{R}{R0})\right)^{k-1}\right)^3} \tag{38}$$

and

$$\begin{aligned}\frac{\partial^2 T}{\partial R \partial A_j} = \Bigg[&(j-1)(\ln(\frac{R}{R0}))^{j-2} \sum_{k=1}^{n} A_k \left(\ln(\frac{R}{R0})\right)^{k-1} \\ &- 2(ln(\frac{R}{R0}))^{j-1} \sum_{k=1}^{n} A_k(k-1) \left(\ln(\frac{R}{R0})\right)^{k-2}\Bigg] / R\left(\sum_{k=1}^{n} A_k \left(\ln(\frac{R}{R0})\right)^{k-1}\right)^3.\end{aligned} \tag{39}$$

The partial derivatives of R_i are

$$\begin{aligned} \frac{\partial R_i}{\partial V_i} &= \frac{V_{in_i}}{(V_{in_i} - V_i)^2}, \\ \frac{\partial R_i}{\partial V_{in_i}} &= \frac{V_i}{(V_{in_i} - V_i)^2} \end{aligned} \tag{40}$$

By substituting equations (35), (36), (38) and (39) to equations (31), (32) and (33) and those equations and equations (36),(37) and (40) to equation (29) the error of the temperature reading given by equation (25) is solved for measured voltages V and V_{in}.

If a measured quantity is gained from averaging independently measured data points, the error of the average is [17, 18]

$$\delta\langle x\rangle = \sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i - \langle x\rangle)^2}, \tag{41}$$

Where $\langle x\rangle$ is the average, x_i are the data points and n is the amount of data points.

If weighed averaging is used on experimental data with errors, the data points are $x_i \pm \sigma_i$, the weighed average is [17, 18]

$$\langle x\rangle = \frac{\sum_i w_i x_i}{\sum_i w_i}, w_i = \frac{1}{\sigma_i^2}. \tag{42}$$

The error of the weighed average is [17, 18]

$$\sigma_{\langle x\rangle} = \frac{1}{\sqrt{\sum_i w_i}}. \tag{43}$$

2.5 Finite element method simulations

The Finite element method (FEM) [19, 20] is a method that is used to simulate various problems in the fields of structural analysis, thermal transport, fluid flow and many others. [20] In FEM simulations continuous geometries such as rods, surfaces or 3D geometries are divided into a finite number of elements. The behaviour of these elements is described by a finite number of parameters. [19] The elements form a model where a continuous object is modelled with a finite system of parameters. This model is an approximation of the physical system and its accuracy depends on several factors that are explored in this section. The behaviour of the model can

then be simulated by forming a system of equations that govern the interactions between the elements and solving the equations numerically.

The process of dividing a geometry into a mesh of nodes and their connections is called meshing. For 3D-problems tedrahedral or hexahedral elements are generally used. [20] Generally, smaller mesh elements are used for important details to make the simulation more accurate. [20] In large uniform areas or areas of less importance, a wider mesh is sufficient. More elements and degrees of freedom increase the computational requirements. The amount of degrees of freedom can often be reduced by simplifying the model and by using symmetries. [20] The meshing is usually at least partially automated. [20] Different meshes or sizes of mesh elements can be used in different areas in a single model.

For field problems such as thermal conduction, a general form of the partial differential equations used with FEM is [19]

$$-\left(\frac{\partial q}{\partial x}+\frac{\partial q}{\partial y}+\frac{\partial q}{\partial z}\right)+Q=C\frac{\partial \phi}{\partial t}, \tag{44}$$

where ϕ is the field variable describing a physical quantity, $q=-D\nabla\phi$ is the flux of the quantity, Q is the source/sink therm describing the rate with which the quantity is generated or destroyed and D and C are material properties. [19] In section 2.1 equation (1) can be used for thermal conduction simulations by substituting ϕ with T, D with thermal conductivity k and C with density multiplied by specific heat ρc in equation (44). In steady-state simulations the equations can be simplified to their time-independent forms while time-dependent simulations require the time-dependent forms of the equations.

Partial differential equations have two forms. The "strong" form, which requires strong continuity on the dependent field variables. [20] The equations discussed in sections 2.1 and 2.2 are the strong forms. Weak forms do not require as strong continuity of the dependent variables. Using the weak forms of equations makes it easier to obtain an approximate solution. [20] There are multiple different ways to create a weak form of a partial differential equation from the strong form. Widely used methods include energy principles an weighted residual methods. [20]

Thermal conductivity, density and specific heat are the material properties used in thermal conduction simulations. In different FEM simulations different material properties are used, such as Young's Modulus and poisson's ratio for stress simulations [20]. Different areas of the model could consist of different materials, and therefore

have different values of material properties. [20]

Boundary conditions are set on the edges of the model and initial conditions for the field variables are given. Improper selections of these conditions cause the simulation to produce inaccurate results or cause there to be no convergence. The kinds of boundary conditions used in thermal transport simulations are discussed in section 2.3. The initial condition in thermal transport simulations is the temperature in each region of the model.

In models where multiple physical phenomena are simulated, coupling these physical phenomena are required if these phenomena can not be solved independently. [19] Section 4.3 presents a model where both thermal transport and fluid flow are simulated. As the fluid transports heat with conduction and differences in temperature cause fluid flow, the phenomena need to be coupled. There are generally two classes of coupled systems: those where coupling occurs on domain interfaces via boundary conditions, and those where the physical phenomena overlap, where coupling is done through the differential equations used in the simulation. [19] In a model with thermal transport in both fluid and solid both of these couplings exist: an interface coupling in between the fluid and solid, and the coupling of thermal transport to the fluid flow in the fluid to simulate convection.

The equations from all the individual elements are collected into a global finite element equation. The assembly at a particular node is done by adding all the contributions of elements connected at a node. [20] A global coordinate system is established for the model. The assembly results in a matrix equation containing the equations from the individual elements.

After the model and global equations are established a solver is used to solve the global equation. Two common types of methods for solving these equations are direct methods and iterative methods. [19, 20] Direct methods operate on fully assembled systems of equations and therefore work well on small equation systems. They demand large storage space. [20] Iterative solver methods work well on larger equation systems and generally avoid fully assembling the equations and therefore save storage space. [20] Improper selections of simulation settings or model parameters can cause the solver to not converge to a solution.

For time-dependent simulations time-stepping is used. The simulation begins with the initial conditions and a new solution is calculated for each time step. There are two common methods of time stepping: explicit and implicit. This is continued

until the set end time is reached. A time history of the solution is established from the solutions at the time steps. [20]

3 Experimental Methods

In this work the temperature gradient over a sample of polydimethylsiloxane (PDMS) was measured using negative temperature coefficient (NTC) thermistors. A temperature gradient over the sample was created by heating the sample from the bottom. The results of the measurements were compared to a simulated model of the same sample that is discussed in section 4.2. The experimental results are shown in sections 5.1 and 5.2 and compared to the simulation results in section 5.4.

3.1 Sample preparation

Seven EPCOS B57541G1103F005 10kΩ NTC thermistors [11] were moulded inside of a disk of Sylgard184 PDMS [21]. The PDMS mould was made into a 14cm diameter plastic petri dish. The mould was made with 149.2 ± 1g of Sylgard184 PDMS base and 14.8 ± 0.5g of Sylgard184 PDMS curing agent. The curing was done in a Memmert UFE-400 oven in 50^{o}C for 80 minutes. Sari Pohjola aided in the PDMS moulding process. The thermistors were held in place during the moulding process by a 3D-printed plastic frame designed by Sanna Aikio. The frame was made to have a top with a large amount of holes both to make the frame less obstructive to air flow and also to hold the wires of the NTCs in place during moulding and the experiments. The frame was also used to place the NTCs at different distances from the bottom of the mould. This allowed the temperature gradient across the mould to be measured. The sample with the frame and the thermistors is shown in figure 2 A.

The plastic petri dish was broken and the NTCs were connected to an interface circuit shown in figure 3. The circuit was designed by Rami Aikio. The interface circuit is made up of a voltage divider circuit for each NTC as discussed in section 2.4. The circuit has Bourns CR0603 10kΩ resistors [22] as the reference resistors and amplifiers with a gain of 1. The voltages were measured with a NI USB6002 DAQ USB device [23].

The NTCs are labelled in this work with numbers from 1 to 7. Their positions inside the mould were measured with a ruler through the translucent PDMS. In

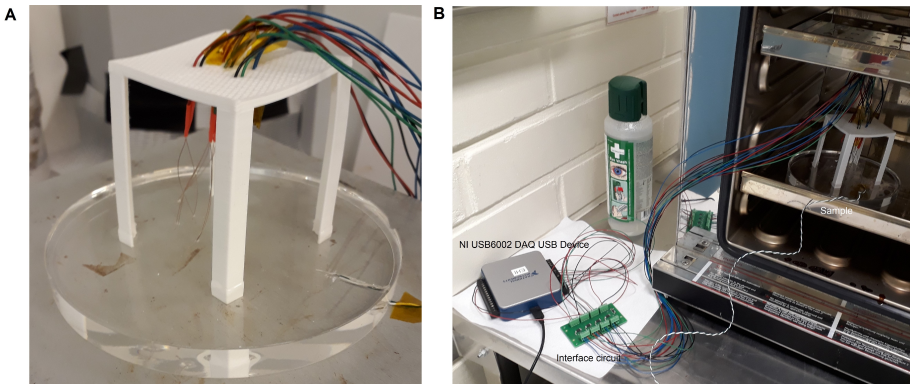

Figure 2. A: The PDMS mould with the white 3D-printed plastic frame, used for holding the NTCs in place during the moulding process. The heads of the NTCs are moulded inside the PDMS. The coloured wires connect the NTCs to the interface circuit. B: An image of the calibration measurement. The PDMS mould sample is in the open Memmert UFE-400 oven and the interface circuit and USB-6002 outside of the oven.

order to measure the NTCs distances from the bottom of the PDMS mould after the measurements detailed in sections 3.2 and 3.3 the PDMS mould was cut into pieces near the NTCs. This way, the measurement could be done closer to the NTC with less distortion from the PDMS material. The thickness of the mould was also measured near each NTC to ensure the mould was uniform in thickness. The mould was found to be 1.00±0.05cm thick in all NTC positions. The measured NTC positions are shown in table 1.

An eight NTC ('NTC 8') was not moulded inside PDMS but was instead used to measure temperature between the mould and a hot plate. One of the NTCs moulded inside the PDMS ('NTC 5') was not used in the measurements because the measurement circuit only had space for seven NTCs. NTC 5 was selected as the one left out because it is the closest in depth to another NTC, NTC 2.

3.2 NTC calibration measurements

The theory of calibrating NTC thermistors is discussed in section 2.4. The calibration was done in a Memmert UFE-400 oven where the temperature was measured by a NI USB-TC01 temperature input device with a K-type thermocouple [24]. Figure 2 B shows the calibration measurement setup with the oven door open.

