

FMH606 Master's Thesis 2022  
Process Technology

**Understanding influential process  
parameters of NPK fertilizer production; A  
study on prilling process in the NPK3 prilling  
tower (POR)**



**Knowledge grows**

Anders Aaland Ytterbø

Faculty of Technology, Natural Sciences and Maritime Sciences  
Campus Porsgrunn



**Course:** FMH606 Master's Thesis 2022

**Title:** *Understanding influential process parameters of NPK fertilizer production; A study on prilling process in the NPK3 prilling tower (POR)*

**Pages:** 165

**Keywords:** *Prilling, Process, NPK, Compound fertilizer, Trajectory, Heat transfer, Transient, Conduction, Convection, Sphere, Crystallization, Solidification, Sensitivity, Mathematical modelling, Python*

**Student:** *Anders Aaland Ytterbø*

**Supervisor:** *Chandana Ratnayake*

**External partner:** *Jo-Ela Johansen (Yara International)  
Franz-Otto von Hafenbrädl (SINTEF Industry)*

### **Summary:**

A transient conduction model was implemented in Python to assess the temperature development of NPK particles, produced at Herøya, Porsgrunn. Temperatures when hitting the tower bottom was calculated to be 102.8, 67.5 and 35.5°C for the core, critical radius and surface respectively. The heat release from the prills was compared to an older model, and the updated model reduced the error from 10.98 to 1.30 %.

A sensitivity analysis was performed on the updated model, varying different parameters. Overall, the temperature was most influenced by the particle diameter, followed by the thermal conductivity. Air temperature influence increases while the radius increase, while the opposite is true for air pressure. Measurement data from the sensitivity analysis was used to produce two different methods for analyzing how the behavior of a process is expected to change, based on parameters; The EAD method, and the Trend-line method. Both showing good results within the reference framework.

The updated model was reconfigured to assess different products, produced at Herøya. Core temperatures were used as a predictor of crystallization degrees through a phase-diagram, which ranged from 93.4 % to 95.0 % for particle sizes of 2.85 mm. While the model shows basic forms of compatibility, further chemical and thermodynamical data is required to assess whether the model produces accurate results or not.

*The University of South-Eastern Norway accepts no responsibility for the results and conclusions presented in this report.*



# Preface

The purpose of this master's thesis was to establish improved connections between relevant process parameters and the ongoing NPK prilling process at Herøya, Porsgrunn. This was to be done by creating a functioning heat- and mass-transfer model which could analyze prill-temperatures during its descent, while reviewing the effects of altering different parameters. This work was a combination of reviewing previous work, looking at suitable upgrades, and implementing them in a custom model. The model in this thesis is written by the author, using an older model created by Per Stokka as a basis and reference.

I would like to inform the reader that the purpose of the contents of this thesis is to inspire continual improvement of the process development and characterization at Yara International. As such, I would prefer the reader to consider the model in this thesis to be a foundation upon which more complex elements can be added, rather than a "plug-and-play" product.

The following master's thesis was part of the subject FMH606 Master's Thesis 2022, under the Process Technology field of study. The thesis was a collaboration between the University of South-Eastern Norway, SINTEF Industry and Yara International. I would like to extend my sincerest gratitude to my supervisors Chandana Ratnayake, Franz-Otto von Hafenbrädl and Jo-Ela Johansen for their support and time spent with me on this project. The topic of this thesis proved to be an interesting challenge, and I could not have done it without your help. I would also like to thank Ingrid Buckhurst for helping me navigate through a jungle of internal reports, and to help me tackle some of the complex chemistry behind the compound fertilizer.

Porsgrunn, 17th May 2022

Anders Aaland Ytterbø



# Contents

<b>Preface</b>	<b>5</b>
<b>Contents</b>	<b>9</b>
List of Figures . . . . .	13
List of Tables . . . . .	15
<b>1 Introduction</b>	<b>21</b>
<b>2 System description</b>	<b>23</b>
2.1 Mixing and melt composition . . . . .	25
2.2 Prill cooling and solidification . . . . .	26
2.3 Model assumptions . . . . .	27
<b>3 Velocity model</b>	<b>29</b>
3.1 Stokka's velocity model . . . . .	30
3.2 Updated velocity model . . . . .	32
3.2.1 Numerical solution of velocity equations . . . . .	35
3.2.2 Initial conditions . . . . .	37
<b>4 Heat transfer model</b>	<b>41</b>
4.1 Convection heat transfer . . . . .	41
4.1.1 Forced, external convection . . . . .	42
4.1.2 Dimensionless convection parameters . . . . .	43
4.1.3 Convection coefficient for a sphere in a fluid stream . . . . .	44
4.2 Conduction heat transfer . . . . .	47
4.2.1 Lumped-capacitance method . . . . .	48
4.2.2 Transient conduction . . . . .	51
4.2.3 Critical radius . . . . .	56
<b>5 Phase changes</b>	<b>59</b>
5.1 Introduction to phase-changes . . . . .	59
5.1.1 States of matter . . . . .	59
5.1.2 Heat capacities . . . . .	62
5.2 Solid-liquid phase diagrams . . . . .	63
5.2.1 Ternary phase diagrams . . . . .	64

5.2.2	Crystallization of NPK fertilizer . . . . .	65
5.3	Stokkas heat capacity model . . . . .	66
5.3.1	Application to the conduction model . . . . .	68
<b>6</b>	<b>Results</b>	<b>69</b>
6.1	Velocity model . . . . .	69
6.1.1	Prill trajectories . . . . .	70
6.1.2	Vertical position and velocity profiles . . . . .	71
6.1.3	Time-of-flight and terminal velocity . . . . .	73
6.2	Composite Ytterbø model . . . . .	74
6.2.1	Validation of updated model . . . . .	78
6.3	Sensitivity analysis . . . . .	79
6.3.1	Example: Calculating new critical temperature . . . . .	82
6.4	Model adaptation for other products . . . . .	85
6.4.1	NPK 25-2-6 (MOP) . . . . .	86
6.4.2	NPK 21-4-10 S+B . . . . .	87
6.4.3	NPK 21-06-13S . . . . .	88
6.4.4	NPK 25-07-07 (MOP) . . . . .	89
<b>7</b>	<b>Discussion</b>	<b>91</b>
7.1	The updated velocity model . . . . .	91
7.2	Updated heat transfer model . . . . .	92
7.3	Sensitivity analysis . . . . .	93
7.4	Model adaptation and NPK product analysis . . . . .	95
<b>8</b>	<b>Conclusion</b>	<b>97</b>
<b>9</b>	<b>Further work</b>	<b>99</b>
9.1	Model upgrades . . . . .	99
9.2	Product data upgrades . . . . .	100
	<b>Bibliography</b>	<b>101</b>
<b>A</b>	<b>Task description</b>	<b>105</b>
<b>B</b>	<b>Sensitivity analysis data</b>	<b>109</b>
B.1	Summary data . . . . .	109
B.2	Individual run data . . . . .	116
<b>C</b>	<b>AN-MAP-Water phase diagram</b>	<b>137</b>
<b>D</b>	<b>NPK product data</b>	<b>139</b>
D.1	NPK 25-2-6 . . . . .	140



D.2	NPK 21-4-10 . . . . .	142
D.3	NPK 21-06-13 . . . . .	144
D.4	NPK 25-07-07 . . . . .	146
<b>E</b>	<b>Python model source code</b>	<b>149</b>
E.1	Composite Ytterbø model . . . . .	149
E.2	Root program . . . . .	164



# List of Figures

- 2.1 Simplistic sketch of a prilling tower and relevant processes. . . . . 24
- 3.1 Simplistic sketch of a particle falling in a fluid, and the forces acting on it. 29
- 3.2 Sketch of Stokka’s velocity phases. The model consists of an acceleration phase at the beginning of the trajectory, before eventually reaching its terminal velocity. Horizontal velocity,  $u_h$ , is shown in green, vertical velocity,  $u_v$ , is shown in blue while their resulting total velocity vector,  $u_{tot}$ , is shown in red. . . . . 31
- 3.3 Force decomposition on a particle with an x-, and y-velocity. Subscripts denote nature of force, then direction. . . . . 33
- 3.4 Visualising the spiraling movement of the particles out of the centrifuge. . 34
- 3.5 Mesh in time with corresponding discrete values (unknowns). . . . . 36
- 3.6 Geometric considerations of a centrifuge/prilling bucket in regards to particle ejection. . . . . 38
- 4.1 Fluid flow past a hot surface. Figure shows the a velocity and temperature profile as a function of distance from the plate (y). . . . . 42
- 4.2 Reynolds experiment. The experiment was carried out by adding a dye to water travelling at different speeds. The dye had approximately the same specific density as water, and would therefore follow the fluid-stream pattern. [29] . . . . . 44
- 4.3 Simple sketch of a fluid ( $T_\infty, u_\infty$ ) flowing past a sphere with a set surface temperature ( $T_s$ ) in a specific convective regime ( $Re, Nu, Pr, h$ ). The sketch shows the streamlines of the fluid, and a wake left behind the sphere. . . . 45
- 4.4 Conduction heat transfer in a 1D-rod. Figure shows system sketches and temperature graphs as a function of time and position. . . . . 47
- 4.5 Sketch of system where lumped capacitance method is commonly used. Figure shows a solid of uniform initial temperature, suddenly exposed to different conditions. . . . . 49
- 4.6 Comparison of three different solid systems. System one exhibits a high thermal conductivity, resulting in nearly even surface temperatures. System 2 has a lower thermal conductivity, causing an increase in  $\Delta T_s$ . System 3 has the lowest thermal conductivity and therefore also the highest  $\Delta T_s$ . . 50
- 4.7 Spherical object being exposed to convection conditions. . . . . 52

4.8	Representing a point in space using spherical coordinates. An arbitrary point P is shown (purple dot) using $r$ , $\theta$ and $\varphi$ coordinates instead of the cartesian $x$ , $y$ and $z$ . . . . .	53
4.9	Graph showing the equation $1 - x \cot x - 0.3 = 0$ . This is equivalent to solving equation 4.27 using $Bi = 0.3$ . The positive roots of this equation ( $\zeta_n$ ) are used in equation 4.25 and 4.26. This figure was made using GeoGebra's online graphing tool. . . . .	54
4.10	Comparison between a one term, two term and three term Fourier series solution, where the object being investigated is a sphere, suddenly subjected to convection conditions. [38] . . . . .	56
4.11	Sketch of particle showing the core radius, $r = 0$ , critical radius, $r = r_{crit}$ and outer (surface) radius, $r = r_o$ . The sketch also shows how the critical radius is defined: the radius where 20% of the volume of the sphere is contained. . . . .	57
5.1	Sketch of two common crystalline structures of AN. The left sketch shows a cubic lattice structure, which commonly occurs at temperatures ranging from $\sim 170$ to $125$ °C. The right sketch shows a tetragonal structure, which occurs at a temperature range between $125$ to $84$ °C. . . . .	60
5.2	State transition of water, represented in a temperature-energy diagram. Blue lines show sensible heat which changes temperature, while the red lines show latent heat, altering the state of the substance. . . . .	61
5.3	Diagram of a general binary eutectic mixture. [42] . . . . .	63
5.4	General ternary diagram. Figure shows calculation of weight-% at a single point. Layout created using TernaryPlot.com. . . . .	64
5.5	Characteristics of a steep crystallization curve. $T_n$ and $T_{n+1}$ are temperatures at the start and end of a specific interval. $\epsilon$ is the degree of crystallization within that interval. . . . .	66
5.6	Breakdown of heat capacity intervals and degrees of crystallization for a particle. . . . .	66
5.7	Flowchart depicting the general calculation procedure, accounting for Stokka's heat capacity model. . . . .	68
6.1	Particle trajectories for different particle diameters. . . . .	70
6.2	Prill trajectories as a function of centrifuge speeds (RPM). The prill diameter in this simulation was kept constant at $d_p = 2.85$ mm. . . . .	71
6.3	Vertical position for different particle diameters as a function of time. . . . .	72
6.4	Horizontal ( $u_x$ ) and vertical ( $u_y$ ) velocity profiles for different particle diameters. . . . .	72
6.5	Comparison of time-of-flight for different particle sizes between Stokka's original velocity model and the updated velocity model. . . . .	74

6.6	Graph showing the equation $1 - \zeta_n \cot(\zeta_n) - 0.5446836972029505 = 0$ . The first four positive roots and their values are shown with a red triangle. This figure was made using GeoGebra's online graphing tool. . . . .	76
6.7	Temperature development during the descent of a single prill at three different radii. . . . .	77
6.8	Figure showing the temperature- and temperature difference development on the surface of a prill as a result of varying the ambient air temperature in the composite Ytterbø model. Black dotted lines represent the reference value, which, in the case of ambient air temperature is equal to a simulation with $T_{air,in} = 22^\circ\text{C}$ . . . . .	81
6.9	Temperature development of a 2.85 mm prill of NPK 25-2-6. Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line). . . . .	86
6.10	Temperature development of a 2.85 mm diameter prill of NPK 21-4-10 S+B. Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line). . . . .	87
6.11	Temperature development of a 2.85 mm prill of NPK 21-06-13S. Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line). . . . .	88
6.12	Temperature development of a 2.85 mm prill of NPK 25-07-07 (MOP). Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line). . . . .	89
C.1	AN-MAP-Water phase diagram. Phase diagram was made digital by the author, but all information was gathered from Stokka, P. (1985) [20]. . . .	137
D.1	Simplified sketch of the AN-MAP-Water phase diagram, can be used for reference when examining figures in Appendix B. . . . .	139
D.2	AN-MAP-Water phase diagram for NPK 25-2-6 . . . . .	140
D.3	AN-MAP-Water phase diagram for NPK 21-4-10 . . . . .	142
D.4	AN-MAP-Water phase diagram for NPK 21-06-13 . . . . .	144
D.5	AN-MAP-Water phase diagram for NPK 25-07-07 . . . . .	146