Voltage over the NTC was measured for each NTC separately and the input voltage was measured jointly for each calibration temperature. LabView [25] was

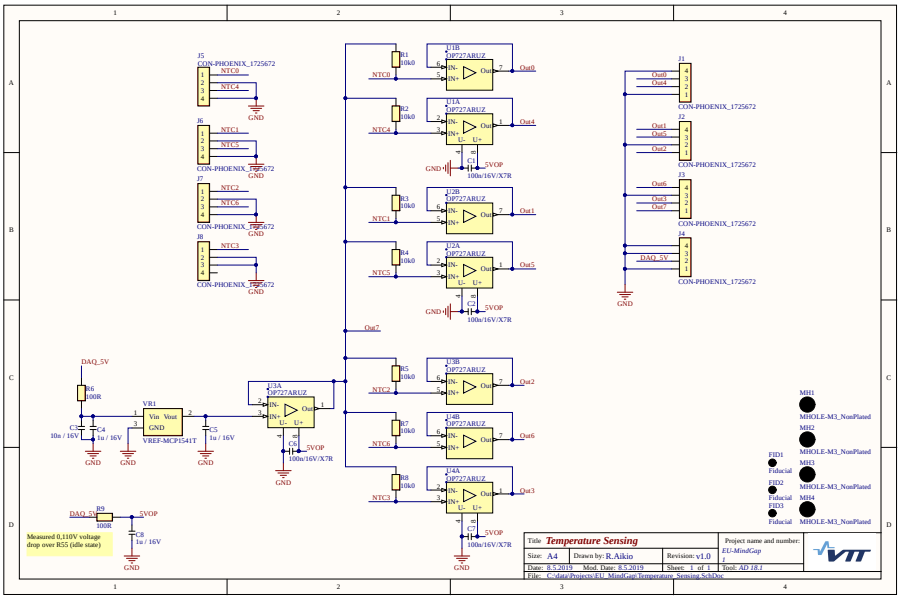

Figure 3. Interface circuit board for NTC temperature measurements, designed by Rami Aikio.

used to record the results. The LabView program used in all the measurements was designed by Eero Hietala. Ten calibration points were measured for each NTC and one of them was chosen as the reference point needed in equation (23). The calibration coefficients were solved with MATLAB by least-squares fitting equation (23) to the calibration points. The results are discussed in section 5.1

The calibration coefficients and the voltages of the reference calibration point were used in LabView to directly calculate temperature from the measured voltage values with equation (25). The calibration errors were calculated with the method discussed in section 2.4.3 for each thermistor over the calibration range which was between 294.70K and 337.72K. The error curves are shown in figure 12. The accuracy of the calibration was deemed appropriate for the verification measurements.

Table 1. NTC locations in the PDMS mould. The z-coordinate was measured from the bottom of the mould to the bottom of each NTC. The x and y axes intersect in the centre of the plastic frame holding the NTCs in place. The thickness of the mould was measured at each NTC location with every result being 1.00±0.05cm, showing that the mould was uniform in depth. NTC 7 touches the top of the mould.

NTC	distance from bottom [cm]	x [cm]	y [cm]
1	0.05±0.05	-0.30±0.05	-0.10±0.05
2	0.35±0.05	-0.80±0.05	1.80±0.05
3	0.50±0.05	-1.60±0.05	0.40±0.05
4	0.75±0.05	1.90±0.05	0.00±0.05
5	0.40 ±0.05	0.40±0.05	3.60±0.05
6	0.55 ±0.05	0.00±0.05	1.80±0.05
7	1.00 (at the top of mould)	-0.60±0.05	-1.40±0.05

3.3 PDMS temperature measurements

In order to create a temperature gradient across the PDMS, the PDMS was heated from below with a Präzitherm type PZ28-2 hot plate. The measurement had to be done in a fume hood for safety reasons. In order to ensure natural convection was cooling the top of the PDMS and not forced convection from the fume hood, aluminium foil was used to make a cover for the experiment. The foil cover blocked air flow to and from the the sample while still allowing space for natural convection to cool the top of the sample. Figure 4 shows the foil cover over the sample. Appendix A contains additional pictures of the experimental set-up.

Using the NTCs embedded into the PDMS mould, the temperature gradient across the PDMS was measured. NTC 8 was used to measure the temperature from between the hot plate and the PDMS mould. Thermal paste was used between the sample and the heater to achieve a better thermal contact between the mould and the heater. The calibration discussed in section 3.2 was used to get accurate results.

Common room temperature and humidity meters were used to measure the temperature and humidity of the air near, but not on top of, the heater. Due to availability a Traceable® 628-0031 was used for most measurements and for some an iRox room temperature meter was used instead. The iRox meter showed the temperature to one decimal place, while the Traceable meter showed no decimal places. The air temperature slightly rose in some of the measurements likely due to

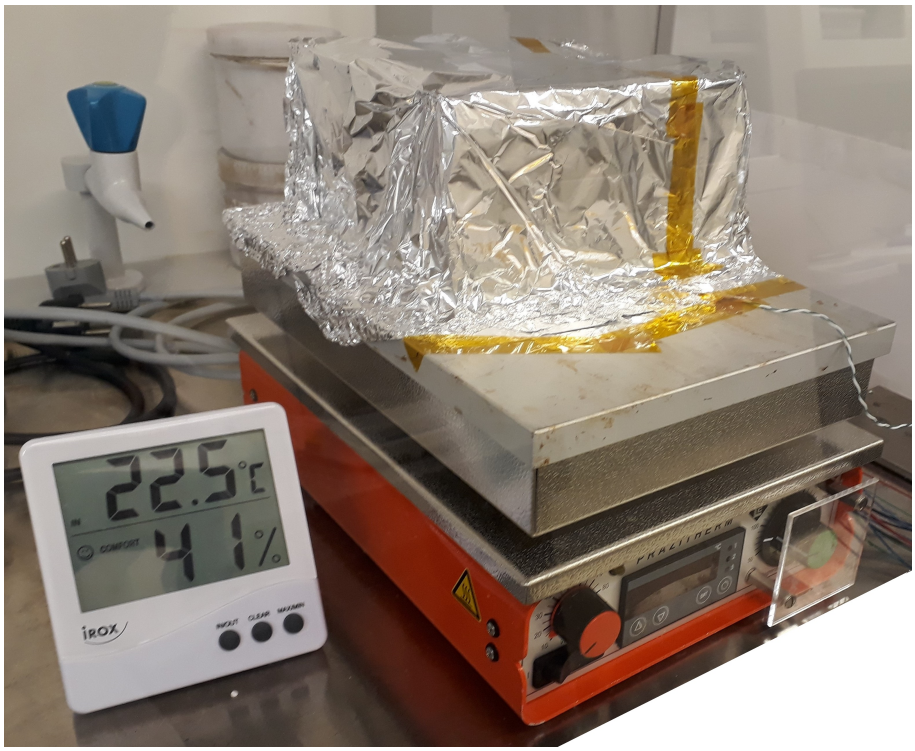

Figure 4. An image of the foil cover used to block forced convection from the laminar cabinet. The room air temperature and humidity meter is next to the heater.

the heater being on for a long time during the experiment.

Two kinds of measurements were made: the steady-state temperature gradient of the PDMS was measured in several different temperatures, and the transient temperature gradient of the PDMS was measured during heating from room temperature to several different temperatures.

During heating, the heater temperature rose linearly with time until it reached the set temperature. After this, the temperature of the plate rose and fell cyclically, staying in the proximity of the set temperature.

The power setting of the hot plate was set to 15% in all the measurements, as a lower power caused the heating to take a long time and a higher settings caused the temperature of the plate to vary more near the set temperature. The maximum temperature setting on the hot plate was always set higher than the desired hot plate temperature.

Six measurements were made with the temperature rising to the set temperatures from room temperature. Measurements 1, 2 and 3 continued even after the heater

Table 2. Verification measurements with NTC thermistors embedded into a PDMS mould. In measurements 1, 2 and 3, data was also recorded as the hot plate was turned off and the mould and hot plate cooled. In measurements 7 and 8, three different steady-states (a, b and c) were measured in series, without cooling the mould to room temperature in between. The relative humidity of the air decreased during measurements 1, 3 and 6 and increased during measurements 2, 5, 7b and 7c. A change in air temperature always means that air temperature rose. An iRox temperature meter was used to measure air temperature and humidity in measurements 5, 7a, 7b and 7c. In the other measurements a Traceable 628-0031 was used.

Measurement	Date	set T_H [K]	T_{air} [^{o}C]	ϕ
1	3.9	313.15	22	51%-44%
2	4.9	318.15	22-23	44%-51%
3	2.9	323.15	24-25	55%-53%
4	2.9	328.15	23	56%
5	27.8	333.15	22.6-23.0	52%-54%
6	3.9.	338.15	22-23	37%-36%
7a	23.8	318.15	23.1-23.3	46%
7b	23.8	328.15	23.4-23.5	46%-47%
7c	23.8	338.15	23.6-23.8	47%-48%
8a	5.9.	313.15	23	48%
8b	5.9	323.15	23	48%
8c	5.9.	333.15	23	49%

was turned off, measuring the cooling of the PDMS in addition to the heating. In two measurements three different steady states were measured in series. Table 2 shows the measured temperatures, air temperatures and humidities. Measurements 4 and 6 were made after measurements 3 and 1 respectively on the same days. The hot plate was allowed to cool for hours but there was still some heat left in the hot plate from the previous measurements. In the simulations corresponding to measurements 4 and 6, this was compensated by slightly changing the initial value of the temperature of the model to match the measured values.

The voltages over the NTCs and the calibrated temperatures were measured with the same measurement set-up as in sections 3.1 and 3.2. The results are shown in figures 13 and 14 and table 6. The results are further discussed in section 5.2.

4 Simulation Methods

4.1 Comsol Multiphysics

The Comsol Multiphysics [26] simulation program was used to complete the finite element method simulations in this work. Version 5.6 was used for the steady-state verification simulations and version 5.4 was used for all the other simulations. Comsol Multiphysics uses the finite element method discussed in section 2.5 to solve partial differential equations for the simulation. The set of finite differential equations used is dependent on the physics model and boundary conditions set for the simulation. In Comsol Multiphysics, different physical phenomena are divided into interfaces, and multiphysics interfaces are used to couple these when needed. The interfaces used in this work are the heat transport in solids and fluids interface, the single-phase flow interface, the microfluidics interface and the multiphysics interface is the nonisothermal flow interface. The heat flux module was used to create the newton cooling boundary condition.

In the Comsol Multiphysics 5.4 interfaces the partial differential equations used in the simulations are always the weak forms of the equations. [26] In strong form equations the differential equations must be satisfied in every point of the geometry. The weak form has a less stringent continuity requirement. Weak form equations are commonly used in FEM simulations. In Comsol Multiphysics 5.4. the strong form of the equations used are converted to the weak forms by multiplying them with an arbitrary test function and integrating the result over the domain [26]. The equations referred to in sections 4.2-4.5 are used by Comsol Multiphysics 5.4. in their weak forms instead of the strong forms.

Comsol multiphysics also has 3D design tools that can be used to make the model geometry that were used in this work, libraries for material properties, and automated tools to create a mesh. The model geometries in this work were made using these tools. The material property libraries were used if the manufacturer provided datasheets did not include relevant material properties. The automated meshing tools were used to create the meshes in used in this work. The size of the

mesh elements can be set and the program automatically creates a mesh based on them. In the 3D models that were used in this work, the free tetrahedral mesh provided by Comsol Multiphysics was used most often.

The "Comsol Multiphysics LiveLink^{TM} for MATLAB" [26] was used in this work to set model variables, extract results, define functions and run models. LiveLink for MATLAB allows the user to run models in loops changing some variable between runs. In time-dependent simulations, it made possible to set variables as a function of time by defining a function in MATLAB and setting the value of the variable to the MATLAB function.

4.2 PDMS verification simulation

In order to validate the use of Comsol Multiphysics in the simulation of PDMS-based microfluidics, verification simulations that compare simulated results with experimental data is needed. This kind of verification has been used with Comsol Multiphysics in different applications, such as heat transfer in buildings [27], hot disc temperature sensors [28] and in food engineering [29]. The experimental methods and measurements are discussed in section 3.3. The simulation model was designed to closely match the physical sample while being simple and allowing reasonably short simulation times.