# List of Tables

3.1	Decomposed velocity and the forces acting in each direction. . . . .	34
3.2	Initial conditions for updated velocity model. . . . .	39
5.1	Temperature breakpoints used in determining the intervals, and subsequent heat capacities for different NPK products. . . . .	67
6.1	Parameters and values used for assessing a standalone velocity model. . . .	69
6.2	Time-of-flight and terminal velocities for 6 different particle diameters using an updated velocity model. . . . .	73
6.3	Time-of-flight, terminal velocities and acceleration velocities for 6 different particle diameters using a Python adapted version of Stokka's BASIC program model. . . . .	73
6.4	Simulation parameters for running the composite Ytterbø model. These parameters are essential parameters upon which further necessary parameters were calculated. . . . .	75
6.5	Simulation parameters calculated using composite Ytterbø model. These parameters are calculated as a part of finding the temperature of a prill as a function of time and radius. . . . .	75
6.6	Initial, final and delta temperatures for the three radii. . . . .	77
6.7	Case numbers, parameters, units, ranges and reference values used in the sensitivity analysis of NPK 21-4-10. . . . .	80
6.8	Average expected temperature difference at given radii per unit increase/decrease for 6 different parameters. . . . .	82
6.9	Sensitivity ranking of different model parameters based on radius . . . . .	82
6.10	Calculated critical temperatures after finding new particle diameter through the EAD and trend-line method. . . . .	84
6.11	Calculated parameters from running the composite Ytterbø model with NPK 25-2-6 (MOP) as product. . . . .	86
6.12	Calculated parameters from running the composite Ytterbø model with NPK 21-4-10 S+B as product. . . . .	87
6.13	Calculated parameters from running the composite Ytterbø model with NPK 21-06-13S as product. . . . .	88
6.14	Calculated parameters from running the composite Ytterbø model with NPK 25-07-07 (MOP) as product. . . . .	89





# Nomenclature

Abbreviation	Explanation
AN	Ammonium nitrate
MAP	Mono-ammonium phosphate
DAP	Di-ammonium phosphate
MOP	Muriate of potash
SOP	Sulphate of potash
NPK	Compound fertilizer (Nitrogen-Phosphate-Potassium)
NP	Nitrophosphate
NSP	Non-spherical particle
CFD	Computational fluid dynamics
LCM	Lumped capacitance method
BC	Boundary conditions
IC	Initial condition
PDE	Partial-differential equation
OTA	One term approximation
T.O.F	Time-of-flight
RPM	Rotations per minute
EAD	Expected average difference

Symbol	Explanation	Unit
$F_g$	Gravitational force	[N], $[\frac{\text{kg}\cdot\text{m}}{\text{s}^2}]$
$F_d$	Drag force	[N], $[\frac{\text{kg}\cdot\text{m}}{\text{s}^2}]$
$F_b$	Buoyancy force	[N], $[\frac{\text{kg}\cdot\text{m}}{\text{s}^2}]$
$m$	Mass	[kg]
$g$	Gravitational acceleration	$[\frac{\text{m}}{\text{s}^2}]$
$\rho_f$	Fluid density	$[\frac{\text{kg}}{\text{m}^3}]$
$V_d$	Displaced volume	$[\text{m}^3]$
$u$	Velocity	$[\frac{\text{m}}{\text{s}}]$
$K$	Drag constant	$[\frac{\text{kg}}{\text{m}}]$
$c_d$	Drag coefficient	[-]

$A_p$	Projected area	$[\text{m}^2]$
$a$	Acceleration	$[\frac{\text{m}}{\text{s}^2}]$
$\phi$	Sphericity	$[-]$
$u_{tot}$	Total velocity	$[\frac{\text{m}}{\text{s}}]$
$u_h$	Horizontal velocity	$[\frac{\text{m}}{\text{s}}]$
$u_v$	Vertical velocity	$[\frac{\text{m}}{\text{s}}]$
$u_s$	Stationary/terminal velocity	$[\frac{\text{m}}{\text{s}}]$
$d_p$	Particle diameter	$[\text{m}]$
$\rho_p$	Particle density	$[\frac{\text{kg}}{\text{m}^3}]$
$V_p$	Particle volume	$[\text{m}^3]$
$t$	Time	$[\text{s}]$
$\omega$	Angular velocity	$[\frac{\text{rad}}{\text{s}}]$
$H$	Height	$[\text{m}]$
$H_e$	Ejection height	$[\text{m}]$
$T$	Temperature	$[\text{K}]$
$k$	Thermal conductivity	$[\frac{\text{W}}{\text{m}\cdot\text{K}}]$
$\dot{q}$	Heat sink/source	$[\text{W}], [\frac{\text{J}}{\text{s}}]$
$\rho$	Density	$[\frac{\text{kg}}{\text{m}^3}]$
$C_p$	Heat capacity, constant pressure	$[\frac{\text{kJ}}{\text{kg}\cdot\text{K}}]$
$\dot{E}_{in}$	Energy flow into the system	$[\text{W}], [\frac{\text{J}}{\text{s}}]$
$\dot{E}_{out}$	Energy flow out of the system	$[\text{W}], [\frac{\text{J}}{\text{s}}]$
$\dot{E}_{gen}$	Energy generated in the system	$[\text{W}], [\frac{\text{J}}{\text{s}}]$
$\dot{E}_{acc}$	Energy accumulated in the system	$[\text{W}], [\frac{\text{J}}{\text{s}}]$
$h$	Convective heat transfer coefficient	$[\frac{\text{W}}{\text{m}^2\cdot\text{K}}]$
$T_f$	Final temperature	$[\text{K}]$
$T_\infty$	Bulk fluid temperature	$[\text{K}]$
$T_i$	Initial temperature	$[\text{K}]$
$t_f$	Final time	$[\text{s}]$
$Bi$	Biot number	$[-]$
$L_c$	Characteristic length	$[\text{m}]$
$T_s$	Surface temperature	$[\text{K}]$
$L$	Length	$[\text{m}]$
$r_o$	Outer radius	$[\text{m}]$
$\alpha$	Thermal diffusivity	$[\frac{\text{m}^2}{\text{s}}]$
$\theta^*$	Dimensionless temperature	$[-]$
$r^*$	Dimensionless radius	$[-]$
$Fo$	Fourier number/dimensionless time	$[-]$
$Q_o$	Maximum theoretical heat transfer	$[\text{J}]$
$Q$	Actual heat transfer	$[\text{J}]$
$r_{crit}$	Critical radius	$[\text{m}]$
$q''$	heat flux	$[\frac{\text{W}}{\text{m}^2}]$

$\delta$	Boundary layer thickness	[m]
$u_\infty$	Bulk fluid velocity	$\left[\frac{m}{s}\right]$
$q''_{conv}$	Convection heat flux	$\left[\frac{W}{m^2}\right]$
$q''_{cond}$	Conduction heat flux	$\left[\frac{W}{m^2}\right]$
$Nu$	Nusselt number	[-]
$Pr$	Prandtl number	[-]
$Re$	Reynolds number	[-]
$\nu$	Kinematic viscosity	$\left[\frac{m^2}{s}\right]$
$\mu$	Dynamic viscosity	$\left[\frac{N \cdot s}{m^2}\right]$
$\Delta H$	Enthalpy change	$\left[\frac{kJ}{kg}\right]$
$\Delta H_{fus,vap}$	Heat of vaporization/fusion	$\left[\frac{kJ}{kg}\right]$
$m_{cr}$	Mass available for crystallization	[kg]
$\varepsilon$	Degree of crystallization	[-]
$Q_{cr}$	Heat of crystallization	$\left[\frac{kJ}{kg}\right]$
$m_{tot}$	Total mass available	[kg]
$T_n$	Temperature at interval end	[K]
$T_{n-1}$	Temperature at interval start	[K]
$C_{ps}$	Heat capacity of solids	$\left[\frac{kJ}{kg \cdot K}\right]$
$\bar{X}_s$	Fraction of solids	[-]
$C_{pl}$	Heat capacity of liquids	$\left[\frac{kJ}{kg \cdot K}\right]$
$\bar{X}_l$	Fraction of liquids	[-]



# 1 Introduction

Ever since the beginning of agriculture during the neolithic revolution, some 12 000 years ago, farming to produce crops have been a major influence on the economic, cultural and technological development of the world. Plants and crops need a specific set of macro- and micro-nutrients to grow effectively, and once this principle was discovered, the growth of agriculture skyrocketed. An important piece of technology that has greatly influenced the role of agriculture in the modern society is the use of fertilizers. Early on, Egyptians, Romans and Babylonians were known to use minerals and organic manure in the soil to increase crop yield. The minerals and manure contained important organic compounds that could be broken down into plant-available nutrients. A highly effective form of manure was in the form of guano, or bat- and seabird excrement. This manure contained a high amount of nitrogen, phosphate and potassium. All chemical compounds that are essential for plant growth. Guano use has been recorded as far back as the 1500's in the Andes, but global use did not occur until the 1900's, when it was sold and shipped to large parts of the US and Europe. Before this time, Europe used a nitrogen-fixation method that involved planting clover, which allowed for organic nitrogen fertilizer production. [1] This was part of the "crop-rotation" method, which sparked agricultural growth in Britain, and is speculated to be one of the main initializers of the Industrial Revolution.

[2]

Fast forward to the beginning of the 1900's, and the introduction of chemically produced nitrogen fertilizers. Over the course of the industrial revolution, nitrogen was shown to have a major impact on crop yield, and instead of using natural clover to fixate atmospheric nitrogen, processes began emerging that took this to the next level. The Birkeland-Eyde process was the one of the first industrial processes that used atmospheric nitrogen to produce the water soluble nitrogen-compound,  $\text{HNO}_3$ , or Nitric Acid [3]. This process was originally developed by Kristian Birkeland, whom, together with Sam Eyde, started the company Norsk Hydro. Around 1927, an inventor, named Erling Johnson procured a method for producing nitrogen fertilizer by acidifying phosphate rock in nitric acid. This process was named the Odda-process, after the location where it was invented. Although Johnson's company did not use the process directly, it was licensed to Norsk Hydro, which used the process to produce compound fertilizers, known as NPK (Nitrogen(N), Phosphate(P), Potassium (K)). [4]. This led to a major rise in the agricultural influence of Norsk Hydro, and in 2004, a new-founded daughter company of Norsk Hy-

dro, named Yara International was noted on the stock exchange for the first time. Yara International would completely take responsibility for fertilizer production, which had previously been under Norsk Hydro. Yara is one of the few companies that still employ the Odde-process to this day, and sport production facilities around the world. One of the main facilities that uses the Odde process to produce compound fertilizers is located at Herøya, Porsgrunn in Norway. Yara Porsgrunn has the highest production capacity of NPK fertilizers in Europe.

[4], [5]

A key in producing NPK fertilizers lie not only in the products chemistry, and chemical makeup, but also its physical properties. Farmers that receive the fertilizers should obtain a product that ensures high efficiency, good crop yields and ease of use by reducing maintenance upkeep. Many of these beneficial properties are ensured by producing particles of specific sizes and shapes. This can be achieved in several different ways, but one way of doing it is by solidification in a prilling tower. The resulting particles are aptly named *prills*. Production of prills is an essential part of the nitrophosphate process at Yara Porsgrunn. The general idea is that the liquid components are solidified by cooling in ambient air. The residence time is obtained by allowing the prills to fall roughly 50-80m against a co-current stream of ambient air. At Porsgrunn's facilities, Yara inhabits two such prilling towers, connected to compound fertilizer production. Because of the general population rise, and increase in demand, both facilities have to operate at larger capacities than what was originally intended. In the last decades, notable problems have occurred when producing specific products in the NPK-3 factory. The main result of these problems are the production of prills that vary in both size and shape, both qualities that are important for the end-result that the farmers obtain. Ideally, the prills would be spherical, to reduce abrasion and loss of material through handling and transporting, while also reducing problems related to end-user dispersion. However, prills that are considered non-spherical are becoming an increasing problem. In order to combat this problem, a large undertaking has been made to try and assess the root causes of these non-spherical particles (NSP). This thesis is an attempt at providing a basis for understanding the physical nature of the NPK products, and how they behave in the prilling tower. Hopefully some connections can be made which will help alleviate production issues in the future. This is done through the lens of a heat and mass balance model.

## 2 System description

The main topic of this thesis is to establish a working, albeit simplified, heat and mass balance model to establish connections between different operating parameters and how they influence the prilling process. The system that is being modeled in this thesis is prilling tower in Yara's NPK factory number 3 (NPK3, or Fgj.3) located at Herøya, Porsgrunn. The main function of this factory is to produce compounds fertilizers in the form of prills, using the nitrophosphate route, which was invented at Odda, Norway. This factory contains both wet- and dry processes. The wet process encapsulates digestion of phosphate rock, crystallization and filtration of calcium nitrate, neutralization of nitrophosphate (NP)-melt and evaporation. After evaporation, the process is considered "dry". The water content of the NP-melt after evaporation is usually around 0.5 % [4]. After evaporation, the NP melt is mixed together with salts and other additives to give the final product the desired nutrient content and specified chemical composition. The melt is solidified by ejecting the molten NP-mix from a centrifuge located high above the ground. The ejected particles are then met with a counter-current flow of ambient air that cools the particles, and cause them to solidify. In order to control dust emission through abrasion during handling and transportation, as well as provide optimal conditions for commercial fertilizer spreaders, it is important that the solidified particles are close to spherical. A key aspect in determining whether a particle will become spherical or not is its degree of crystallization [6]. If a particle is not solidified enough when it hits the tower bottom, it could deform. There are parameters that can cause particles to be considered "non-spherical", even if they survive the fall, but for the purposes of this thesis, any particle that hits the tower bottom that has been crystallized below a certain threshold is considered to become non-spherical. In order to establish whether or not a particle has been properly solidified during its descent, it is important to look at the particles temperature development. Normally, prilling tower models are split in three sections:

1. Liquid temperature drop phase
2. Solidification phase
3. Solid temperature drop phase

This is a common assumption to make when considering single-component particles, such as Urea or ammonium nitrate (AN) [7]–[9], but is unfit for evaluating multi-component mixtures, such as NPK. The model in this thesis incorporates a transient conduction

model together with particle composition and crystallization data to present a temperature development. This is done using four different sub-models:

- Mixer composition
- Particle velocity and trajectory
- Conduction heat transfer
- Heat capacity and crystallization

A basic sketch of the prilling tower system can be seen in figure 2.1.