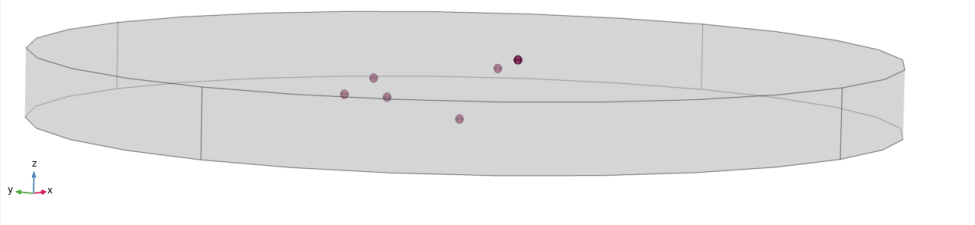

Figure 5. The Comsol Multiphysics geometry used in the verification simulations. The PDMS (grey, translucent) has the glass heads of the NTCs (red) embedded into it.

To create a model that would match closely to the physical sample, the sample dimensions were measured and replicated in the simulated geometry. The PDMS mould was done in a cylindrical petri dish and therefore the PDMS was simulated as a cylinder. The dimensions of the PDMS mold were measured and the values used in the simulation were 14cm as the PDMS diameter and 1.0cm as the PDMS height. The wires and plastic frame shown in figure 2 were not modelled in order to reduce

the complexity of the model. The glass heads of the thermistors were modelled as spheres with a radius of 0.7mm to mach the value given in the NTC datasheet [11]. The NTCs were placed into the model at their measured physical locations. The locations are shown in table 1 and the method they were measured is discussed in section 3.1. The wire-frame geometry of the model is shown in figure 5.

The temperature measured from the hot plate by an NTC in section 3.3 was used as a heat bath temperature below the model. This Dirichlet boundary condition was modelled with equation (14). The vertical walls of the model were modelled to be insulated with no thermal flux through them. This homogeneous Neumann boundary condition was modelled with equation (15). As forced convection of air to and from the sample was blocked in the measurement, the top of the sample and the upper part of the highest NTC were modelled as a boundary heat flux between a horizontal hot plate and naturally convecting, moist air. In the simulation, the air temperature and humidity were from values measured near the experimental set-up. The boundary condition was modelled using the heat flux boundary condition provided in the heat transfer module in Comsol Multiphysics. Moist air was selected as the fluid and the measured values shown in table 2 temperature of the air and the relative humidity were used. Comsol Multiphysics uses equations (11), (12) and (13) to calculate the heat transfer coefficient for a Neumann boundary condition.

The thermal conductivity and density of the PDMS used for the simulation are from the manufacturer's product data sheet [21]. The density was given as the specific gravity of the material. The manufacturer did not specify the temperature of the water used as reference in the specific gravity. The density of water at 25^oC, 997.05kg m^{-3} [30] was chosen and the density calculated as

$$\rho_{\mathrm{PDMS}} = SG_{\mathrm{PDMS}} \cdot \rho_{\mathrm{H_2O}} = 1.03 \cdot 997.05 \frac{\mathrm{kg}}{\mathrm{m}^3} = 1027 \frac{\mathrm{kg}}{\mathrm{m}^3}. \tag{45}$$

The specific heat capacity was not specified by the manufacturer. [21] Therefore the default value used by Comsol was used [26, 31]. The product data sheet values were used instead of the Comsols built-in PDMS material properties in order to make the simulation more accurate to the experimental case. The NTC heads are glass-coated, so they were simulated as glass spheres. The Comsol built-in glass (quartz) properties were used. For the PDMS-air interface, the built-in moist air setting was used to calculate fluid properties. The relevant material properties for

Table 3. Material properties used in the verification simulations. k is the thermal conductivity, ρ is density and C_p is the heat capacity in constant pressure.

Material	k [W/(m·K)]	ρ [kg/m^3]	C_p [J/(kg·K)]
PDMS	0.27 [21]	1027 [21]	1461 [31]
glass	1.4	2210	730
PET	0.19 [32]	1403.5 [33]	1143.8 [32]

the simulation are shown in table 3.

The mesh was an automatically generated physics-controlled mesh with an element size of "extremely fine". With these settings Comsol generates a 3-dimensional free tetrahedral mesh where the mesh element size decreases near objects and edges with small dimensions. In this model, This causes the areas near the NTC heads to have a tighter mesh than areas away from them. A simulation was also done with a significant decrease in mesh size with both the maximum and minimum element sizes multiplied with 0.5. This increases the computation time but did not relevantly change the simulation results, as shown in table 8. Therefore the "extremely fine" mesh size is deemed appropriate for the simulation. Figure 6 shows an image of the meshed model.

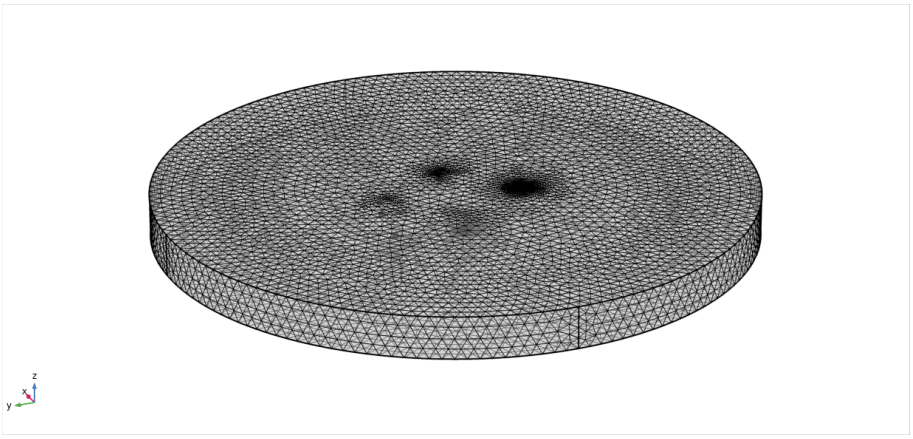

Figure 6. Comsol Multiphysics mesh plot of the PDMS mould model showing the automatically generated free tetdrahedral mesh

Two different kinds of simulation were made: steady-state and time-dependent. In the steady-state simulations, the temperature of the hot bath was fixed at a value of T_H and the air temperature fixed at a value of T_{air}. Equations (2) (12) and (13) were used in their time-independent forms where the terms including time or

time derivative are given the value of 0. Several steady-state simulations were made with different values of T_H. The values of T_H, T_{air} and ϕ_{air} used in the simulations are given in table 2, where ϕ_{air} is the relative humidity of the air. Figure 7 shows an isothermal contours plot of the simulated temperature across the mould. The steady-state simulations were run with the aid of Jarno Petäjä.

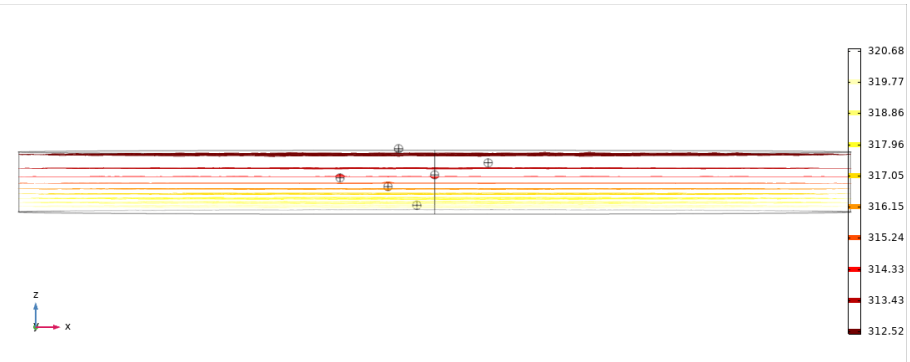

Figure 7. Comsol Multiphysics isosurface plot of the temperature across the PDMS mould in simulation 4, at 15 minutes.

In order to have the hot plate temperature be as accurate as possible in the time-dependent simulation, a least-squares fit to the measured data of NTC 8 was made for each time-dependent measurement. Because the heating has several distinct phases the data was separated into sections: before heater was turned on, during the heating phase, after heater reached the set temperature, and after the heater is turned off. Each section was fitted separately. For the first two sections, 3rd-degree polynomial fits were used and for the third section a power function

$$y = a \cdot x^b + c, \tag{46}$$

was used. The final section was only measured in measurements 1, 2 and 3 due to time constraints. The equation used for it was the second-degree exponential function

$$y = a \cdot e^{b \cdot x} + c \cdot e^{d \cdot x}. \tag{47}$$

The fits were done in MATLAB:s curve fitting tool, using the default values of all of the fitting settings. The coefficients calculated by the least-squares fit were then used to define a MATLAB function that returned the hot plate temperature as a function of time.

In order to reduce simulation time and memory needed for the simulation and

results, the results of the time-dependent solver was set to generate values for the simulation once per minute. To improve accuracy, the time-stepping of the solver was set to 'intermediate' which mandates the solver to calculate the values of the dependent variables at least once every time step. [26] Equations (2), (12) and (13) were used in their time-dependent forms.

The results of the simulations are comprised of the volume averages of the temperatures of the NTC glass heads calculated in the Comsol software and extracted to MATLAB. For the time-dependent case, the temperature averages are evaluated every minute. The results of the simulations are discussed and compared to the experimental measurements in section 5.2.

4.3 Microfluidic channel simulations

The results of the verification measurements and simulations showed good agreement as discussed in section 5.4. Therefore, the simulation tools used in section 4.2 can reasonably be expected to produce useful results. New simulations were made with a geometry that includes microfluidic channels. These channels were modelled according to designs used in microfluidic nucleic acid amplification reactions [1]. As the operating temperature of nucleic acid amplification methods such as the polymerace chain reaction (PCR) [3] and LAMP [2] needs to be accurately controlled, the aim of these simulations is to provide a starting point for heater design and temperature control.

PCR and LAMP require different operating temperatures. PCR reactions require cycling between different temperatures at a range of 323.15 to 363.15K. [3] Efficient heating and cooling can reduce the time it takes to go between the temperature steps, shortening the time needed for the amplification process. In LAMP, the reaction is isothermal [2] and the heating only needs to quickly reach the operating temperature and maintain it during the reaction.

The model includes two layers of PDMS: a bottom layer in which the microfluidic channels have been printed and a thicker lid enclosing the channels inside PDMS. There is also a polyethene terephthalate (PET) substrate as the bottom layer of the model. The PET material properties used in the simulation are shown in table 3. As the producers material data sheet did not include thermal conductivity or specific heat capacity [33] values measured for the material by Lopes and Felisberti [32] were used instead. In order to calculate the density of the PET used, the area

yield value for the 125μm film was multiplied by the layer thickness of 125μm and the result was inverted to arrive at the value shown in table 3. The thickness of the PET substrate is 125μm, the PDMS bottom layer 300μm and the thickness of the lid 5mm. The microfluidic channels are 100μm high and 400 μm wide and connected to four microfluidic chambers that are 2mm in diameter and their centers are 5mm apart from each other. The Comsol Multiphysics geometry of the model is shown in figure 8.

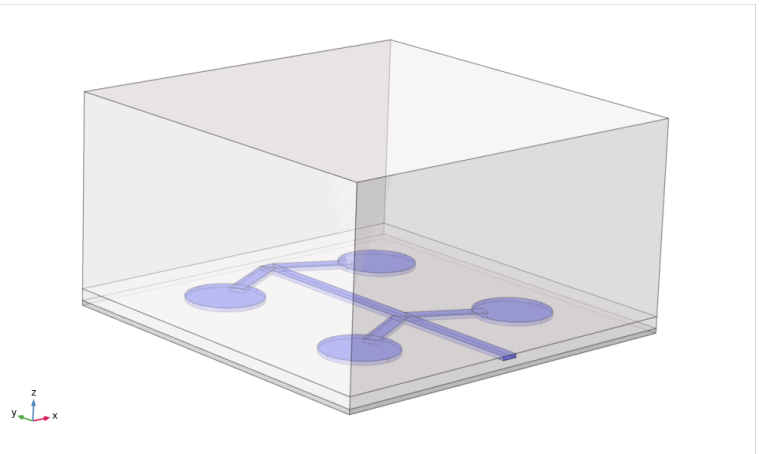

Figure 8. The Comsol Multiphysics geometry of the PDMS microfluidic platform. The PDMS (white, translucent) is divided into the thin base and the thicker lid. the microfluidic channels and chambers (blue) are printed into the top of the PDMS base. A thin PET wafer (gray) is the bottom layer of the model.

Adding the microfluidic channels necessitates modelling fluid flow in the channels. In order to keep the simulations as simple as possible the creeping (stokes) flow interface was used in the fluid flow modelling. Water was used as the fluid, as the liquids used in the microfluidic channels are mostly water. The Comsol Multiphysics default water properties were used. [26] A point at the end of the channel was chosen as the constant pressure point needed in order for the simulation to converge. [26]

As in section 4.2 the bottom of the PDMS was set to a constant temperature boundary condition and the top to a horizontal hot plate and naturally convecting moist air boundary condition. The room air temperature was set to 298.15K and the moisture to 50%. The end of the microfluidic channel was set to the open boundary condition, allowing fluid flow in and out of the model area. The initial speed of the fluid was set to 0 and the initial pressure to 1 atm. The pressure of the fluid outside the open boundary condition was set to 1 atm. Initial temperature for the entire

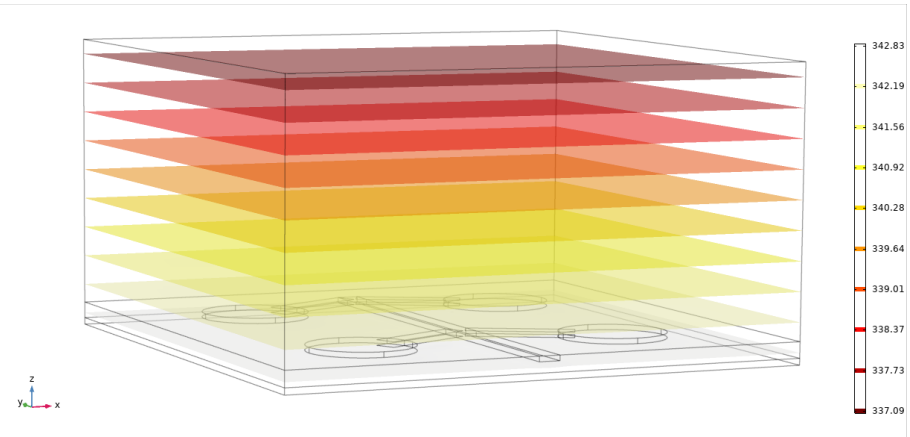

Figure 9. Simulated isotherms in the steady-state simulation with a 363.15K heating at the bottom of the model

model was set to 298.15K. Coupling the fluid flow and heat transfer interfaces was done using the nonisothermal flow multiphysics coupling. The mesh was done by Comsol Multiphysics with a "normal" mesh size set for the PDMS and PET and a "finer" mesh size for the microfluidic channel.

Again, both steady-state and time-dependent simulations were made. For the steady-state simulations, equations (2), (6), (7), (8), (12) and (13) were used in their time independent forms. A comsol multiphysics isothermal contours plot of the steady-state measurement in shown in figure 9. The temperature of the bottom of the model was set to 323.15K, 343.15K and 363.15K. In the time-dependent simulations the change in the temperature of the bottom was considered to happen instantaneously. The same temperatures of 323.15K, 343.15K and 363.15K were used. 180 seconds were simulated, with a simulation step of 5 seconds. The results consist of the volume averages of the temperatures on the microfluidic chambers and the surface average of the top of the model.

A test for mesh convergence was also done with a tighter mesh with the mesh size "extra fine" for the microfluidic channel and "fine" for everything else. These results are discussed in section 5.5. In the time-dependent simulations, the results are recorded between time steps every five seconds. From the time dependent measurements the time needed to reach the steady-state temperature in the chambers is of interest. However the results of the verification showed in section 5.4 showed that the simulation is not completely accurate in this regard and a longer time can be expected than what the simulation shows.

4.4 The bisection method with Comsol Multiphysics and MATLAB

One of the most reliable root-finding algorithms is the bisection method. It finds a root of a function f that is continuous in the interval [a,b], where $f(a)f(b) < 0$, and therefore $f(a)$ and $f(b)$ have different signs. [34, 35] As f is continuous and $f(a)$ and $f(b)$ have different signs, and therefore there must be at least one root between a and b due to the intermediate value theorem. [34]

To find one of the roots of f, the bisection method divides the interval in two, placing the midpoint c as [34, 35]

$$c = a + \frac{1}{2}(b - a). \tag{48}$$

Now, if $f(a)f(c) < 0$ at least one root exists in the interval [a,c]. If $f(a)f(c) > 0$ at least one root exists in the interval [c,b]. Now this second interval can be bisected as the second iteration of the bisection method algorithm. [34, 35]

If $[a_{n-1}, b_{n-1}]$ is the interval after $n-1$ iterations, $x_n = c_n$ is the midpoint of that interval and $\alpha \in [a,b]$ is a root of f, $|x_n - \alpha| \leq \epsilon$ if [34]

$$\left(\frac{1}{2}\right)^n (a_0 - b_0) \leq \epsilon, \tag{49}$$

where ϵ is the tolerance for error. The number of iterations needed to reach this accuracy is [34]

$$n = \frac{log(b-a) - log(\epsilon)}{log(2)}. \tag{50}$$

If a variable in a Comsol Multiphysics model affects a result of a steady-state simulation the bisection method can be used in order to look for an optimal value of the variable. The bisection method has been used with comsol multiphysics to look for optimal simulation variables in several studies [36–38].

In the context of designing heaters for a specific reaction in a microfluidic platform the operating temperature of the reaction can be thought of as optimal. Therefore function f is

$$f(x) = T_s(x) - T_O, \tag{51}$$

where T_O is the operating temperature of the reaction and $T_s(x)$ is the simulated temperature in the fluid. In this case, x is the variable that is changed in the

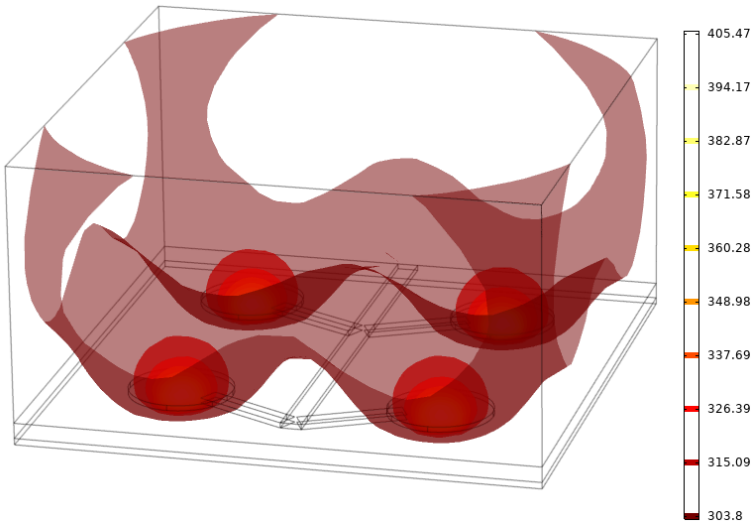

Figure 10. The Comsol Multiphysics isothermal contours plot of the power heating simulation, with the power of each point heater being 0.24804W.

simulation to find the value closest to the physical case. In this work this method was used to find the power needed to heat up a microfluidic channel to a set temperature. These simulations are discussed in section 4.5.

4.5 Microfluidic channel power heating simulations

In addition to the simulations presented in section 4.3 the same microfluidic chip geometry shown in figure 8 was used in a simulation of point heating. In this simulation the top of the model was set to the same natural convection boundary condition than in section 4.3. Below each of the four round microfluidic chambers, a point on the bottom surface of the model was set as a point heat source. The simulation did not converge without a set temperature somewhere in the model. Therefore the bottom surface of the model was set to 298.15K to simulate the chip laying on a table with a very large heat capacity.

In this model the heating is not done by a hot plate that is much larger than the sample itself, and therefore the effect of the sides of the model should be taken into account. These sides are given the Comsol Multiphysics default naturally convection from a vertical hot plate heat flux boundary condition that is similar to the natural convection from a horizontal hot plate boundary condition used on the top of the model. Each point heater is set to the same power. The point heaters are also given a radius of the diameter of the microfluidic chamber divided by 4. In the simulation,

the heat flux in the model formed spherical isotherms around the point heat sources, shown in figure 10.

After initial tests to see that the power values needed to heat the average temperature of the fluid in the microfluidic channels to 323.15K, 343.15K and 363.15K showed that the power needed was somewhere between 0.1W and 1W. The bisection method outlined in section 4.4 was used to find the values of power with higher accuracy. The function with which power P affects the average temperature of the fluid, $T_f(P)$, is unknown but because it is a physical function it can be assumed to be continuous. It is also evident that a higher heater power should result in a higher temperature. For each fluid operating temperature function (51) was used as the function for which the root needs to be found.

The error tolerance P_ϵ was set to 0.0001W. Using equation (50) the number of iterations needed to reach this accuracy was found to be 14. As a higher heating power means a higher temperature of the fluid, the condition to find on which side of the midpoint was simplified to if

$$T_O < T_s(P_{c_i}), \tag{52}$$

then

$$P_{b_{i+1}} = P_{c_i}, \tag{53}$$

else

$$P_{a_{i+1}} = P_{c_i}, \tag{54}$$

Where P_{c_i} is the midpoint of $[P_{a_i}, P_{b_i}]$. The algorithm written in MATLAB was also given a maximum number of iterations of 20 to end the simulation if there was an error in the code and to not cause infinite loops. The results of this simulation are discussed in section 5.6.

5 Results

5.1 NTC calibration results

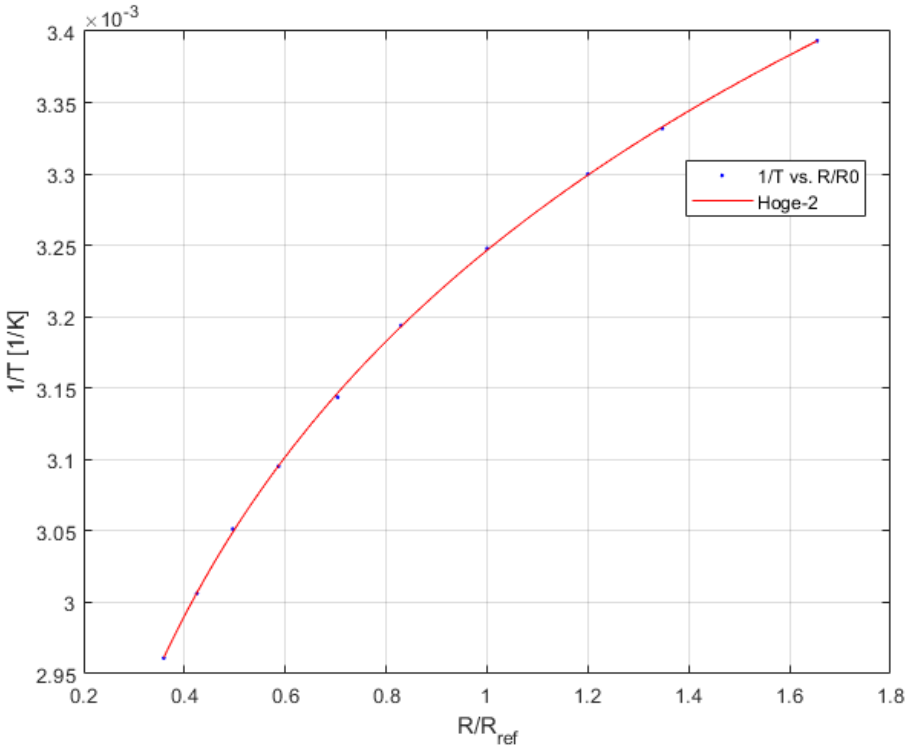

Figure 11. Hoge 2 fit for NTC 1. The dots signify the measured calibration points while the line shows the Hoge 2 equation fit to the calibration points.