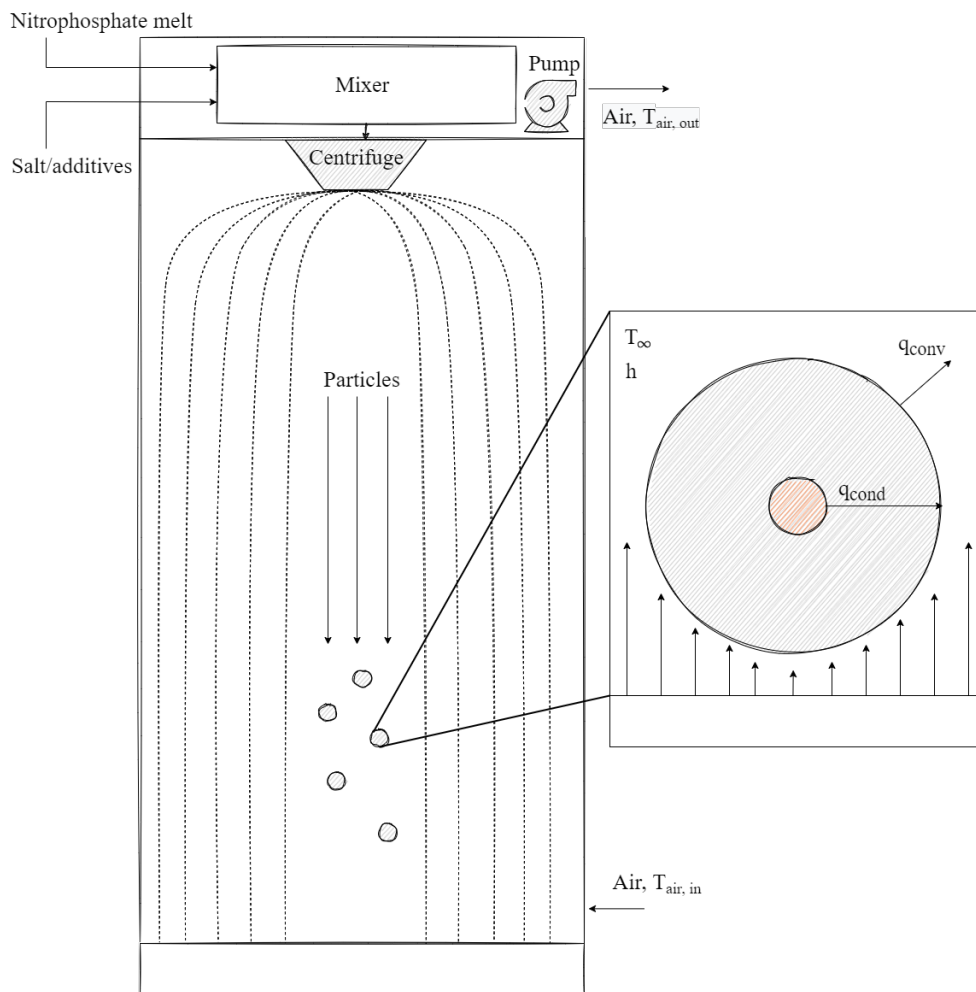


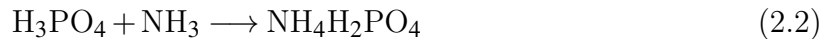
Figure 2.1: Simplistic sketch of a prilling tower and relevant processes.

Now that a general process overview has been established, it is time to look a bit more closely into the different sub-categories, and what assumptions are made when generating a mathematical computer model.

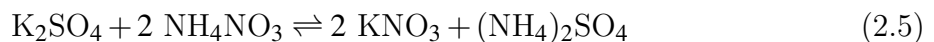
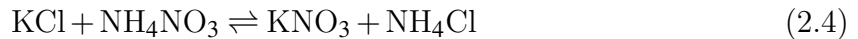


## 2.1 Mixing and melt composition

The mixer is considered in this thesis to be the first step of the overall prilling process. There are two main components in the mixer, and these are nitrophosphate (NP) melt and a salt mixture. Before entering the mixer, the main constituents of the NP-liquor (N/P solution) are phosphoric acid ( $\text{H}_3\text{PO}_4$ ) and nitric acid ( $\text{HNO}_3$ ). Other constituents are dissolved calcium and magnesium, hydrofluoric acid (HF) and insoluble compounds. Before entering the mixer, this liquid is treated with mixed with ammonium nitrate ( $\text{NH}_4\text{NO}_3$ ) and neutralized in ammonium gas ( $\text{NH}_3$ ). This section is where the nitrogen and phosphates are converted to the form they will have in the product, and is where the N/ $\text{P}_2\text{O}_5$  ratio is controlled. In addition, calcium and other impurities are precipitated as fluorides and phosphates. The main reactions that take place include [4]:



Ammonium nitrate (AN) and mono-ammonium phosphate (MAP) are considered to be the main constituents. Di-ammonium phosphate (DAP) can also be present, but its formation is strictly controlled through pH regulation. DAP will more readily dissociate back into ammonia gas [10], which is due to the increased ammonia vapour pressure during formation [11]. The low pH value controls the concentration of  $\text{HPO}_4^{2-}$  ions, and therefore DAP formation. (Figure 4, p.431 [12]). After neutralization, the NP-melt is sent to the mixer, where salts and other additives are added. The main salt component is potassium (K), in the form of potassium chloride (KCl) or potassium sulphate ( $\text{K}_2\text{SO}_4$ ), however, other salts such as ammonium sulphate ( $(\text{NH}_4)_2\text{SO}_4$ ), magnesium sulphate ( $\text{MgSO}_4$ ), dolomite and borax can be added. These are largely considered as inert [12] (i.e they do not alter or participate in chemical reactions), but potassium chloride/sulphate are not. These compounds react with the ammonium nitrate to produce potassium nitrates ( $\text{KNO}_3$ ), ammonium chlorides ( $\text{NH}_4\text{Cl}$ ), and ammonium sulphates according to the following reactions [12]:



As seen in the reactions above, these are equilibrium reactions. This means that they are *reversible*. Which means that the reaction can progress forwards or backwards depending on process conditions [13]. Neither potassium chloride or potassium sulphate will be 100% converted during the fall, but changes continue to occur, even after storage [14]. The choice between supplying potassium in the form of chloride or sulphate is dependent on the crop to be fertilized and the area. Potassium sulphate is usually preferred in soils where chloride accumulation is problematic [15]. In addition to adding salts and other micro-nutrients, off-spec product that do not pass screening are sent back to the mixer to

be recycled. Smaller particles are sent back as-is, while larger particles are crushed before recycling [16].

## 2.2 Prill cooling and solidification

The mixer is designed in such a way as to produce a constant overflow. This overflow is fed to a rotating prill bucket, where the mixture is ejected. The bucket itself features perforated holes of varying sizes, where the smaller holes are located at the bottom and the larger holes are located at the top. Even though the holes in the prill bucket consist of set sizes, Rayleigh, L. (1887) found that the final prill sizes are heavily influenced by the break-up of the jet as the liquid is ejected from the bucket [17]. This explains why it is common to see particles with sizes that significantly deviate from the holes of the prilling bucket, and why bucket design is highly influential.

Once the melt is ejected from the bucket, it is immediately met by a stream of ambient air, flowing in a cross current manner. The air is pumped into the tower from an inlet at the bottom, through to the fans in the prill tower stack, situated at the top. The formed droplets spread out, according to their sizes and point of ejection, and form distinct trajectories. Throughout the fall, the droplet experiences a convection heat transfer from its surface to the surrounding air, and a conduction heat transfer from the core of the droplet to the surface. The convection heat transfer is largely dependent on the temperature gradient between the surrounding air and the surface, and the conduction transfer is highly dependent on the thermal conductivity and heat capacities of the material(s) that makes up the melt. A significant challenge when modelling such a system is to account for the ever-changing properties of the droplet and its surroundings. The flow-regime of the air has been observed to be laminar in certain areas, while turbulent pockets appear other places. Ricardo et.al (2013) reported perturbations in the air flow at and around the melt-ejection height, due to high droplet concentration. [18]. The composition of the liquid and solid fractions of the particles are constantly altered due to the cooling while in flight, which also leads to a change in thermal conductivity and heat capacity. In order to produce a feasible model, without the use of CFD tools, some assumptions had to be made.

[4], [19]

## 2.3 Model assumptions

A summary of the assumptions used during creation of the composite Ytterbø model is given below.

- Laminar air flow throughout the tower.
- Constant air temperature throughout the tower,  $\frac{dT_{air}}{dH_{tower}} = 0$ .
- Particle movement is considered to be two, dimensional. I.e, no spiraling movement.
- Falling particles do not influence each other.
- All heat absorbed by the air comes from the heat released by the particles during their descent.
  - Radiation heat from the walls is considered to be negligible.
  - Radiation heat from particles at tower bottom is considered to be negligible.
- Radiation coefficient is considered to be constant.
- Convection heat coefficient is averaged during the entire descent, and is constant along the particle surface.
- Thermal conductivity of particles are considered constant, regardless of salt composition.
- Crystallization behaviour of NP-melt is assumed to only be dependent upon AN, MAP and water content.
  - Remaining compounds are considered inert.
  - Potassium chloride's contribution to initial crystallization temperature is not accounted for.
  - Poly-phosphates contribution to crystallization behaviour is not accounted for.
- Chemical changes happen instantly, at specified temperatures.
- Heat of crystallization for all products are based on data available from AN-MAP-KCl melt, regardless of melt composition.
- The behavior of particles with a given distribution can be examined by analyzing a single particle with the average size,  $d_p$ .

[6], [20]–[22]



### 3 Velocity model

In order to establish a functioning heat- and mass transfer model in the prilling tower, it is important to accurately describe how a particle descends in the tower. This can be investigated by looking at its velocity and particle trajectory. Combining these factors will give us a residence time for the particle, or "time-of-flight", which is very important in determining how much heat is released, since we know that the amount of heat released is a function of time. Consider a particle falling in a fluid (air, water etc.) This general system can be depicted as seen in figure 3.1 below.

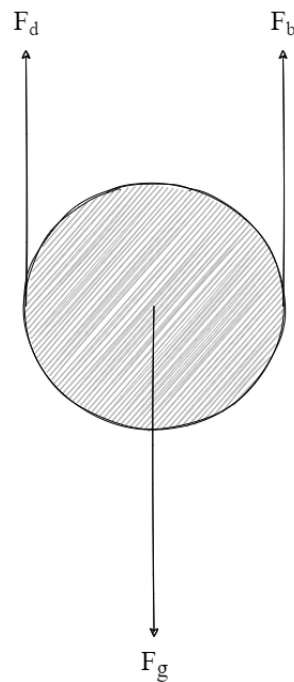


Figure 3.1: Simplistic sketch of a particle falling in a fluid, and the forces acting on it.

Three forces are acting on the particle. The force of gravity  $F_g$ , is pulling the particle downwards while the drag,  $F_d$ , and buoyancy force,  $F_b$ , are resisting this downward motion, pulling in the opposite direction. The drag force can be described as a resistant force. While the particle is falling, it has to push the fluid out of its way, which requires force. The drag force is a function of flow conditions and particle geometry. The buoyancy force

can be described as an upward force that is generated due to a pressure difference in a fluid. Generally, fluid pressure increases with depth, which in turn causes an imbalance between the forces acting on an immersed object. This imbalance generates a net upward thrust, given that the density of the immersed object is lower than the density of the fluid. The net buoyancy force is dependant on the volume of fluid that has been displaced.

[23]

The equations for these forces are dependent on the geometry and flow conditions, but considering a particle, can be expressed as:

$$F_g = mg \quad (3.1)$$

$$F_b = \rho_f V_d g \quad (3.2)$$

$$F_d = Ku^2 \quad (3.3)$$

Where:

$$K = \frac{1}{2} \rho_f C_d A_p \quad (3.4)$$

And from the immortal teachings of Isaac Newton, we know that:

$$\sum F = ma \quad (3.5)$$

[23], [24]

These equations can be combined to create our basis for calculating particle trajectory and accompanying velocities. Before those equations are derived, Stokka's velocity model will be presented.

### 3.1 Stokka's velocity model

Stokka, P. (1989) derived a simplified velocity model, fit for use in a BASIC program [6]. The model essentially describes a particle as having two discrete phases when descending through the tower:

1. An accelerating phase
2. A non-accelerating phase

That is, the particle accelerates until reaching terminal velocity, where the velocity is kept constant. Stokka uses the following assumptions in his model:

- The initial vertical velocity is  $0 \frac{m}{s}$
- The vertical velocity during acceleration  $= \frac{2}{3} u_s$
- The initial horizontal velocity  $= u_h$

- Particles in question are considered to be perfectly spherical,  $\phi = 1$
- The acceleration phase lasts for 10 s, given particle sizes between 2.5 and 3.5 mm

A graphical summary of this system can be seen in figure 3.2

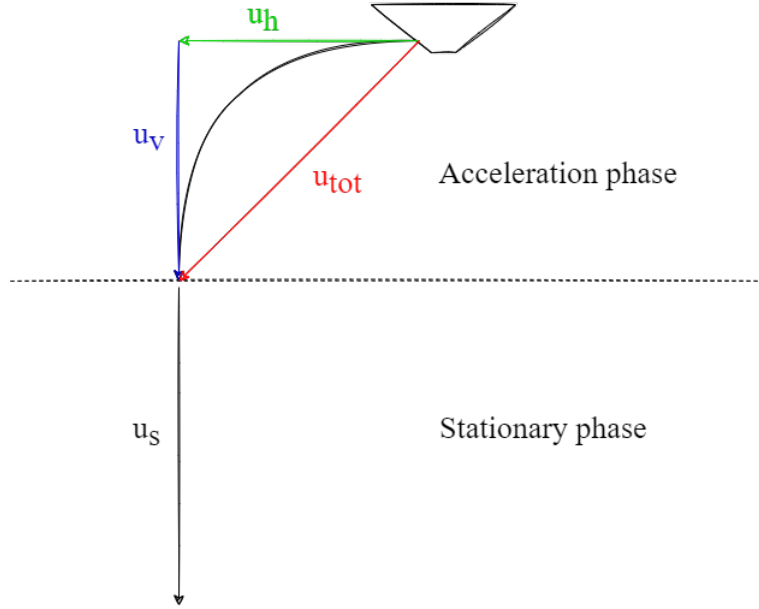


Figure 3.2: Sketch of Stokka's velocity phases. The model consists of an acceleration phase at the beginning of the trajectory, before eventually reaching its terminal velocity. Horizontal velocity,  $u_h$ , is shown in green, vertical velocity,  $u_v$ , is shown in blue while their resulting total velocity vector,  $u_{tot}$ , is shown in red.