The Hoge-2 function (23) used to calibrate the NTCs and the error propagation in calibration are discussed in section 3.3. In order to reduce the effect of the random fluctuations of the reading, each calibration point was the average of the set of measured voltages at the corresponding calibration temperature. The errors for the averages were calculated with equation (41). The calibration points for NTCs 1-7 were measured at the same time and the calibration points for NTC 8 were measured separately from the others. Therefore NTC 8 has different calibration temperatures compared to the other NTCs. Ten calibration points were measured for all of the NTCs.

Table 4. Calibration coefficients for each calibrated NTC rounded to 5 decimal places.

NTC	A_1	A_2	A_3	A_4
1	3.24662e-3	2.87360e-4	7.80148e-6	-8.18540e-7
2	3.24699e-3	2.85111e-4	6.50138e-6	-9.29824e-7
3	3.24720e-3	2.85277e-4	7.60065e-6	3.37398e-7
4	3.24804e-3	2.85482e-4	9.03174e-6	3.27219e-6
5	3.24716e-3	2.83962e-4	5.05444e-6	-1.58852e-6
6	3.24743e-3	2.84961e-4	7.01707e-6	3.76925e-7
7	3.25007e-3	2.85698e-4	1.68327e-5	1.46013e-5
8	3.23083e-3	2.83412e-4	5.28301e-6	1.73681e-6

The Hoge 2 fit generated by MATLAB:s fitting tool for one of the NTCs is shown in figure 11. In order to generate the fit according to equation (23) one of the calibration points, in this case the third point that was at the temperature of 307.90K was chosen as the reference point for each the first seven NTCs. NTC 8 was calibrated separately but the third point was also selected as the reference point for it. Table 4 shows the calibration coefficients for each NTC.

The measurement errors for each used device were read from their manuals [11, 23, 24]. For the calibration temperatures, each measurement also had a reading inaccuracy as the thermocouple reading fluctuated slightly during the measurement and was recorded by hand. The temperature of the oven was never completely static, instead cyclically fluctuating. Averaging was used in order to compensate for the temperature fluctuation.

Figure 12 shows the error of the calibration for each NTC calculated according to the method discussed in section 2.4.3. These errors stay under 1.4K in the calibration area, but increase quickly outside of it. The calibration area is marked in figure 12 with dashed vertical lines. The calibration was determined to be accurate enough for the verification measurements.

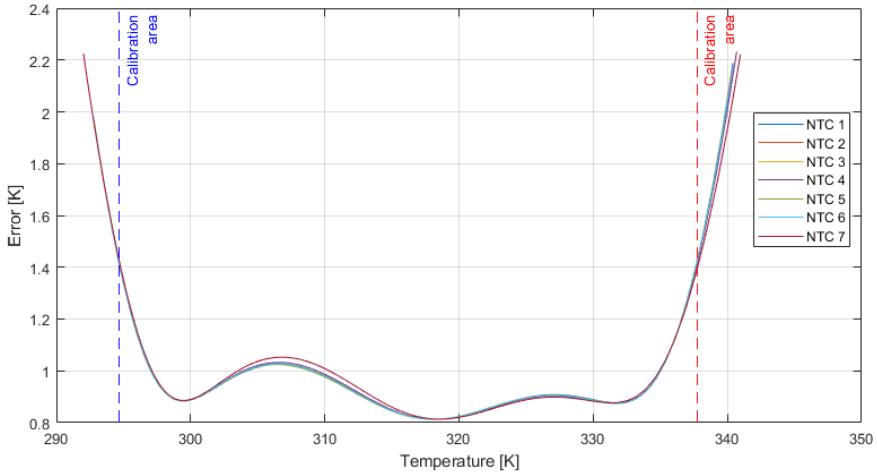

Figure 12. The error curves of the calibrated NTCs. The dashed vertical lines denote the area between the coldest and hottest calibration point.

5.2 Verification measurement results

In order to reduce the effect that the cycling of the hot plate near the set temperature has on the readings of the NTCs and to reduce the effect of noise the readings are averaged over a period of 30s for measurements 8a, 8b and 8c and 50s for measurements 7a, 7b and 7c. As LabView was set to record a data point every second and average 10 readings for each data point, 300 or 500 readings were averaged. Because the errors of the calibration were known a weighted average was calculated with equation (42) The error of the average was calculated with equation (43). The averaged results in the steady-state measurements are shown in table 5.

The results of the steady-state measurements are plotted with the corresponding simulation results in figure 14. The measured and simulated data sets are compared in section 5.4. The hot plate temperatures measured with NTC 8 and the air temperatures and humidities are shown in table 6.

In the time-dependent measurements 1-6 the same LabView settings were used as in the steady-state measurements. Therefore a data point is recorded every second, with 10 values being averaged to each data point. However, sometimes LabView did not record a data point, leaving a one-second gap in the recorded data. This was mitigated by having LabView also record a time stamp for each data point and plotting according to the time stamp and not the index of the temperature values. Figure 13 shows the data from measurement 4, where the hot plate was set

Table 5. Steady-state measurement results in the verification measurements. The unit of the measured results is Kelvin.

	NTC 1	NTC 2	NTC 3	NTC 6	NTC 4	NTC 7
7a	318.18± 0.12	317.38± 0.12	316.96± 0.12	316.75± 0.12	316.24± 0.12	315.00± 0.13
7b	327.94± 0.13	326.71± 0.13	326.37± 0.13	325.63± 0.13	324.49± 0.13	323.80± 0.13
7c	337.5± 0.2	335.9± 0.2	335.58± 0.15	334.55± 0.14	333.13± 0.13	332.37± 0.13
8a	313.2± 0.2	312.7± 0.2	312.3± 0.2	312.4± 0.2	311.9± 0.2	310.5± 0.2
8b	323.13± 0.13	322.24± 0.13	321.66± 0.12	321.55± 0.12	320.94± 0.12	319.21± 0.12
8c	332.79± 0.13	331.54± 0.13	330.84± 0.13	330.56± 0.13	329.74± 0.13	327.41± 0.13

Table 6. Steady-state hot plate temperatures measured with NTC 8. The hot plate temperature values are weighted averages rounded and shown with their errors from the raw data points given in the data points column. T_{air} has the measured air temperature values during the measurement and ϕ_{air} has the measured relative humidity values.

	T_H [K]	T_{air}[^{o}C]	ϕ	Data points
7a	318.29 ± 0.12	23.1-23.3	46%	2080-2130
7b	328.33 ± 0.13	23.4-23.5	46%-47%	3910-3960
7c	338.37 ± 0.13	23.6-23.8	47%-48%	7590-7640
8a	313.3 ± 0.2	23	48%	1950-1980
8b	323.37 ± 0.11	23	48%	3700-3750
8c	333.38 ± 0.14	23	49%	6020-6070

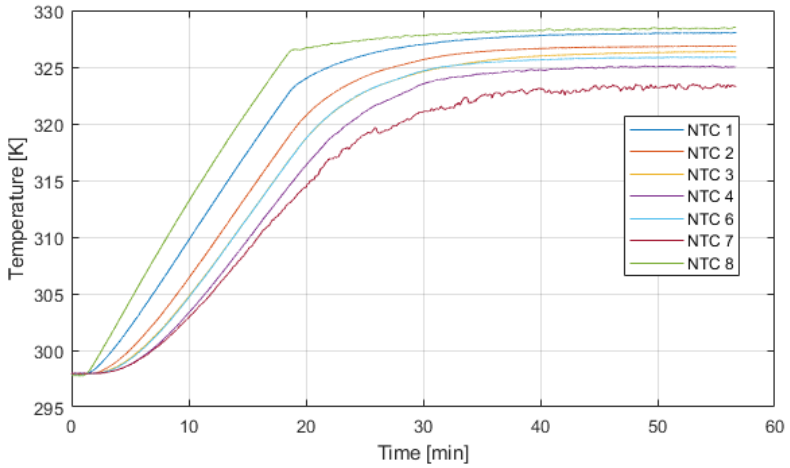

Figure 13. Temperatures measured with the NTCs embedded into the PDMS mould, measurement 4.

to 328.15K.

Measurements 1-3 also measured a cooling period after the heater was turned off, while measurements 4-6 measured only the heating to steady-state. During the cooling period as the temperature of the PDMS became closer to the air temperature, the cooling became slower. The temperature of the heater influenced the temperature of the mould during the cooling, and the temperature gradient over the mould diminished quickly. Figure 16 shows the results of measurement 2 with the simulated values.

5.3 PDMS verification simulation results

The results of the Comsol Multiphysics simulations are the volume average temperatures of the simulated glass NTC heads. In the steady-state simulations, a single value was extracted from each NTC. These values are shown in table 7 and plotted with corresponding measured values in figure 14. For the time-dependent simulation one set of temperature values was generated for every minute of the simulation.

In both the steady-state and time-dependent cases, the temperature gradient across the simulation formed almost flat isotherms. In figure 7 these isotherms are shown in a Comsol Multiphysics isosurface plot of the sample during the heating phase of simulation 4.

A set of steady-state simulations with a denser mesh were also made. This increased simulation time significantly but the results as shown in table 8 change less

Table 7. Simulated steady-state results with the verification model. The units for the values are Kelvin. Each value is the volume average of the temperature of the NTCs in the simulation model. The values are rounded to two decimal places.

	NTC 1	NTC 2	NTC 3	NTC 6	NTC 4	NTC 7
7a	318.02	317.24	316.86	316.74	316.24	315.50
7b	327.92	326.68	326.09	325.90	325.11	323.96
7c	337.79	336.06	335.24	334.97	333.88	332.27
8a	313.13	312.55	312.27	312.18	311.80	311.26
7b	323.02	322.00	321.51	321.34	320.69	319.74
7c	332.88	331.37	330.65	330.41	329.45	328.05

Table 8. The difference between the simulated values with a 0.5 times smaller mesh to the ordinary simulated results. Calculated as simulated value with normal mesh - value with smaller mesh. The unit for each value is Kelvin. The values are rounded to two significant digits.

	NTC 1	NTC 2	NTC 3	NTC 6	NTC 4	NTC 7
7a	-0.00030	0.00064	0.00054	0.00032	-0.00062	-0.0025
7c	-0.0050	0.00012	0.0057	0.0070	0.015	0.025
8b	-0.00058	0.00085	0.00089	0.00065	-0.00036	-0.0024

than 25mK for each datapoint. As the inaccuracy of the measured values is much higher the less dense mesh was used in order to save computation time.

5.4 Comparison between measured and simulated values

The steady-state simulated and measured results are shown in the same plot in figure 14. For each hot plate temperature, the measured and simulated results are very close to each other though the difference increases near the top of the mould. The difference between the measured and simulated values are shown in table 9.

As discussed in section 4.2 the air flow around NTC 7 was not fully simulated but instead approximated with the same boundary condition as the top of the PDMS mould. Therefore, there is unknown air flow around this NTC. It is reasonable that the simulation would be more inaccurate than with the NTCs fully encased in the mould where the air flow does not directly affect them.

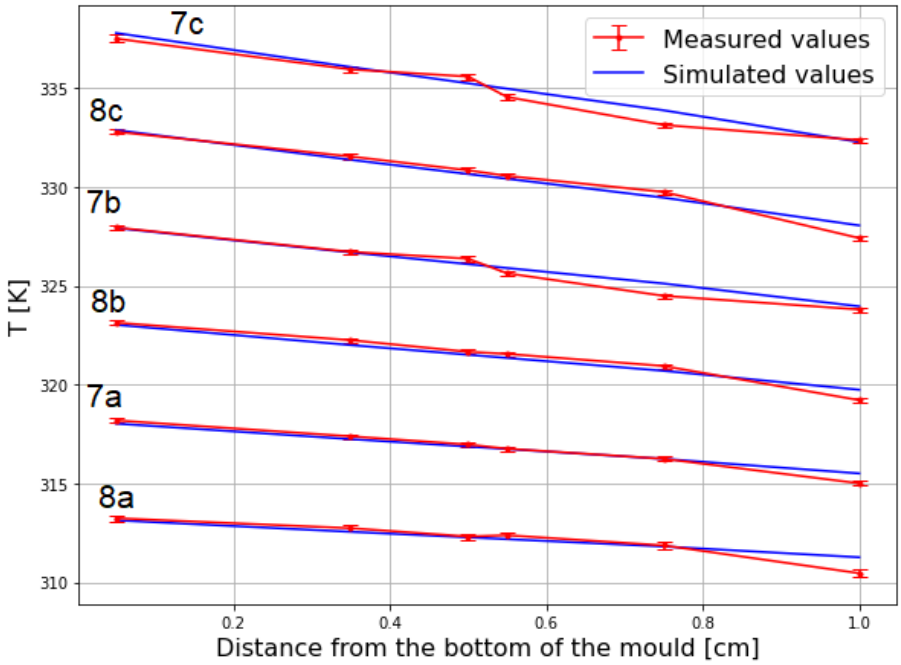

Figure 14. Measured and simulated results for the verification measurements. Each pair of lines represents a different measurement and simulation with the hot plate temperatures shown in table 6.

Table 9 and figure 14 show that in steady-state the simulated and measured values are in close agreement with six different hot plate temperatures. For all NTCs and measurements the values are within 0.81K of eachother. Most of the values are within 0.2K of each other and many of the simulation values fall within the error bounds of the measured values shown in table 2. The differences seem to increase towards the top of the PDMS.

The results of measurement and simulation 4 are shown in figure 15. Similarly to the steady-state simulations, the end of the simulation closely matches the measurement results. However, during the heating phase the simulated results rise much faster than the measured ones as a result of the heater temperature rising.

Figure 16 shows the results of measurement and simulation 2. These included a cooling period during which the PDMS mould cooled steadily after the heating had been turned off. The result is similar to the heating measurements as the measured values lag behind from the simulated values.

A possible reason for this is the heat capacity of the PDMS. As discussed in section 4.2 the heat capacity in the simulation was the default heat capacity for PDMS in the Comsol Multiphysics material library [26] because the manufacturer

Table 9. Measured temperature - Simulated temperature in the steady-state verification measurements. Each value is rounded to 2 decimal places. The unit of each value is kelvin.

	NTC 1	NTC 2	NTC 3	NTC 6	NTC 4	NTC 7
7a	0.16	0.14	0.10	0.01	0.005	-0.50
7b	0.02	0.03	0.28	-0.27	-0.62	-0.16
7c	-0.28	-0.12	0.34	-0.42	-0.74	0.10
8a	0.10	0.18	0.04	0.20	0.05	-0.81
8b	0.11	0.25	0.15	0.20	0.24	-0.53
8c	-0.09	0.17	0.19	0.14	0.29	-0.64

of the PDMS did not include a value in their data sheet. [21] The accuracy of the steady-state simulation is not affected by the thermal capacity of the PDMS as the time-independent form of equation (2) does not include it. Increasing the thermal capacity of the simulated PDMS could slow the conduction of heat through the PDMS as shown in equation 3.

For the purposes of this work, the time-dependent simulation also produces results that are close to the measured results. However, when using this simulation method in subsequent work, it is important to take into account that heat propagates through the PDMS slower than the simulation would predict and the steady state is achieved later.

5.5 Microfluidic simulation results with hot plate heating

After the verification process, a set of simulations of a microfluidic PDMS chip were made. The simulation settings and methods are discussed in section 4.3. Steady-state and time-dependent simulations were made. The results consist of the average temperature of fluid in the microfluidic channel, the temperature of the PDMS and the average temperature of the surface at the top of the model. The average and maximum magnitudes of the velocity of the fluid in the microfluidic channels was also evaluated.

The results of the steady-state simulations are shown in table 10. Figure 9 shows the isotherms over the simulated model in the steady-state heating with 363.15K. Table 10 shows a large increase in the thermal gradient over the model when the temperature at the bottom increases. Importantly, the temperature of the

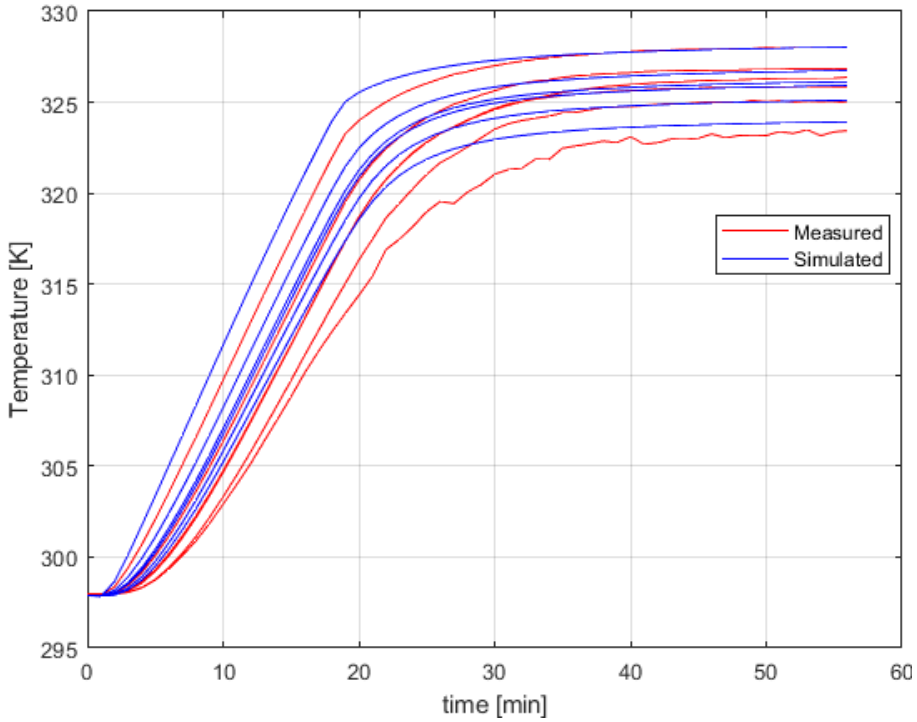

Figure 15. Measured and simulated temperatures, measurement 4 and simulation 4.

microfluidic channel differs from the temperature at the bottom by 0.2363K with T_H=323.15K while hot plate temperature of T_H=363.15K increases this difference to 0.7704K. The total thermal gradient over the model also increased from 3.0951K to 10.1043K.

Tests for mesh convergence with the microfluidic simulation model were also made. With the Comsol Multiphysics predefined mesh size of extra fine for the microfluidic channels and fine for the rest of the model, the simulation time tripled from around one minute to around three. The change in the results of the simulation is shown in table 11. The change in temperatures between the mesh is negligible as it is at least two orders of magnitude smaller than the difference between the hot plate and fluid temperatures shown in table 10. While the tighter mesh seems to allow slightly faster fluid movement the difference is at least an order of magnitude smaller than the fluid velocity itself and does not seem to affect the temperatures that are the focus of the simulation. Therefore the wider mesh can be used in the simulations.

In the time dependent heating simulations, the temperature changes from the initial value of 25^{o}C towards the temperature gradient of the steady-state. As

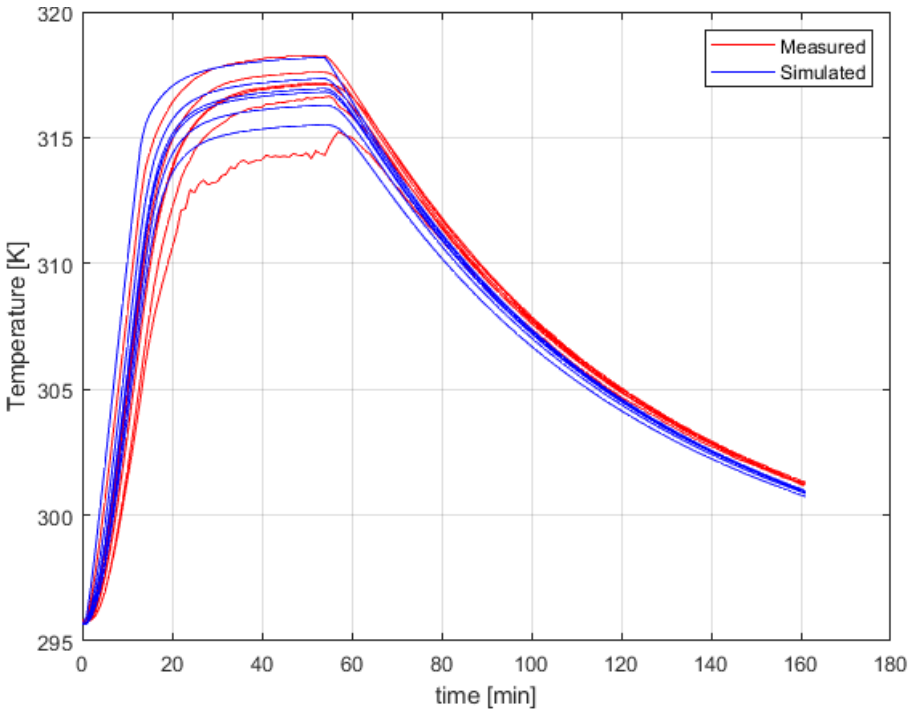

Figure 16. Measured and simulated temperatures, measurement 2.

discussed in section 5.4, these simulations should be considered with care, as the simulation results reached the steady-state faster than the measured results in the verification process. The simulation can still be expected to give a rough estimate of the time required to reach the steady-state. Figure 17 shows the simulated temperatures for three different hot plate temperatures.

Figure 17 shows that for every temperature, the average temperature of the fluid rises very quickly while the temperature of the top of the chip is much slower to

Table 10. Steady-state simulation results with the microfluidic chip model. T_F is the average temperature of the fluid in the channels and T_{top} is the average temperature of the top of the model. The average and maximum magnitudes of velocity of the fluid in the channels is also shown. The unit of the temperatures and their differences is kelvin.

T_H	T_F	T_{top}	$T_H - T_F$	$T_H - T_{top}$	$\overline{\lvert v \rvert}$ [m/s]	$\lvert v \rvert_{max}$ [m/s]
323.15	322.9137	320.0549	0.2363	3.0951	$0.1804 \cdot 10^{-7}$	$0.3344 \cdot 10^{-4}$
343.15	342.6636	336.7743	0.4864	6.3757	$0.4847 \cdot 10^{-7}$	$0.8936 \cdot 10^{-4}$
363.15	362.3794	353.0457	0.7706	10.1043	$1.0072 \cdot 10^{-7}$	$1.8504 \cdot 10^{-4}$

Table 11. The effect of a tighter mesh size to the results in Steady-state simulation with the microfluidic chip model. The values are calculated as normal mesh - tighter mesh. The values for normal mesh are shown in table 10

T_H [K]	T_F [K]	T_{top} [K]	$\overline{\lvert v \rvert}$ [m/s]	$\lvert v \rvert_{max}$ [m/s]
323.1500	$0.0889 \cdot 10^{-3}$	$0.0295 \cdot 10^{-3}$	$-1.3094 \cdot 10^{-9}$	$-0.4036 \cdot 10^{-5}$
343.1500	$0.1906 \cdot 10^{-3}$	$0.0627 \cdot 10^{-3}$	$-3.4760 \cdot 10^{-9}$	$-1.0786 \cdot 10^{-5}$
363.1500	$0.3102 \cdot 10^{-3}$	$0.1008 \cdot 10^{-3}$	$-7.1681 \cdot 10^{-9}$	$-2.2334 \cdot 10^{-5}$

Table 12. The results of the point heat source simulations for the microfluidic platform model. T_O is the optimum temperature, P is the heating power setting found with the optimization algorithm and T_s is the temperature simulated with the heating power.