Now that the framework of the model has been shown, the equations that govern these phases can be presented. As shown in figure 3.2, the velocity vectors,  $u_h$  and  $u_v$ , can be combined to create a total velocity vector for the acceleration phase,  $u_{tot}$ . This can be combined with the Pythagorean theorem to produce the following equation:

$$u_{tot} = \sqrt{u_h^2 + u_v^2} \quad (3.6)$$

This equation is then combined with one of Stokka's original assumptions to yield:

$$u_{tot} = \sqrt{u_h^2 + \left(\frac{2}{3}u_s\right)^2} \quad (3.7)$$

For the stationary velocity,  $u_s$ , Stokka uses the following equation:

$$u_s = 1.74 \sqrt{\frac{d_p(\rho_p - \rho_f)g}{\rho_f}} \quad (3.8)$$

The derivation of this equation is not included in the original report, so it is included here. The derivation comes down to using equation 3.1 to 3.5, and setting:

$$\sum F = ma = 0 \quad (3.9)$$

because there is no acceleration. We consider a fully immersed object ( $V_d = V_p$ ), and write the sum of forces as its constituent elements:

$$\begin{aligned} F_g - F_b - F_d = 0 &\implies F_d = F_g - F_b \implies \\ \frac{1}{2}\rho_f C_D A_p u^2 = V_p \rho_p g - V_p \rho_f g &\implies u^2 = \frac{2V_p g}{A_p C_D} \frac{(\rho_p - \rho_f)}{\rho_f} \end{aligned} \quad (3.10)$$

where the volume and *projected* area of a sphere is given by:

$$V_p = \frac{4}{3}\pi r^3 = \frac{4}{3}\pi \left(\frac{d_p}{2}\right)^3 = \frac{1}{6}\pi d_p^3 \quad (3.11)$$

$$A_p = \pi r^2 = \pi \left(\frac{d_p}{2}\right)^2 = \frac{1}{4}d_p^2 \quad (3.12)$$

Combination of these equations result in:

$$u^2 = \frac{2\frac{1}{6}\pi d_p^3 g}{\frac{1}{4}d_p^2 C_D} \frac{(\rho_p - \rho_f)}{\rho_f} \quad (3.13)$$

Cancelling out terms, and leaving  $u_t$  on the left-hand side yields:

$$u_t = \sqrt{\frac{4}{3C_D}} \sqrt{\frac{d_p(\rho_p - \rho_f)g}{\rho_f}} \quad (3.14)$$

where the first square-root term equates to  $\approx 1.74$  if  $C_D \approx 0.44$ , which is a common drag coefficient value to use for spherical objects falling in air [25]. Hence, the resulting equation becomes equal to equation 3.8. This model gives a quick method of estimating time-of-flight of particles of sizes between 2.5 and 3.5 mm. However, with the use of modern programming languages, it is worth looking at a way to more accurately describe the velocity of particles throughout the tower, combined with their resulting trajectories. This will be explored in the following section.

## 3.2 Updated velocity model

In Stokka's model, the acceleration phase is set to last for 10 seconds, while some assumptions about the relationships between the velocities are made in the acceleration



phase. Instead of dividing the velocities into two sections, it would be beneficial to look at the differential form of Newton's 2<sup>nd</sup> law. Consider equation 3.5 written on a differential form:

$$\sum F = ma = m \frac{du}{dt} \quad (3.15)$$

Using this, it would be possible to continually evaluate the velocity of the particle. In addition, by looking at figure 3.1 we know that gravity works downwards, buoyancy works opposite to gravity, while the drag force works opposite to the velocity. However, in the system description, it is given that the particles do not simply move downwards. They arc outwards, which means that at each time,  $t$ , a particle will have a velocity in the x- and y-direction [24]. This can be solved by decomposing the forces into their respective x-, and y-components. A graphical sketch of such a decomposition is shown in figure 3.3.

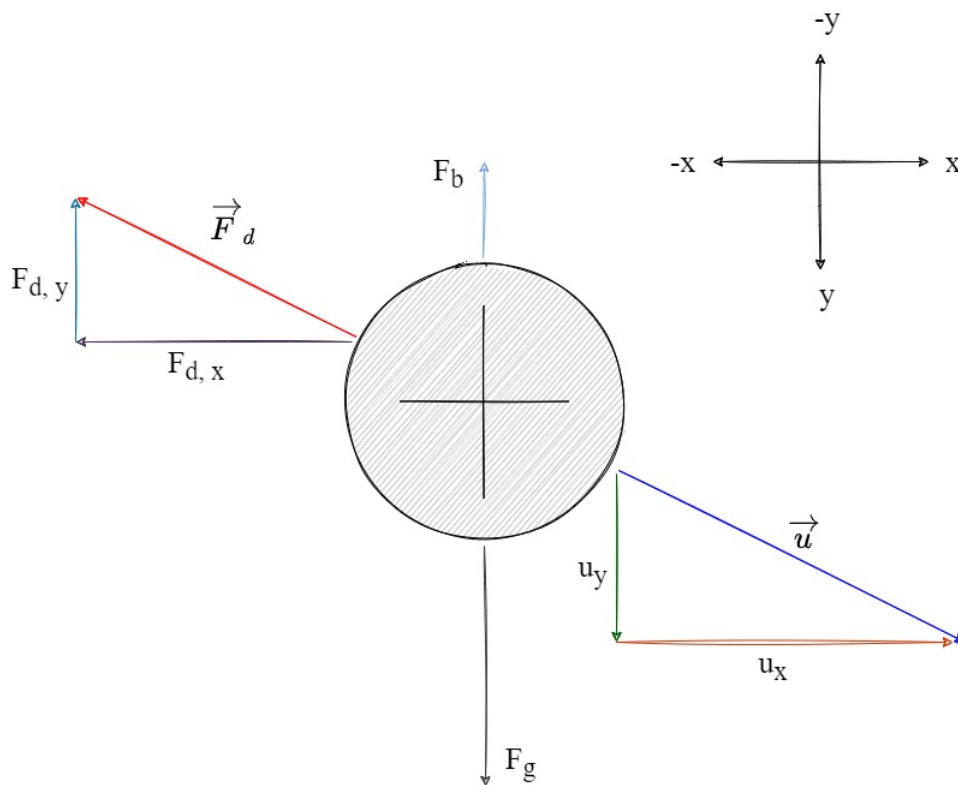


Figure 3.3: Force decomposition on a particle with an x-, and y-velocity. Subscripts denote nature of force, then direction.

This system might seem a little more complex than the one give in figure 3.1, but given that the equations for all forces already have been accounted for in equation 3.1 to 3.4, a lot of the work has already been done. We start by listing the forces, and what component acts in which direction.

Table 3.1: Decomposed velocity and the forces acting in each direction.

	Velocity	Gravity	Buoyancy	Drag
x-direction	$u_x$	0	0	$F_{d,x}$
y-direction	$u_y$	$F_g$	$F_b$	$F_{d,y}$

Using table 3.2 and equation 3.15, terms for the velocity derivative in the x- and y-direction are presented as:

$$m \frac{du_x}{dt} = -F_{d,x} \quad (3.16)$$

and:

$$m \frac{du_y}{dt} = F_g - F_b - F_{d,y} \quad (3.17)$$

which will constitute the core equations. Note that this is under the assumption that there is no spiraling movement, and that the velocity trajectory is according to figure 3.4. The viscous nature of the NP-melt do invoke spiraling [4], but that will not be covered here. This simplification reduces the problems to two dimensions, and as an approximation, holds true for most practical applications.

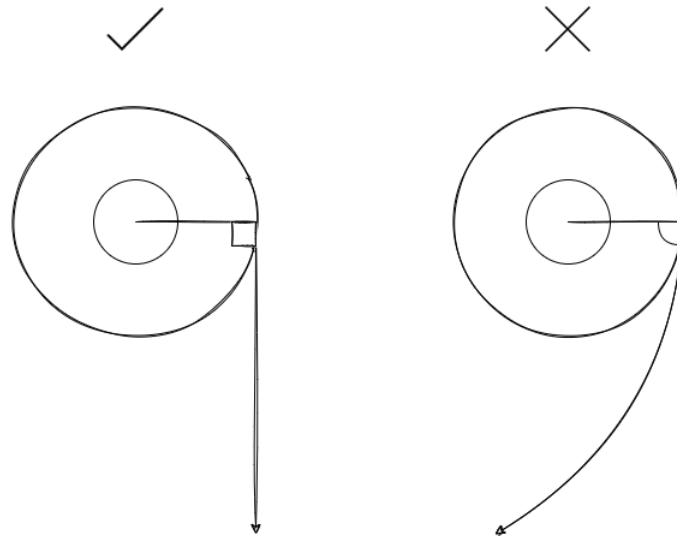


Figure 3.4: Visualising the spiraling movement of the particles out of the centrifuge.

We continue under the newly made assumption by looking at an altered description of equation 3.10:

$$m \frac{du}{dt} = V_p \rho_p g - V_p \rho_f g - Ku^2 \quad (3.18)$$

which, when decomposed in accordance with equation 3.16 and 3.17 becomes:

$$m \frac{du_x}{dt} = -Kuu_x \quad (3.19)$$

$$m \frac{du_y}{dt} = mg - V_p \rho_f g - Kuu_y \quad (3.20)$$

Which can be written as:

$$\frac{du_x}{dt} = -\frac{K}{m}uu_x \quad (3.21)$$

$$\frac{du_y}{dt} = g - \frac{V_p \rho_f g - Kuu_y}{m} \quad (3.22)$$

Now, we have two dependent solutions, since they are both connected by the variable  $\mathbf{u}$ , which can be written using the Pythagorean theorem as:

$$u = \sqrt{u_x^2 + u_y^2} \quad (3.23)$$

Solving this co-dependence requires a numerical solution, the scheme of which will be described in the following sub-chapter.

### 3.2.1 Numerical solution of velocity equations

In this section, the equations presented in the previous section will be solved using the *Euler forward* method. The Euler forward method is based on a forward finite difference method, which can be explained in 4 steps [24]:

1. Introduce  $N_{t+1}$  points in time,  $t_0, t_1, \dots, t_{N_t}$  for the relevant time interval.
2. Utilize that the differential equation is valid at the mesh points.
3. Approximate derivatives by finite differences.
4. Formulate a computational algorithm that can compute a new value  $u_n$  based on previously computed values  $u_i, i < n$ .

Points 1. and 2. are best shown graphically, and an example of a mesh, consisting of points at  $n$  points in time can be seen in figure 3.5.

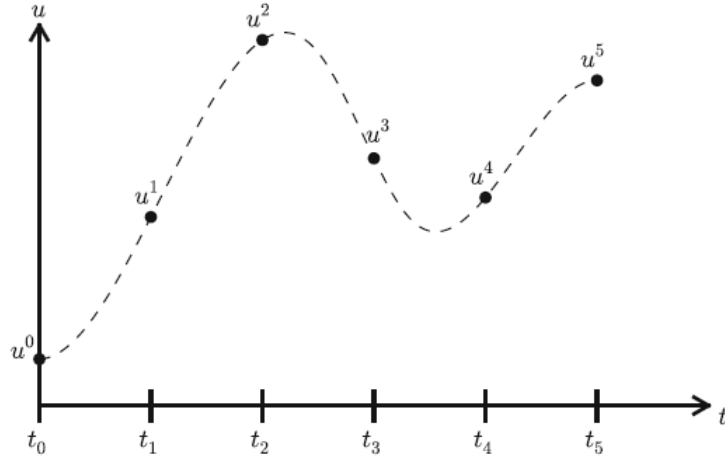


Figure 3.5: Mesh in time with corresponding discrete values (unknowns).

Approximating derivatives by finite differences is the core of this model, and can be expressed mathematically. For example, consider the acceleration of an object,  $a(t)$ , this can be approximated as:

$$a(t) = \frac{du}{dt} \approx \frac{\Delta u}{\Delta t} = \frac{u(t + \Delta t) - u(t)}{\Delta t} \quad (3.24)$$

This approximation can then be rewritten, using the derivative to find the value of  $u$  at the next timestep,  $t + \Delta t$ :

$$u(t + \Delta t) = u(t) + a(t)\Delta t = u(t) + \frac{du}{dt}\Delta t \quad (3.25)$$

This procedure is carried out on our system of equations to yield the following equations for x- and y-velocities:

$$u_x(t + \Delta t) = u_x(t) + \frac{du_x}{dt}\Delta t \quad (3.26)$$

$$u_y(t + \Delta t) = u_y(t) + \frac{du_y}{dt}\Delta t \quad (3.27)$$

Furthermore, using the knowledge that the velocity is the derivative of position, we can write two new equations:

$$x(t + \Delta t) = x(t) + u_x(t)\Delta t = x(t) + \frac{dx}{dt}\Delta t \quad (3.28)$$

$$y(t + \Delta t) = y(t) + u_y(t)\Delta t = y(t) + \frac{dy}{dt}\Delta t \quad (3.29)$$

Which yields four equations that calculate the necessary values to plot and predict a particle's velocities and trajectory (2D location).