T_O [K]	P [W]	T_s [K]	$T_s - T_O$ [K]	iterations
323.15	0.2480	323.1464	-0.0036	14
343.15	0.4476	343.1487	-0.0013	14
363.15	0.6474	363.1446	-0.0054	14

respond. In the simulated time of three minutes the temperatures reach values very close to the steady-state values in table 10.

5.6 Microfluidic simulations with point heat sources

The results of the simulation with the point heaters are shown in table 12. Each bisection took the predicted 14 iterations and achieved convergence with no issues. As the simulation time for a single simulation was approximately 1 minute, each run with the bisection algorithm took approximately 15 minutes.

The accuracy of finding a power suitable for each temperature is also shown in table 12. For each temperature, a power was found where the temperature was well within one degree of the operating temperature.

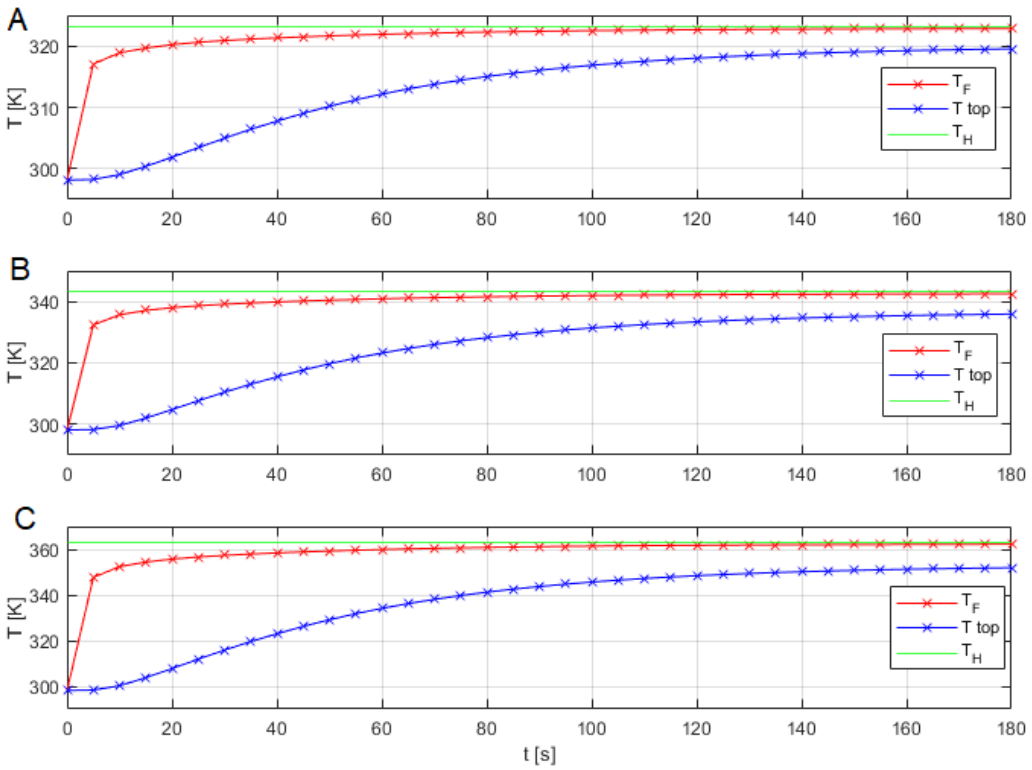

Figure 17. Simulated hot plate heatings of the microfluidic chip model from 298.15K to 323.15K (A), 343.15K (B) and 363.15K. (C)

6 Discussion

In this work, finite element method (FEM) simulations were used in order to predict the thermal transport properties of microfluidic platforms made of polydimethylsiloxane (PDMS) during different heating processes. The validity of the simulation method was verified by comparing the results of a FEM simulation to experiments with a PDMS mould where the thermal gradient and transient thermal transport properties were measured with negative temperature coefficient thermistors (NTCs). The NTCs were used to measure the temperature in the PDMS mould at different distances from the heater, establishing a measured thermal gradient across the mould. A FEM simulation model was made to match the experiment as closely as possible.

Measurements and FEM simulations in both the steady-state and time-dependent heating cases were made with the verification mould. The steady-state measurement and simulation results were similar as seen in figure 14 and table 9. The time-dependent simulations and measurements were close to each other, but it became clear that the changes in temperature occur faster in the simulation than in the experimental sample. Figures 15 and 16 show this behaviour both during heating and cooling. The steady states reached at the ends of these simulations again showed an agreement between the simulated and experimental cases.

There are multiple factors that can affect the rate of heat transfer in the model. Thermal diffusivity, discussed in section 2.1 is affected by the thermal conductivity, density and specific heat capacity of the material. In the manufacturer data sheet of the PDMS used in this work, Sylgard 184, [21] there was no mention of the heat capacity of the material and the density was given as the specific gravity with no mention how that specific gravity was calculated. Therefore I used an another source for the heat capacity of the PDMS and solved the density using water at 25^{o}C as the reference material. This solution is functional but not does not appear to be exact. The source used for the heat capacity [26, 31] did not specify the manufacturer of the PDMS. There was a discrepancy in the thermal conductivity values between the source (0.19) [26, 31] and the Sylgard 184 manufacturer datasheet (0.27) [21]. It is therefore possible the heat capacity of Sylgard 184 differs from the value used in the

simulation.

Other factors that change the rate of heat transfer in the FEM simulation model include the thermal contact resistance between the heater and the mould and the characteristics of the air flow in the vicinity of the mould. Altering these factors would also have changed the steady-state simulation. Because the results of the steady state measurements and simulations were very close to each other, it seems these factors are accurate in the FEM simulation and the material properties are the reason for the disparity in the time-dependent case. In the future, calorimetric measurements on the PDMS could be used to obtain more accurate material properties of the Sylgard 184 PDMS used.

FEM simulations using the same boundary conditions and material properties as in the verification process were made on the PDMS microfluidic chip design. In addition, simulations with point heat sources underneath the microfluidic channels were made where the heating power of the point heat source was found using the bisection method. These simulations included fluid flow in the microfluidic channels and were intended to aid in designing heating methods for nucleic acid amplification reactions. Again, both steady-state and time-dependent simulations were made.

The results of the steady-state simulations show a large increase in the thermal gradient over the model when the temperature of the heater is increased. The time-dependent measurements give an indication about how long reaching the steady state would take. With the time-dependent FEM simulation the result of the verification should be considered. The experimental sample will likely take more time to reach the steady state than the simulated model as changes to the temperature of the sample were slower in the experimental sample than in the FEM simulation during the verification. This problem can be mitigated by first setting the heater temperature higher than the desired temperature, causing a larger gradient and therefore a higher thermal flux and a faster change of temperature.

In the samples with very small dimensions some of the boundary conditions used in the FEM simulation can be expected to be less accurate when compared to experimental results. For example in these simulations the sides of the model were set to an insulated boundary condition. In the verification measurements this was justifiable because the area of the top of the mould was much larger than the sides. In addition, the mould is large enough that the NTCs were far away from the sides. In the printed microfluidic samples that may not always be the case. In the

work of Hiltunen et al. [1], figure 2h shows a sample with microfluidic channels that are close to the sides of the sample. The effects caused by the border depend on the type of heater and how far away from the border the microfluidic channels are. Generally, the temperature in experiments is lowered because of these effects. As the required fluid temperature increases, so too should the compensation between the heater temperature and the required fluid temperature as the temperature gradient across the PDMS becomes larger.

Even with the limitations to the accuracy of the FEM simulation method it was shown in this study to be accurate enough to use in designing heater components for PDMS microfluidics. Optimization algorithms such as the bisection method can be used with the simulations to find optimal heater designs customized to the microfluidic platform. Verification of the FEM model with experimental measurements gives confidence to the validity of using the simulation in the design of samples.

7 Conclusion

Heat transport over polydimethylsiloxane microfluidic platforms was simulated using finite element method simulations in order to investigate two possible heater designs for the microfluidics. The use of the simulation method was validated by a set of experimental measurements on a PDMS sample and simulations on a model made of the sample. The temperature gradient across the sample was measured with negative temperature coefficient thermistors as it was heated from below with a hot plate. Both steady-state and time-dependent measurements were made.

The results of these experiments and simulations closely matched each other in the steady-state. At all the measured datapoints the results of the measurements and simulations were within 0.81K of each other. Most of the results were within 0.2K of each other. In the time-dependent results the temperature of the PDMS changed faster in the simulations than in the measurements. This disparity can be caused by inaccurate material properties of the PDMS in the simulation model.

After verifying the simulation results corresponded with the experimental results the same simulation method was used to simulate PDMS microfluidics. Two different possible heating systems were simulated. In the first model the microfluidics were heated from below with a hot plate and in the second model it was heated with point heat sources below the microfluidic chambers inside the PDMS. With the first model, steady-state temperature gradients and heating times were simulated. The second model was used with the bisection method optimization algorithm to find an optimal heating power with an error tolerance of 0.1 mW for the point heat sources to reach an optimal temperature within the microfluidic chambers. The results of the simulations can be used as a step in the process of designing heater systems for these microfluidics.

References

[1] J. Hiltunen et al. “Roll-to-roll fabrication of integrated PDMS–paper microfluidics for nucleic acid amplification”. In: *Lab on a chip* (2018). DOI: 10.1039/C8LC00269J.

[2] T. Notomi et al. “Loop-mediated isothermal amplification of DNA”. In: *Nucleic Acids Research* 28.12 (2000), e63. DOI: https://doi.org/10.1093/nar/28.12.e63.

[3] Y. Zhang and P. Ozdemir. “Microfluidic DNA amplification—A review”. In: *Analytica Chimica Acta* 638.2 (2009), pp. 115–125. DOI: https://doi.org/10.1016/j.aca.2009.02.038.

[4] H. Hoge. “Useful procedure in least squares, and tests of some equations for thermistors”. In: *Review of Scientific Instruments* 59.975 (1988). DOI: https://doi.org/10.1063/1.1139762.

[5] D. W. Hahn and M. N. Özisik. *Heat Conduction*. 3rd ed. John Wiley & Sons Inc., 2012.

[6] G. Ruocco. *Introduction to Transport Phenomena Modeling*. Springer international publishing, 2018. DOI: https://doi.org/10.1007/978-3-319-66822-2.

[7] A. Bejan. *Convection Heat Transfer*. 4th ed. John Wiley & Sons Inc., 2013.

[8] J. Tu, G. H. Yeoh, and C. Liu. *Computational Fluid Dynamics - A practical approach*. 3rd ed. Elsevier Science and Technology, 2018.

[9] M. Kutz. *Handbook of measurement in science and engineering*. 1st ed. John Wiley & Sons Inc., 2013.

[10] J. Fraden. *Handbook of modern sensors*. Springer international publishing, 2016. DOI: https://doi.org/10.1007/978-3-319-19303-8.