[24], [26]

### 3.2.2 Initial conditions

In order to initiate the model described in section 3.2, we need to provide some initial conditions. These conditions include:

- Initial vertical velocity,  $u_{y,0}$
- Initial horizontal velocity,  $u_{x,0}$
- Initial total velocity,  $u_0$
- Initial vertical position,  $y_0$
- Initial horizontal position,  $x_0$

It is assumed that when a particle exits the centrifuge, at  $t=0$ , its velocity will be purely horizontal, i.e

$$u_{y,0} = 0 \quad (3.30)$$

#### Initial horizontal velocity

When a process operator looks at the centrifuge speed, it is given in RPM, or rotations-per-minute. It is therefore beneficial to make use of an equation that can relate RPM to velocity. This can be done by using conservation of angular momentum, and setting the horizontal velocity equal to the tangential velocity in the centrifuge:

$$u_{x,0} = u_t = \omega r \quad (3.31)$$

In this equation,  $\omega$  is given in  $\frac{rad}{s}$ , which means we need to convert rotations-per-minute to radians-per-second. Checking dimensions to calculate:

$$\frac{rotation}{min} \times \frac{min}{s} \times \frac{rad}{rotation} \quad (3.32)$$

Which becomes:

$$RPM \times \frac{1}{60} \times 2\pi \quad (3.33)$$

Which gives us the complete equation:

$$u_{x,0} = RPM \frac{2\pi r}{60} \quad (3.34)$$

#### Initial total velocity

The initial total velocity is calculated using equation 3.23:

$$u_0 = \sqrt{u_{x,0}^2 + u_{y,0}^2} \implies u_0 = u_{x,0} \quad (3.35)$$

## Initial positions

Both the initial x-, and y-positions depend on the centrifuge (prill-bucket) dimensions. The particles exit at some radius  $r(x)$ , which in turn makes them exit at some height  $y$ . Assuming a two-dimensional shape of the bucket as shown below:

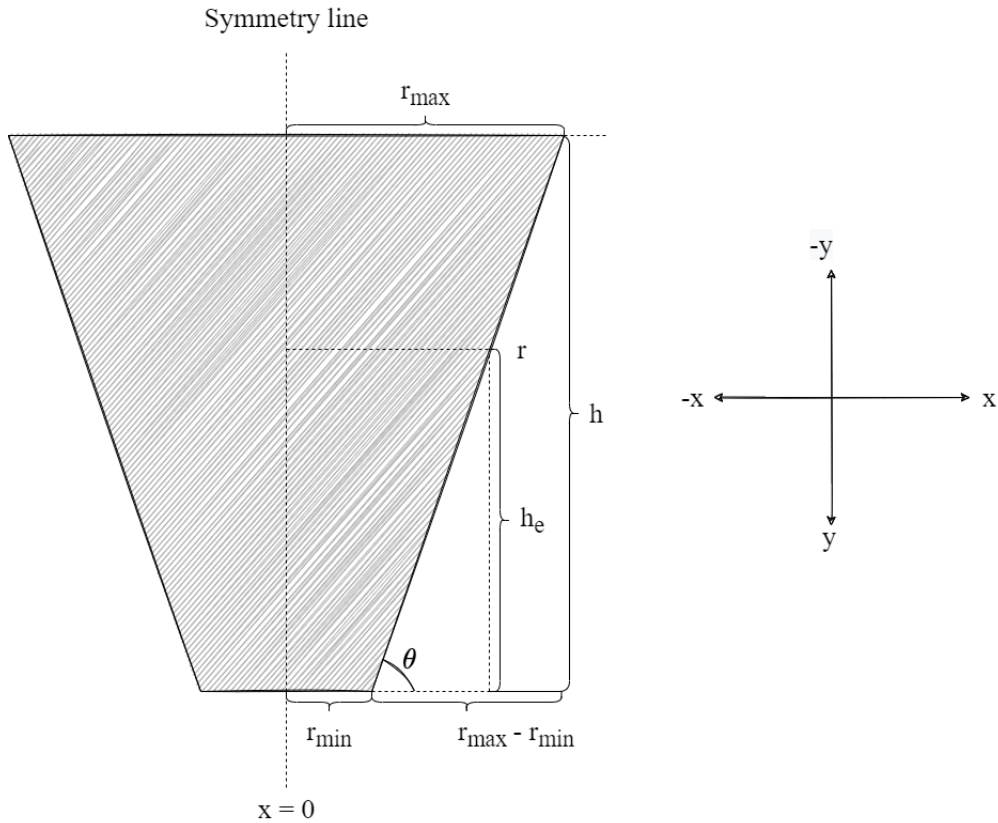


Figure 3.6: Geometric considerations of a centrifuge/prilling bucket in regards to particle ejection.

According to figure 3.6, a particle will be ejected at some some x-position, will will have the same length as  $r$ , and some y-position which will be equivalent to  $h_e$ . Due to the design of the prilling bucket, it is reasonable to specify a discharge radius as an input parameter. This can then be linked to the discharge height,  $h_e$ , by considering a right triangle that uses  $h$  and  $r_{max} - r_{min}$  as legs, and a given centrifuge angle,  $\theta$ .  $\theta$  can be calculated by using the centrifuge dimensions as:

$$\tan \theta = \frac{\textit{opposite}}{\textit{adjacent}} = \frac{h}{r_{max} - r_{min}} \implies \theta = \arctan \frac{h}{r_{max} - r_{min}} \quad (3.36)$$

This angle is the same if we consider a right-triangle composed of  $r - r_{min}$  and  $h_e$  as legs.

We can use  $\theta$  to calculate the unknown discharge height by using:

$$\tan \theta = \frac{h_e}{r - r_{min}} \implies h_e = \tan \theta (r - r_{min}) \quad (3.37)$$

Therefore, we can write the following:

$$x_0 = r \quad (3.38)$$

and

$$y_0 = h_e = \tan \theta (r - r_{min}) \quad (3.39)$$

### Summary of initial conditions

The following subchapters leave us with the following initial conditions, as seen in table 3.2:

Table 3.2: Initial conditions for updated velocity model.

Term	Symbol	Initial condition
Vertical velocity	$u_{y,0}$	0
Horizontal velocity	$u_{x,0}$	$RPM \frac{2\pi r}{60}$
Total velocity	$u_0$	$u_{x,0}$
Vertical position	$y_0$	$\tan \theta (r - r_{min})$
Horizontal position	$x_0$	$r$





## 4 Heat transfer model

Heat transfer is the main model to be developed in this thesis. From the system description, it is apparent that heat transfer occurs in three ways:

- Convection
- Conduction
- Radiation

Heat transfer can be broadly defined as "... *thermal energy in transit due to a spatial temperature difference*" [25]. In a prilling tower, the main heat release occurs between the particle and surrounding fluid, and on the inside of the particle itself. This presents the need for developing a heat model that accounts these methods of heat transfer. The radiation contribution is considered to be small, and constant, and will not be evaluated further in this thesis. This chapter starts off by looking at the two main methods of prilling tower heat transfer, followed by Stokka's approach and culminating in a updated model for predicting the temperature of a particle as a function of its radius,  $r$ , and time,  $t$ .

### 4.1 Convection heat transfer

Convection heat transfer is a term used for heat transfer due to random molecular motion (diffusion) or energy that is transferred by a bulk fluid. In short terms, we can say that convection is *heat transfer through a fluid in the presence of bulk fluid motion* [25]. There are several ways of classifying convection, such as:

- Natural/free convection: Natural movement of fluid due to buoyancy or other effects
- Forced convection: Forced fluid movement, initiated by a pipe etc.
- Internal convection: Fluid flows inside a channel
- External convection: Fluid flows over a surface

There are different ways to mathematically explain these concepts, not all of which are relevant here. Therefore, based on the system description, we will go into detail regarding forced external convection, since this applies to a forced air stream flowing past a particle.

### 4.1.1 Forced, external convection

In order to thoroughly understand the concept of convection, we need to analyze the *boundary layer*. Consider the following system: In figure 4.1, the boundary layer is shown

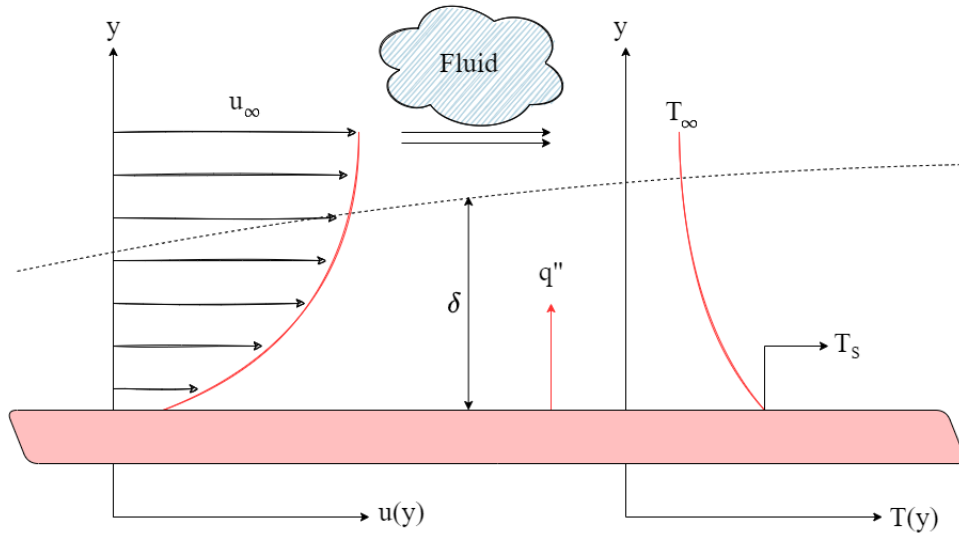


Figure 4.1: Fluid flow past a hot surface. Figure shows the a velocity and temperature profile as a function of distance from the plate ( $y$ ).

as a dashed line. Inside the boundary layer, there are velocity and temperature gradients. At the surface, we have the following conditions:

$$u|_{y=0} = 0, \quad T|_{y=0} = T_s$$

That is, the velocity of the fluid is 0 at the surface, while the temperature of the fluid is  $T_s$ , which can be called the *no-slip condition* and the *no-temperature-jump condition* respectively. At the end of the boundary layer,  $y = \delta$ , the fluid has these properties:

$$u|_{y=\delta} = u_\infty, \quad T|_{y=\delta} = T_\infty$$

In addition, the boundary layer thickness can be described as the point where the temperature,  $T$ , or the velocity,  $u$ , is 0.99 times the free stream value:

$$\delta_t = y|_{T=0.99T_\infty}, \quad \delta_u = y|_{u=0.99u_\infty}$$

The trick here, is that the boundary layer thickness,  $\delta$ , is not constant. It changes depending on the length (or scale) of the surface. The length where the boundary layer is still developing (i.e  $\delta \neq \text{constant}$ ) is dependent on multiple factors, such as geometry or flow conditions. The velocity and thermal boundary layers develop simultaneously, and affect each other, which in turn affects the heat flux from the surface to the fluid. The

heat flux from a surface to a fluid can be described by **Newton's Cooling law**, which states that:

$$q'' = h(T_s - T_\infty) \quad (4.1)$$

where  $h$  is the convection heat transfer coefficient. Using the knowledge of the no-temperature-jump condition, it can be shown that the convection conditions at the surface is equal to the conduction conditions. This can be expressed as:

$$q''_{conv} = q''_{cond} \longrightarrow h(T_s - T_\infty) = -k_{fluid} \left. \frac{\partial T}{\partial y} \right|_{y=0} \longrightarrow h = \frac{-k_{fluid} \frac{\partial T}{\partial y}}{T_s - T_\infty} \quad (4.2)$$

This can be cumbersome to work with, and as for conduction, it is highly beneficial to approach convection problems in a dimensionless manner. Three commonly used parameters in convection heat transfer are the Nusselt number,  $Nu$ , the Prandtl number,  $Pr$  and the Reynolds number,  $Re$ . We will define these parameters in the following section.

[25], [27]

#### 4.1.2 Dimensionless convection parameters

The Nusselt number can be interpreted as the ratio between convective and conductive heat transfer in the fluid boundary layer (visualized in figure 4.1). A higher Nusselt number signifies that convective heat transfer plays a larger role than the conductive heat transfer. Mathematically, it is defined as:

$$Nu = \frac{hL}{k} \quad (4.3)$$

which is analogous to the Biot number in conduction heat transfer. The Prandtl number is a measure of the rate of heat dissipation versus momentum dissipation in a fluid. Some fluids, such as oil, have very high Prandtl numbers. This means that heat dissipates very slow compared to the momentum. Other fluids, such as most gases have a Prandtl number  $\approx 1$  [28] which signifies that both heat and momentum dissipates at the same rate, leaving their respective boundary layer lengths,  $\delta$ , approximately equal. The Prandtl number is defined as:

$$Pr = \frac{\nu}{\alpha} = \frac{\mu C_p}{k} \quad (4.4)$$

Lastly, there is the Reynolds number. The Reynolds number is a measure of the ratio between the inertial forces and the viscous forces. Viscous forces are inter-molecular forces that keep the fluid flowing in an orderly fashion. The ratio between them are used as a marker for labeling flows as either laminar, mixed or turbulent. The exact break-points are different depending on flow geometries. For example, for a fluid flowing in a pipe, the

flow is classified as laminar if  $Re < 2300$  and turbulent if  $Re > 2900$  [25]. Mathematically, the Reynolds number is calculated as:

$$Re = \frac{\rho u_{\infty} L_c}{\mu} \quad (4.5)$$

Reynold created this equation in 1883 after performing an experiment with fluid velocities in a pipe, using a permanganate dye to mark the flow. Some of his own sketches of the experiment can be seen in figure 4.2.

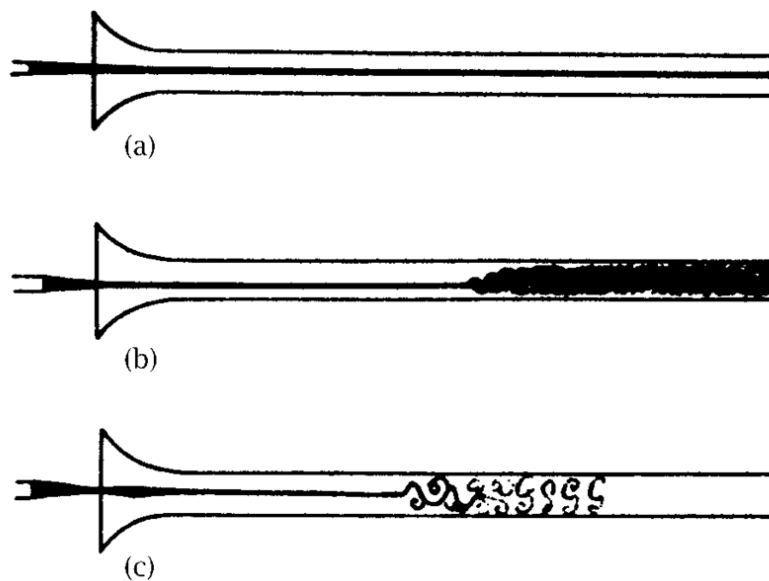


Figure 4.2: Reynolds experiment. The experiment was carried out by adding a dye to water travelling at different speeds. The dye had approximately the same specific density as water, and would therefore follow the fluid-stream pattern. [29]

### 4.1.3 Convection coefficient for a sphere in a fluid stream

Now that some general convection topics, and the most important dimensionless parameters have been introduced, it is time to go into more detail about the convection coefficient,  $h$ . This coefficient is usually what we want to evaluate when looking at convection problems. Once  $h$  has been calculated, everything else falls into place. As stated in the system description, the problem at hand is calculating a convection heat transfer coefficient for a solid sphere in a moving air stream, as shown in figure 4.3. There are several ways of solving this problem, and it has been in the interest of engineers to come up with simple, powerful empirical correlations. This has been achieved through several decades of experiments and statistics, and for a sphere it is common to use the Ranz and Marshall correlation [30]:

$$Nu = 2 + 0.6Re_p^{1/2} Pr^{1/3} \quad (4.6)$$

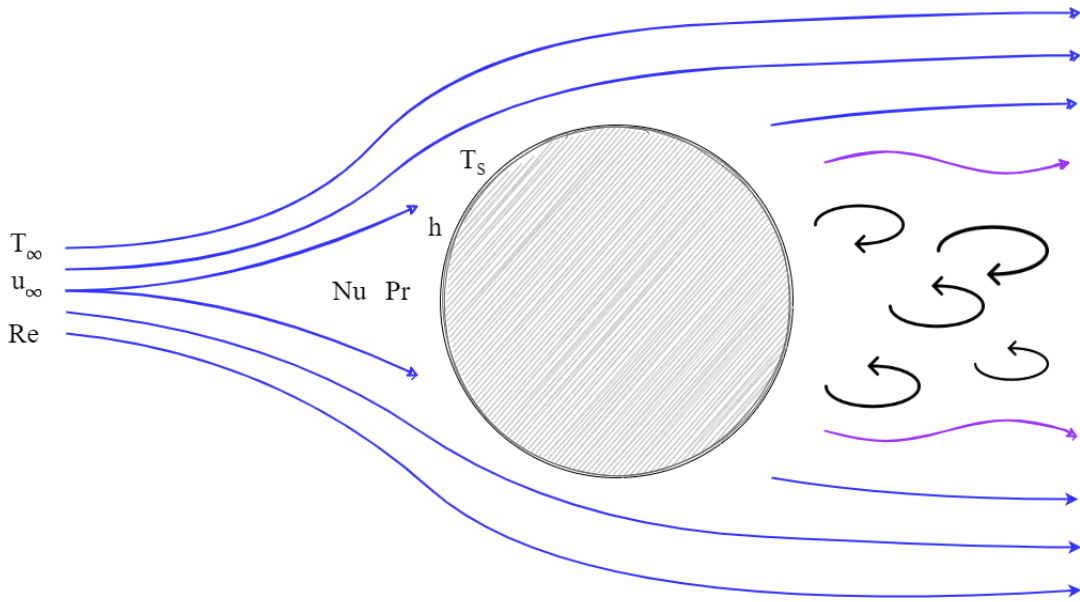


Figure 4.3: Simple sketch of a fluid ( $T_\infty, u_\infty$ ) flowing past a sphere with a set surface temperature ( $T_s$ ) in a specific convective regime ( $Re, Nu, Pr, h$ ). The sketch shows the streamlines of the fluid, and a wake left behind the sphere.

This can be rewritten using the definition of the Nusselt number (equation 4.3) and rearranged to give:

$$h = (2 + 0.6Re_p^{1/2} Pr^{1/3}) \frac{k}{L_c} \quad (4.7)$$

where, for a sphere, it is common to use to particle diameter ( $D_p$ ) as the characteristic length:

$$L_c = D_p$$

This equation is valid when we are dealing with the special case of freely falling liquid drops/particles. The heat transfer from a particle to the surrounding fluid can then be calculated by using equation 4.7 in Newton's cooling law. It is noted that equation 4.1 gives the heat flux, and in order to get the heat transfer *rate*, we need to multiply by the particle surface area:

$$\dot{q} = q'' A_p = h A_p (T_s - T_\infty) \quad (4.8)$$

and, if we want an absolute measure of heat released during a specific time interval, we can multiply by  $\Delta t$  [6]:

$$Q = h A_p (T_s - T_\infty) \Delta t \quad (4.9)$$

### Simplifying assumptions

Now that some general convection equations have been established, we need to look at the specific case, and make the necessary adaptations. These adaptations/assumptions

are:

1. Air flow in the tower is assumed to be laminar and constant
2. Air temperature in the tower is assumed to be constant
3. The particle is considered to be stationary
4. Chemical changes happen instantly, at a specified temperature.

In the previous section, it has been established that:

$$h = f(L, k, \mu, C_p, \rho, u_\infty)$$

The main point of establishing a laminar air-flow with a constant temperature is that the number of Reynolds number calculations are reduced from any number of time-steps (i.e 1000) to one simple calculation. A constant temperature renders the fluids density, viscosity, thermal conductivity and heat capacity constant, so  $Re$ ,  $Pr$ , and  $Nu$  remain constant. Simplifications were therefore made to the convection calculations, in order to keep simulation times from spiraling out of control.

## 4.2 Conduction heat transfer

Conduction heat transfer can be explained as the heat transfer between molecules within a substance. Consider an example where a rod is heated at one end, while other end is being kept at room temperature. This is actually an example that many of us are familiar with. Maybe you needed to disinfect a needle to do an *in-promptu* medical procedure on your self. If you heat the tip of the needle long enough, your fingers will get burnt! This is all due to conduction. The hot tip of the needle contains more energy, and so the particles at that end will vibrate with a higher frequency. This causes *collisions*. These collisions causes an energy transfer between the molecules, and given enough time (and energy), that energy can traverse all along the shaft of the needle and reach the tip of your fingers. This can be illustrated as seen in figure 4.4. Explaining heat conduction is often done

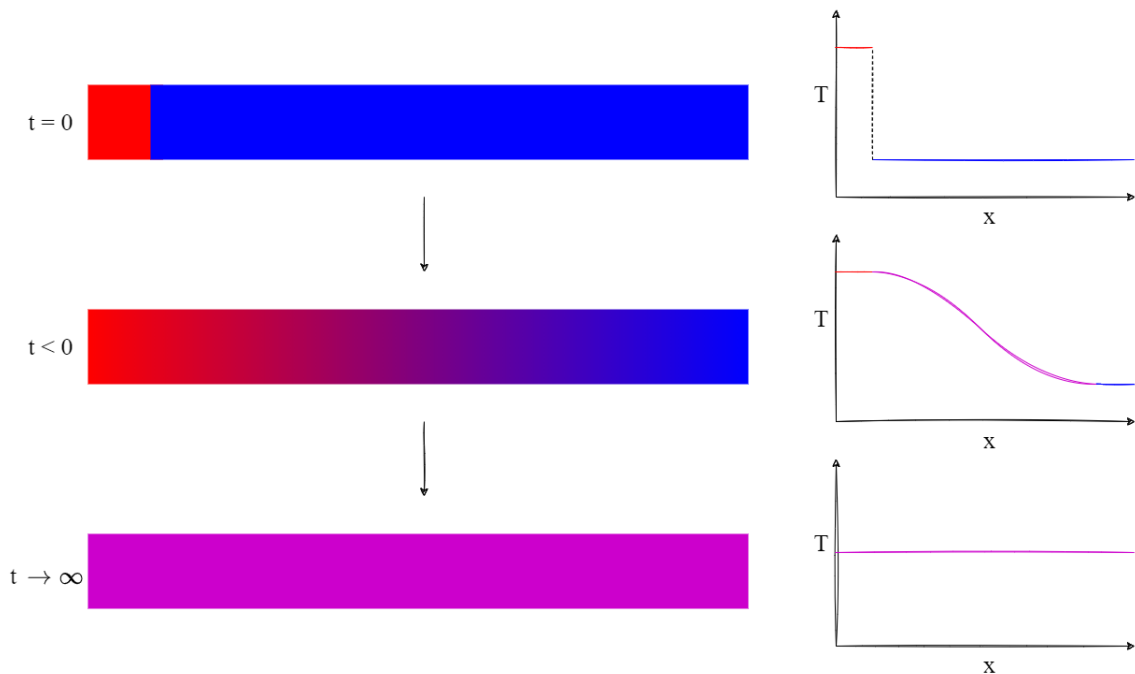


Figure 4.4: Conduction heat transfer in a 1D-rod. Figure shows system sketches and temperature graphs as a function of time and position.

by using the heat diffusion equation. This equation can look quite daunting, but can be combined with all sorts of systems and conditions to fit specific cases. The general heat diffusion equations can be written as:

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) + \dot{q} = \rho c_p \frac{\partial T}{\partial t} \quad (4.10)$$

for Cartesian  $(x, y, z)$  coordinates, and:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left( kr^2 \frac{\partial T}{\partial r} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial}{\partial \phi} \left( k \frac{\partial T}{\partial \phi} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( k \sin \theta \frac{\partial T}{\partial \theta} \right) + \dot{q} = \rho c_p \frac{\partial T}{\partial t} \quad (4.11)$$

for spherical  $(r, \phi, \theta)$  coordinates. Applying the general heat diffusion equation requires significant knowledge of the system that is being considered. Some information is already known, and some can be assumed. For example, considering the rod in figure 4.4, which is 1-dimensional (only x-direction is being considered). Observe how equation 4.10 immediately becomes simpler:

$$\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \dot{q} = \rho c_p \frac{\partial T}{\partial t} \quad (4.12)$$

Of course, this is only valid for a one-dimensional problem, with a heat source or sink ( $\dot{q}$ ) that varies with time. Conduction problems are commonly categorized into two different branches:

- Steady-state conduction
- Transient conduction

Steady-state conduction occurs when a systems characteristic does not change with time. This can be implemented by setting:

$$\frac{\partial T}{\partial t} = 0 \quad (4.13)$$

in equation the heat equation (equation 4.10 or 4.11). This is a common simplification to use, but it is usually not correct in real-world applications. The upside is that steady-state conduction problems are trivial to solve in comparison to transient systems. However, there are several tools that can be applied to solve such time-dependent systems. One of the most common is the lumped-capacitance method (LCM).

[25], [27], [31]

### 4.2.1 Lumped-capacitance method

In order to simplify equation 4.12 without assuming steady-state conditions, then another reasonable assumption could be that the temperature gradient within the particle/solid itself do not change, that is to say that we assume *spatially uniform temperature distribution in the solid* [25]. This can be written mathematically as:

$$T(x, y, z, t) = T(t) \quad (4.14)$$

In cases when considering such spatial uniform temperature distribution, it is also common to assume negligible radiation contributions. Consider a solid suddenly exposed to a



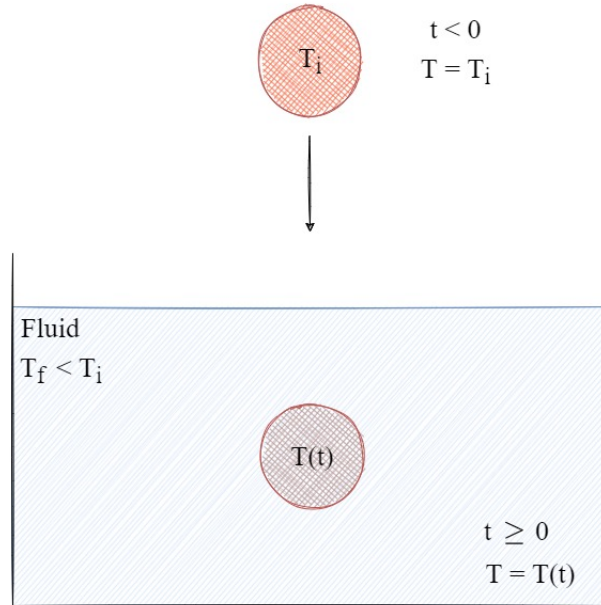


Figure 4.5: Sketch of system where lumped capacitance method is commonly used. Figure shows a solid of uniform initial temperature, suddenly exposed to different conditions.

change in thermal surroundings, as shown in figure 4.5: In such a case, it is possible to write up a general energy balance for the system over the control volume as:

$$\dot{E}_{in} - \dot{E}_{out} + \dot{E}_{gen} = \dot{E}_{acc} \quad (4.15)$$

In addition, if the process being considered is a cooling process without chemical reactions, we can make the following statements:

$$\dot{E}_{in} = 0$$

$$\dot{E}_{gen} = 0$$

Next up is to introduce some new variables that are specific to LCM:

1. Lumped capacitance,  $C_t = mc_p$
2. Convective heat transfer resistance,  $R_t = \frac{1}{hA}$
3. Thermal time constant,  $\tau = R_t C_t$

These variables can be combined with Newton's cooling law (see section 4.1.1) and the energy balance shown in equation 4.15 to produce two equations that can be used to calculate the final temperature of the object, after some specified time  $t_f$ , or the time,  $t_f$ , required to reach some final temperature,  $T_f$ . These equations are shown below.

$$T_f = T_\infty + (T_i - T_\infty)e^{-\frac{t_f}{\tau}} \quad (4.16)$$

$$t_f = \tau \ln \frac{T_i - T_\infty}{T_f - T_\infty} \quad (4.17)$$

Considering the system description given in chapter 2, equation 4.16 and 4.17 are very potent tools, however, they are only valid under certain conditions. These conditions will be explored in the following section.