[11] *NTC thermistors for temperature measurement*. EPCOS AG, 2015. URL: https://th.mouser.com/datasheet/2/400/NTC_Glass_enc_sensors_G1541_coated-1220400.pdf.

[12] G. Liu et al. “Evaluation of different calibration equations for NTC thermistor applied to high-precision temperature measurement”. In: *Measurement* 120 (2018), pp. 21–27. DOI: https://doi.org/10.1016/j.measurement.2018.02.007.

[13] G. Liu et al. “Uncertainty propagation in the calibration equations for NTC thermistors”. In: *Metrologia* 55.3 (2018). DOI: https://doi.org/10.1088/1681-7575/aaba8e.

[14] C. Chen. “Evaluation of resistance-temperature calibration equations for NTC thermistors”. In: *Measurement* 42.7 (2009), pp. 1103–1111. DOI: https://doi.org/10.1016/j.measurement.2009.04.004.

[15] D. White. “Interpolation Errors in Thermistor Calibration Equations”. In: *International Journal of Thermophysics* 38.59 (2017). DOI: https://doi.org/10.1007/s10765-017-2194-x.

[16] P. Hansen, V. Pereyra, and G. Scherer. *Least squares data fitting with applications*. Johns Hopkins University press, 2012.

[17] H. Berendsen. *A Student's Guide to Data and Error Analysis*. Cambridge university press, 2011.

[18] J. R. Taylor. *An Introduction to Error Analysis*. 2nd ed. University Science books, 1997.

[19] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *Finite Element Method: Its Basis and Fundamentals*. 7th ed. Elsevier Science and Technology, 2013.

[20] G. R. Liu and S. S. Quek. *The Finite Element Method: A Practical Course*. 2nd ed. Elsevier Science and Technology, 2013.

[21] *Sylgard 184 Silicone elastomer technical data sheet*. Dow Chemical company, 2017. URL: https://consumer.dow.com/content/dam/dcc/documents/en-us/productdatasheet/11/11-31/11-3184-sylgard-184-elastomer.pdf?iframe=true.

[22] *CR0603/CR0805/CR1206 - Chip Resistors*. Bourns, 2019. URL: https://www.bourns.com/data/global/PDFs/CHPREZTR.pdf.

[23] *NI USB-6002 Specifications*. National Instruments, 2014. URL: https://www.ni.com/pdf/manuals/374371a.pdf.

[24] *NI USB-TC01 Specifications*. National Instruments, 2017. URL: https://www.ni.com/pdf/manuals/374918b.pdf.

[25] *LabView systems engineering software*. National Instruments, 2019. URL: https://www.ni.com/fi-fi/shop/labview.html.

[26] *Comsol Multiphysics Modeling Software*. Comsol Inc, 2019. URL: www.comsol.com.

[27] V. Gerlich, K. Sulovská, and M. Zálešák. "COMSOL Multiphysics validation as simulation software for heat transfer calculation in buildings: Building simulation software validation". In: *Measurement* 46.6 (2013), pp. 2003–2012. DOI: https://doi.org/10.1016/j.measurement.2013.02.020.

[28] B. M. Mihiretie et al. "Finite element modeling of the Hot Disc method". In: *International Journal of Heat and Mass Transfer* 115.B (2017), pp. 216–223. DOI: https://doi.org/10.1016/j.ijheatmasstransfer.2017.08.036.

[29] F. Rabeler and A. H. Feyissa. "Modelling the transport phenomena and texture changes of chicken breast meat during the roasting in a convective oven". In: *Journal of food engineering* 237 (2018), pp. 60–68. DOI: https://doi.org/10.1016/j.jfoodeng.2018.05.021.

[30] X.-F. Pang. *Water: Molecular Structure and Properties*. 1st ed. World Scientific Publishing, 2014.

[31] J. E. Mark. *The polymer data handbook*. 2nd ed. Oxford university press, 2009.

[32] C. Lopes and M. Felisberti. "Thermal conductivity of PET/(LDPE/AI) composites determined by MDSC". In: *Polymer Testing* 23.6 (2004), pp. 637–642. DOI: https://doi.org/10.1016/j.polymertesting.2004.01.013.

[33] *Melinex ST506 Data Sheet*. Dupont Teijin films, 2016. URL: http://usa.dupontteijinfilms.com/wp-content/uploads/2017/01/ST506-Datasheet.pdf.

[34] J. Epperson. *An Introduction to numerical methods and analysis*. 2nd ed. John Wiley & Sons Inc., 2013.

[35] G. M. Phillips and P. J. Taylor. *Theory and Applications of Numerical Analysis*. 2nd ed. Elsevier Science and Technology, 1996.

[36] M. Pavel, R. Tanasa, and A. Stancu. “Magnetic trap effects on nanowire’s dynamics within micro-capillary vessels”. In: *Microfluidics and Nanofluidics* 10.3 (2011), pp. 579–591. DOI: DOI:10.1007/s10404-010-0691-3.

[37] N. Couninot et al. “Capacitive biosensing of bacterial cells: Analytical model and numerical simulations”. In: *Sensors and Actuators B: Chemical* 211 (2015), pp. 428–438. DOI: https://doi.org/10.1016/j.snb.2015.01.108.

[38] W. Li et al. “Topological design of 3D phononic crystals for ultra-wide omni-directional bandgaps”. In: *Structural and Multidisciplinary Optimization* 60 (2019), pp. 2405–2415. DOI: https://doi.org/10.1007/s00158-019-02329-0.

A Experiment photographs

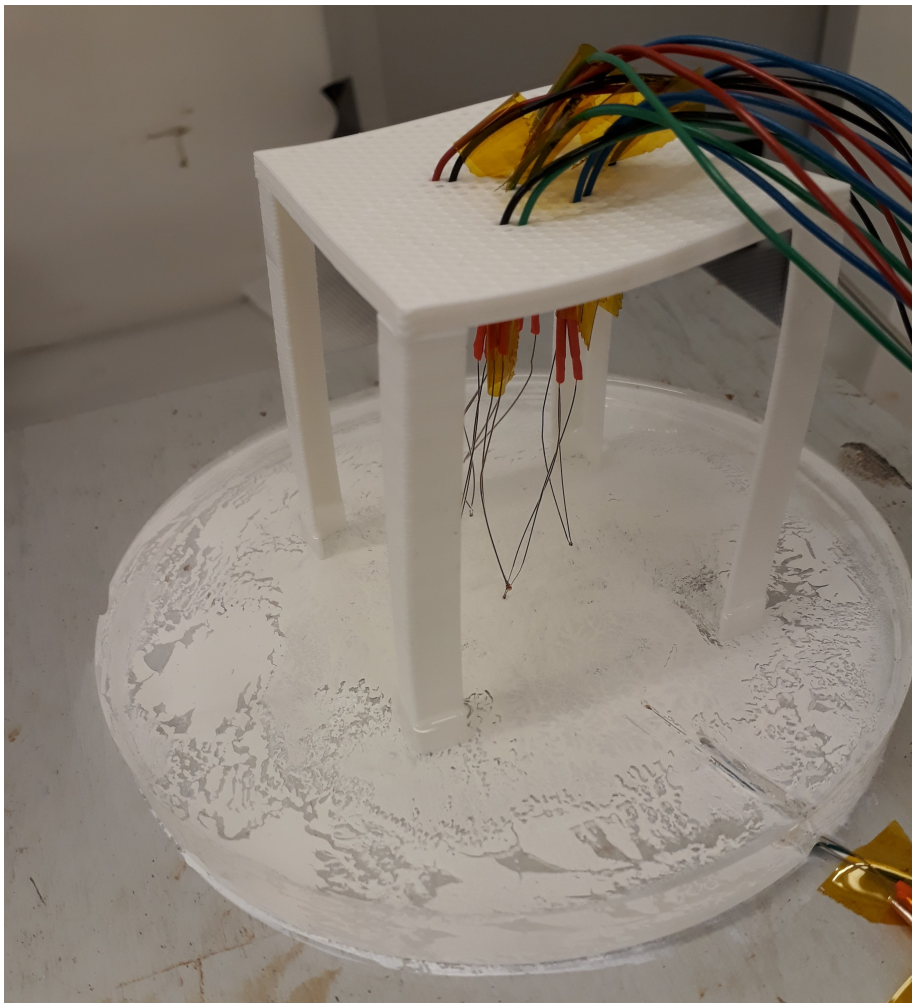

Figure 18. An image of the mould used in the verification measurements with thermal paste below it

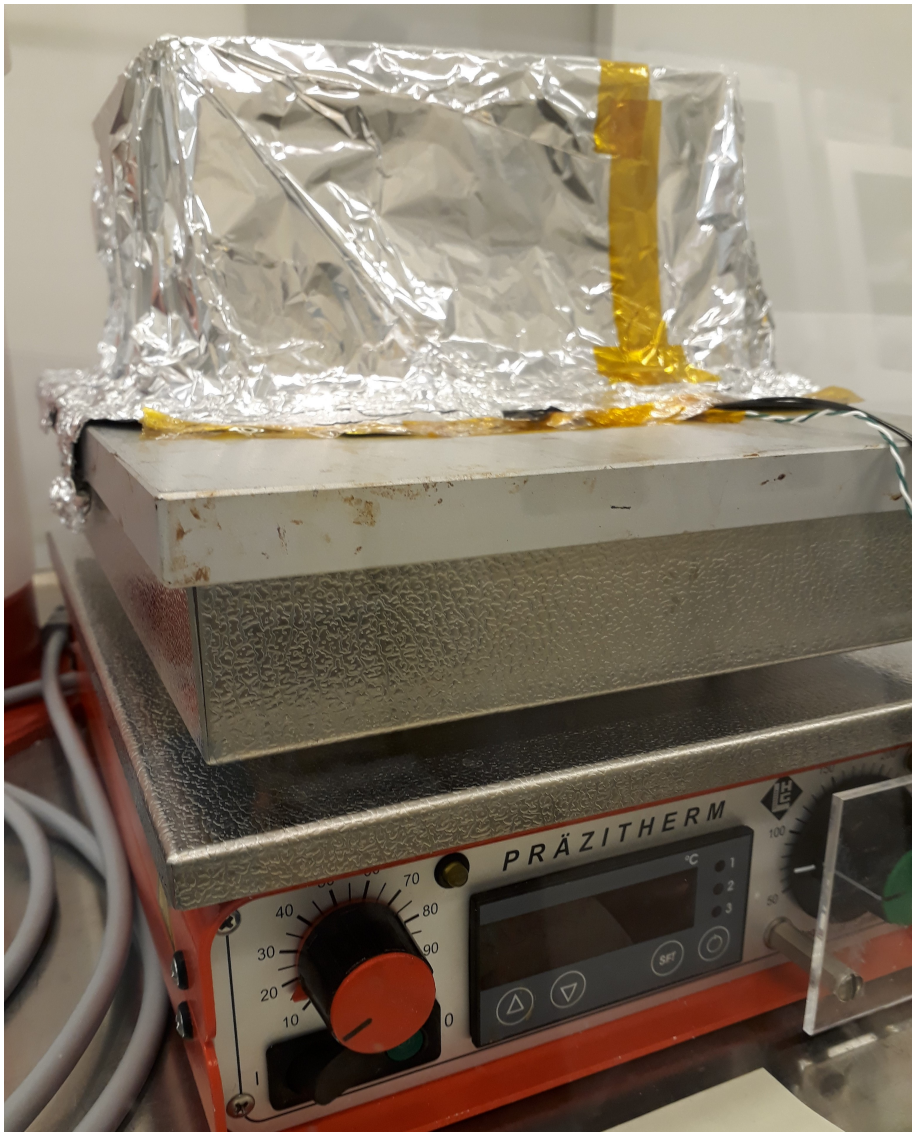

Figure 19. An image of the heater used in the verification measurements