[25], [27]

### Validity of the lumped-capacitance method

Assuming a uniform temperature distribution in a solid may seem like stretch when it comes to conduction problems, but sometimes this assumptions can produce some very accurate results. Other times, it can lead to a large margin of error. Therefore, it is important to assess the validity of LCM before any important calculations are made, as stated by Engel, Y. A. (2007) [27]:

*”The first step in the application of lumped system analysis is the calculation of the Biot number, and the assessment of the applicability of this approach”*

The Biot number,  $Bi$ , is a measure of an objects resistance to conductive heat transfer compared to its resistance to convective heat transfer. Mathematically, it is defined as:

$$Bi \equiv \frac{hL_c}{k} \quad (4.18)$$

If the conductive resistance is high, i.e  $k$  is low, then it is highly likely that temperature gradients will be observed within the material. Consider figure 4.6:

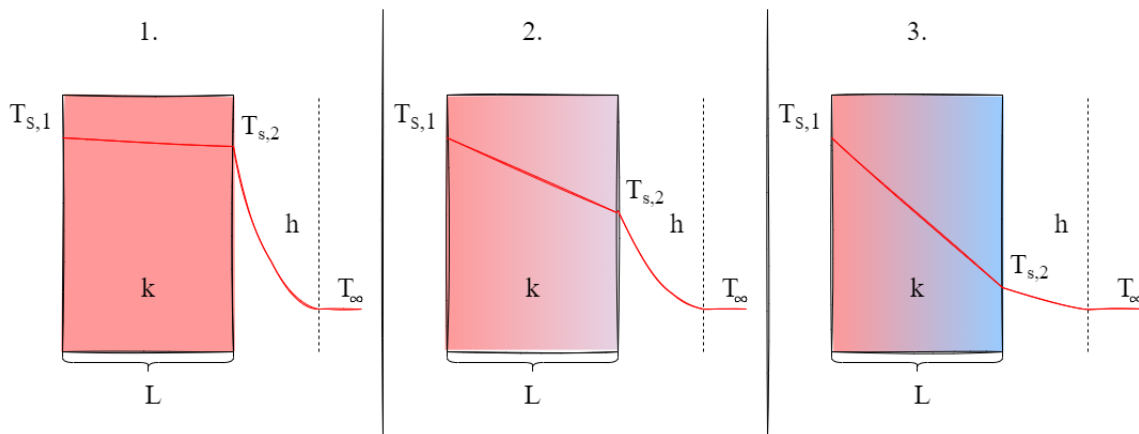


Figure 4.6: Comparison of three different solid systems. System one exhibits a high thermal conductivity, resulting in nearly even surface temperatures. System 2 has a lower thermal conductivity, causing an increase in  $\Delta T_s$ . System 3 has the lowest thermal conductivity and therefore also the highest  $\Delta T_s$ .

Here, three different alternatives are shown. All cases are 1-dimensional heat transfer through a wall with length  $L$ , thermal conductivity  $k$ . The outer surfaces,  $T_{s,2}$  are exposed to convection conditions from a fluid with temperature  $T_\infty$  and convection coefficient,  $h$ . Convection conditions are equal in all three cases, but the thermal conductivity is changed in the following way:

- Case 1: High thermal conductivity
- Case 2: Medium thermal conductivity
- Case 3: Low thermal conductivity

Based on figure 4.6, and the presented characteristic, the following conclusion can be made: *Lower thermal conductivity causes a larger temperature gradient between the surfaces.* This can be expressed mathematically writing the Biot number as:

$$Bi = \frac{hL_c}{k} = \frac{T_{s,1} - T_{s,2}}{T_{s,2} - T_\infty} \implies \Delta T_s = (T_{s,2} - T_\infty) \frac{hL_c}{k} \quad (4.19)$$

Now that a relationship between the conductive and convective forces have been established, alongside an expression that relates the thermal conductivity of a material to the surface gradient, we can apply some validity term to establish **when** LCM is applicable. Generally, it is considered that LCM is applicable if:

$$Bi \leq 0.1$$

Which is accomplished when:

$$T_{s,1} - T_{s,2} \ll T_{s,2} - T_\infty$$

or, in a more general term:

$$hL_c \ll k$$

Looking at figure 4.6, only case 1 would realistically "fulfill" this validity criteria, which means that if we wanted to solve for the temperature as a function of time and position in Case 2 or 3, then we would need an alternate approach. This alternate approach will be discussed in the following section.

[25]

### 4.2.2 Transient conduction

As mentioned in section 4.2, it is often the case that real world system change with time. In addition, if the lumped-capacitance method is invalid, then we need to turn to more rigorous solving of the heat-diffusion equation (equation 4.10 and 4.11). This can be

quite complex, and systematic accurate equation requires both initial conditions (IC's) and boundary conditions (BC's). Consider the following system:

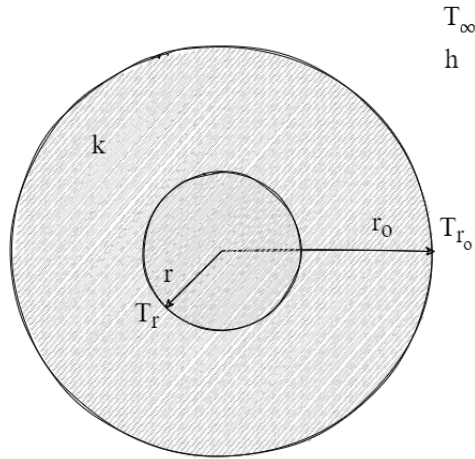


Figure 4.7: Spherical object being exposed to convection conditions.

Figure 4.7 shows a perfectly spherical object, suddenly being exposed to convection conditions. The primary goal is to develop a temperature function of time,  $t$ , and radial position,  $r$ . The object is considered to be at a uniform initial temperature,  $T_i$ .

### Solving transient heat conduction in a sphere

As with the LCM, the first thing to do is to look at the heat-diffusion equation in cylindrical/radial coordinates, which is listed in equation 4.11. It would be helpful to visualize how radial coordinates are considered, this is shown in the figure below. Observing figure 4.8, and using the qualities of a sphere, it is possible to represent transient heat conduction in a sphere as a 1-dimensional problem, only considering the  $r$  coordinate. This is under the assumption that the convection boundaries are equal on all sides of the sphere. This assumption gives rise to the following simplified heat-diffusion equation:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left( kr^2 \frac{\partial T}{\partial r} \right) + \dot{q} = \rho c_p \frac{\partial T}{\partial t} \quad (4.20)$$

Assuming no heat generation, and knowing that the thermal diffusivity,  $\alpha$ , is defined as:

$$\alpha = \frac{k}{\rho c_p}$$

gives us:

$$\frac{\partial T}{\partial t} = \frac{\alpha}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial T}{\partial r} \right) \quad (4.21)$$

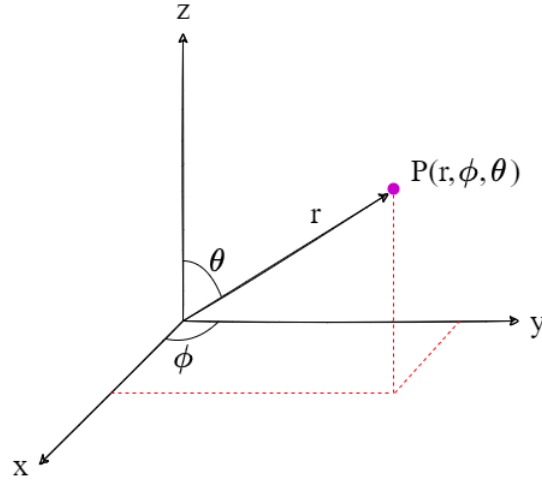


Figure 4.8: Representing a point in space using spherical coordinates. An arbitrary point P is shown (purple dot) using  $r$ ,  $\theta$  and  $\phi$  coordinates instead of the cartesian  $x$ ,  $y$  and  $z$ .

Which is a simpler starting point compared to equation 4.11. What is now obtained is a linear partial differential equation (PDE). In order to solve this analytically, initial- and boundary conditions have to be set. From the introduction to this problem, it is established that at  $t = 0$ , the temperature is uniform with a temperature of  $T_i$  [32]. This can be written as:

$$T(r, 0) = T_i$$

The surface heat flux is governed by Newton's cooling law, which will be set as the boundary condition. This is expressed as:

$$k \frac{\partial T}{\partial r} \Big|_{r=r_o} = h(T_\infty - T_s)$$

After the initial- and boundary conditions have been set, some dimensionless parameters are defined in order to obtain a *universal* solution. These parameters include dimensionless temperature, radius and time:

$$\theta^* = \frac{T - T_\infty}{T_i - T_\infty}, \quad r^* = \frac{r}{r_o}, \quad Fo = \frac{\alpha t}{r^2} \quad (4.22)$$

where we also introduce a new dimensionless boundary condition:

$$\frac{\partial \theta^*}{\partial r^*} \Big|_{r^*=1} = Bi \theta_s^* \quad (4.23)$$

The major benefit of this non-dimensionalizing is that we transform the equation, which leads to a large reduction in variables that needs discrete consideration:

$$T(r, t, T_i, T_\infty, L, k, \alpha, h) \longrightarrow \theta^* = f(x^*, Fo, Bi) \quad (4.24)$$

Bergman, L. T. and Levine, A. S. (2018) [25] state the following:

*... the considerable advantage associated with casting the problem in dimensionless form becomes apparent. The dimensionless equation implies that for a prescribed geometry, the transient temperature distribution is a universal function of  $x^*$ ,  $Fo$ , and  $Bi$ . That is, the dimensionless solution has a prescribed form that does not depend on the particular value of  $T_i$ ,  $T_\infty$ ,  $L$ ,  $k$ ,  $\alpha$ , or  $h$ .*

Now, the problem becomes to find the analytical function that describes the relationship between  $\theta^*$  and  $x_*$ ,  $Fo$  and  $Bi$ . These relationships have been evaluated in multiple textbooks and papers, for several different geometries and are given as a *fourier series*. For a sphere, it is shown that ([32]–[35]):

$$\theta^* = \sum_{n=1}^{\infty} C_n \exp(-\zeta_n^2 Fo) \frac{1}{\zeta_n r^*} \sin(\zeta_n r^*) \quad (4.25)$$

where

$$C_n = \frac{4[\sin(\zeta_n) - \zeta_n \cos(\zeta_n)]}{2\zeta_n - \sin(2\zeta_n)} \quad (4.26)$$

and  $\zeta_n$  are the positive roots of the equation given by:

$$1 - \zeta_n \cot(\zeta_n) = Bi \quad (4.27)$$

A graph of equation 4.27 is shown in figure 4.9.

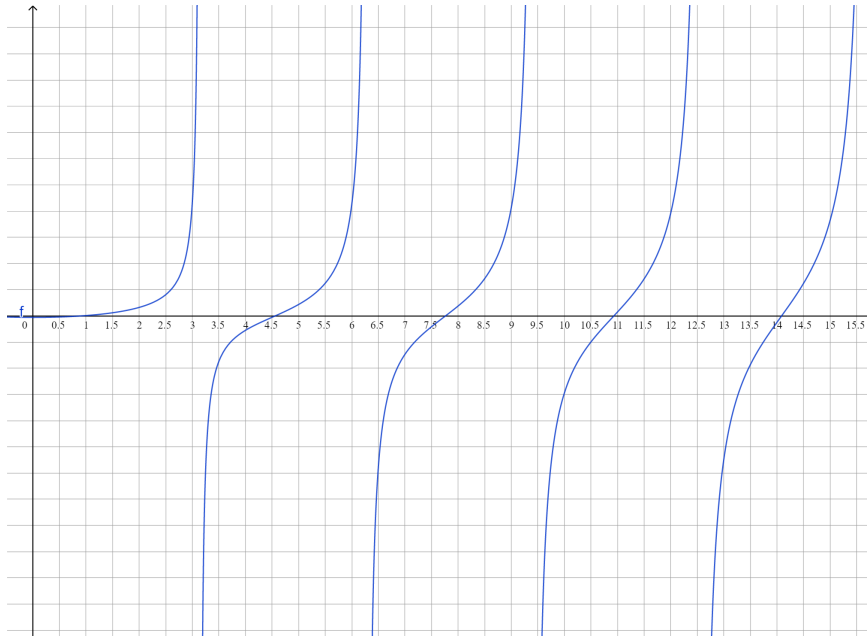


Figure 4.9: Graph showing the equation  $1 - x \cot x - 0.3 = 0$ . This is equivalent to solving equation 4.27 using  $Bi = 0.3$ . The positive roots of this equation ( $\zeta_n$ ) are used in equation 4.25 and 4.26. This figure was made using GeoGebra’s online graphing tool.

Once the root(s), have been found, the temperature is calculated by modifying equation 4.22:

$$\theta^* = \frac{T - T_\infty}{T_i - T_\infty} \implies T = T_\infty + \theta^*(T_i - T_\infty) \quad (4.28)$$

In addition to finding the temperature at a given point in time and space, it is also possible to use these dimensionless parameters and the Biot roots to calculate how much heat has been lost from the object in question [25], [36]. In order make this calculation, we first introduce the *maximum amount of heat transfer that can occur, if the process was continued until  $t = \infty$* :

$$Q_o = \rho_p c_p V_p (T_i - T_\infty) \quad (4.29)$$

The actual heat transfer and the theoretical maximum is connected in the following way:

$$\frac{Q}{Q_o} = 1 - \frac{3\theta_o^*}{\zeta_1^3} [\sin(\zeta_1) - \zeta_1 \cos(\zeta_1)] \quad (4.30)$$

Where  $\frac{Q}{Q_o}$  represents a fraction of maximum heat loss, up to a given time,  $t$ . The center temperature value can be determined from:

$$\theta_o^* = C_1 \exp(-\zeta_1^2 Fo) \quad (4.31)$$

Once the right side of equation 4.30 has been calculated through equation 4.31, and  $Q_o$  has been calculated from equation 4.29, then  $Q$  can be found.

[25]

Linear PDE's have the property that summing together all different solutions, is itself a solution [37]. This is used to make the analytic approximation more closely resemble an *exact* solution. In certain situations, it is sufficient to solve the equations only once, which would require the first root of equation 4.27,  $\zeta_1$ . This is called the *one-term approximation*. In this thesis, the one-term-approximation is used to calculate the heat-loss fraction,  $\frac{Q}{Q_o}$ , but not temperatures. OTA's are not always accurate enough [38], and more terms are usually required for improved accuracy, as seen in figure 4.10. This makes computational programming an excellent tool for evaluating such systems. Some simple for-loops allow the user to calculate and summarise a number of series, defined by how many roots are available, which luckily, can also be calculated using a pre-defined program.

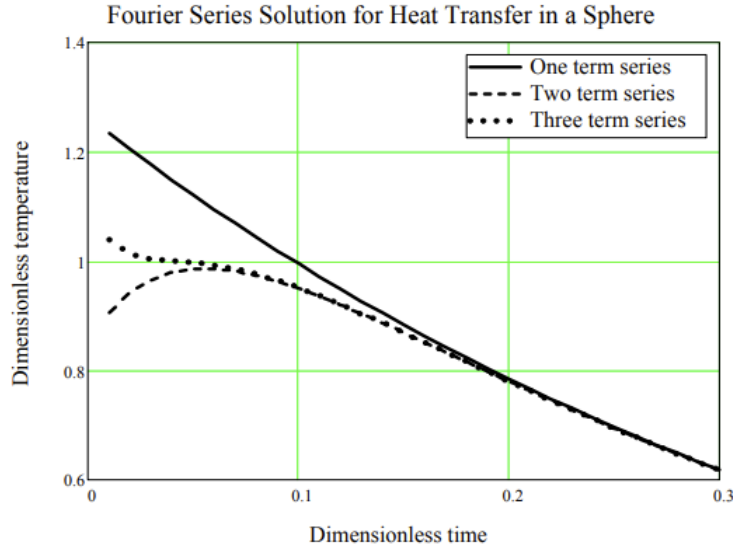


Figure 4.10: Comparison between a one term, two term and three term Fourier series solution, where the object being investigated is a sphere, suddenly subjected to convection conditions. [38]

### 4.2.3 Critical radius

Now that the analytical conduction method has been explained, it is clear that there is a way to calculate the temperature as a function of time, and radius. For NPK products produced in NPK-3, one of the main issues is particle deformation upon hitting the ground. Stokka, P. (1989) presented that for a prill to survive the fall, a minimum of 80% has to be solidified [6]. In this case, it would of interest to examine what happens at this border. It is important to mention that because solidification happens from the surface, towards the core, if we are starting at  $r = 0$  (core), then  $r_{critical}$  will occur when 20% of the volume has been covered. This is illustrated in figure 4.11 In order to calculate the exact critical radius for a given particle size, we start by using the general formula for the volume of a sphere:

$$V_{sphere} = \frac{4}{3}\pi r^3 \quad (4.32)$$

The critical radius is defined as the radius where 20% of the particle volume (from the inside-out) has been occupied, hence:

$$0.2V_p = \frac{4}{3}\pi r_{crit}^3 \quad (4.33)$$

The equation is rearranged to give:

$$r_{crit} = \sqrt[3]{0.2V_p \frac{3}{4\pi}} \quad (4.34)$$



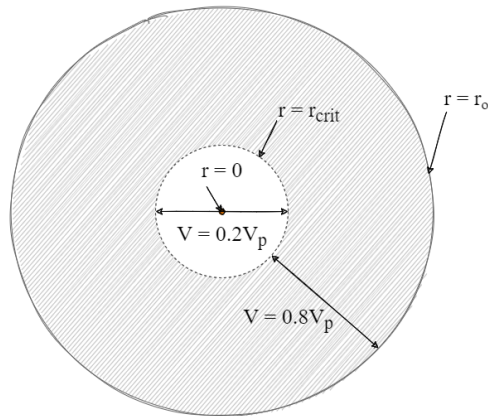


Figure 4.11: Sketch of particle showing the core radius,  $r = 0$ , critical radius,  $r = r_{crit}$  and outer (surface) radius,  $r = r_o$ . The sketch also shows how the critical radius is defined: the radius where 20% of the volume of the sphere is contained.

The conduction model can be used to plot the temperature development at this radius, which will give an implicit idea of the degree of crystallization of the particle. In addition to looking at the critical radius, the core temperature and surface temperatures are also quite interesting, which can be modeled by running the conduction model with radii  $r = 0$  for the core, and  $r = r_o$  for the surface. However, to continue with the model, we need to account for phase changes. This topic will be described in the next chapter.



## 5 Phase changes

Now that the governing heat equations have been presented, it is time to look at what those consequences those equations have. In this report, the topic of evaluation is the cooling of a particle in a prilling tower. There have been several studies on heat-and-mass transfer balances in prilling towers, many of which have been used as sources in this report. However, they all share a common property; they deal with a particle made up of a single component. This is not the case for NPK, as it is a compound fertilizer. Not only is the chemical mixture rather complex, but NPK products differ from each other, depending on N/P ratio, or other product specifications. This complicated chemical puzzle makes modeling the phase-change behavior of the particle rather difficult and cumbersome. In this chapter, the general concept of phase changes will be introduced, as well as an attempted solution to complying with the complex phase changes that occur in an NPK prill during cooling.

### 5.1 Introduction to phase-changes

It is important to start out by clarifying the differences between *phase* and *state*:

- Phase: specific state of matter that is uniform throughout in composition and physical state
- State: one of the distinct forms matter can exist in (solid, liquid, gas and plasma)

This is important because describing a phase is a more strict characterization than describing a state. For example, crystalline ammonium nitrate, ( $\text{NH}_4\text{NO}_3$ ), commonly referred to as AN, can be characterized as a solid. This would describe the chemical's state. If we wanted to describe its phase however, we would need to define *what kind* of crystalline structure the compound currently exhibits. Figure 5.1 is an example of two common crystal phases that AN exhibit [39]. Although this makes it more complicated, the topics will be explained further. Starting out with states of matter.

#### 5.1.1 States of matter

States of matter are familiar to everyone. Every human breathes in gases, and regularly consume both liquids and solids. The transition between these states should also be quite

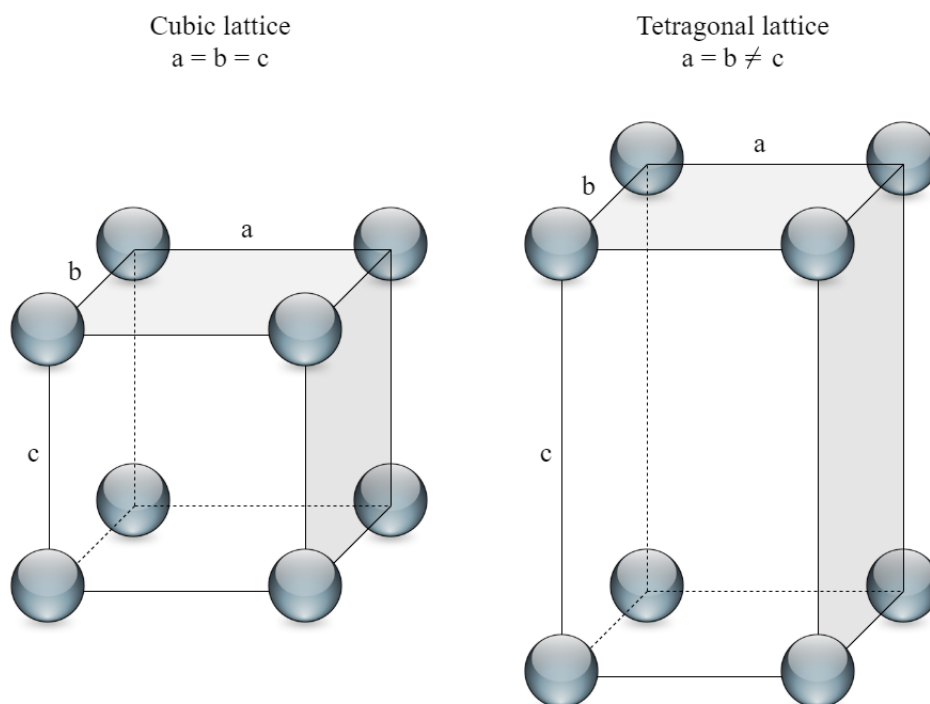


Figure 5.1: Sketch of two common crystalline structures of AN. The left sketch shows a cubic lattice structure, which commonly occurs at temperatures ranging from  $\sim 170$  to  $125$  °C. The right sketch shows a tetragonal structure, which occurs at a temperature range between  $125$  to  $84$  °C.

familiar. Boiling water gives rise to steam, which can again be cooled down, and create droplets. The thermodynamics between such phases for singular compounds (such as water) are well documented. For a substance to change phase, it requires heat transfer. This heat transfer can add or remove heat, depending on the purpose. When dealing with phase changes, it is common to refer to two distinct different types of heat involved in the transfer [13]:

- Sensible heat
- Latent heat

Sensible heat is the heat associated with *temperature change*. If heat is added or removed to a system, and the system's temperature undergoes a change, we say that the heat that has been transferred is sensible heat. Latent heat is heat associated with *phase change*. When a substance reaches a temperature where it can progress from one state to another, the temperature does not change, instead, all the heat transferred is involved in phase transition. This can be seen in figure 5.2. Let us inspect figure 5.2 in closer detail. Assume we have ice, at a temperature of  $-20$  °C, which is heated to  $130$ °C. By continuously adding heat, we will eventually heat up the ice to  $0$  °C. Adding more heat will then cause the ice to transition into liquid water, still at  $0$  °C. If we keep adding heat after all the ice

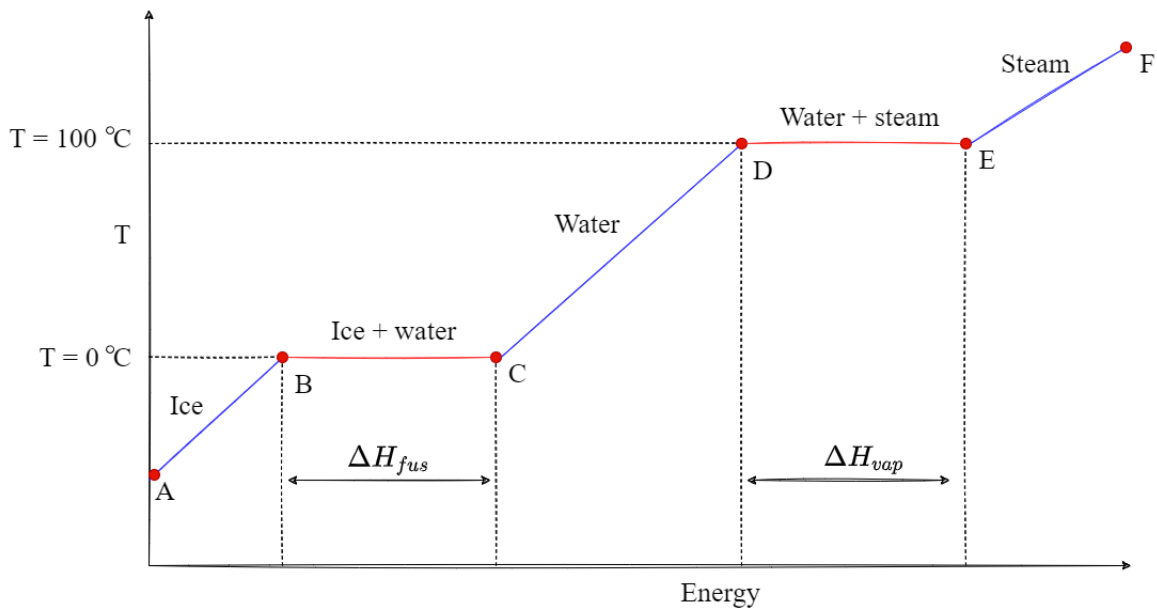


Figure 5.2: State transition of water, represented in a temperature-energy diagram. Blue lines show sensible heat which changes temperature, while the red lines show latent heat, altering the state of the substance.

has melted, the water will again begin to increase in temperature. This increase keeps going until we reach the boiling point at 100 °C. At this point, another phase transition will occur, this time from water to steam. Once all the water has been evaporated, the temperature will once again start increasing. This can be summarized as:

- A  $\rightarrow$  B:  $\Delta T = -20^{\circ}\text{C} \implies$  No change in state, sensible heat
- B  $\rightarrow$  C:  $\Delta T = 0^{\circ}\text{C} \implies$  State change, latent heat
- C  $\rightarrow$  D:  $\Delta T = 100^{\circ}\text{C} \implies$  No change in state, sensible heat
- D  $\rightarrow$  E:  $\Delta T = 0^{\circ}\text{C} \implies$  State change, latent heat
- E  $\rightarrow$  F:  $\Delta T = 30^{\circ}\text{C} \implies$  No change in state, sensible heat

It is common to characterize the latent heat of a compound corresponding to which state it changes to. The two most common are the latent heat of fusion,  $\Delta H_{fus}$ , and latent heat of vaporization,  $\Delta H_{vap}$ . These are connected to the change in enthalpy of a substance during a state change from solid to liquid, and liquid to gaseous respectively.

[13]

### 5.1.2 Heat capacities

Now that we have connected phase changes to changes in temperature, it would be useful to quantify how we can measure what amount of heat is required to bring a substance from one point to another. When two objects of different temperatures are brought into contact (at constant pressure), thermodynamics tells us that heat can be calculated as:

$$Q = mC_p\Delta T = m\Delta H \quad (5.1)$$

And for the heat transferred during phase change:

$$Q = m\Delta H_{\text{vap, fus}} \quad (5.2)$$

[40]

If we were to calculate the total amount of heat that we need to transfer in order for the water in figure 5.2 to go from -20 to 130 °C, we would only need these two equations. Equation 5.1 would be used for the sensible phases and equation 5.2 for latent phases. For single substances, heat capacities and enthalpy curves are easily accessed, which would allow for the aforementioned equations to be used without much issues. NPK products however, are very complex. This makes assessing the heat capacities and enthalpy changes rather difficult. Considering the nature of NPK products, there are different components crystallizing at different temperatures, and they are all intertwined. One way to establish the connections between the different compounds in a product is by using a *phase diagram*. Stokka, P. (1986) writes that if such connections are established in a phase diagram, it can be used as a very important tool to obtain:

- *Temperature where crystallization begins*
- *Temperature where crystallization ends*
- *The composition of a finished product, and their weight fractions*
- *What compounds that precipitate/crystallize at any temperature*
- *Weight fractions between melt-phase (liquid) and solid-phase at any temperature*
- *Weight fractions between uncrystallized compounds*
- *Composition of the melt-phase from start of crystallization to total crystallization*

or, put more simply: *Phase diagrams represent large amounts of information in a very compact form* [20]. Phase diagrams can be given as a function of temperature and pressure for a single compound, or a function of temperature for binary (two compounds) and ternary (three compounds) diagrams. Such diagrams can further be classified into two separate parts; solid-liquid systems and gas-liquid systems. In this thesis, the main focus is on a system that contains solids and liquid, and these systems will be explained further.

## 5.2 Solid-liquid phase diagrams

Consider figure 5.3, showing a general solid-liquid phase diagram of a *eutectic* binary mixture. A eutectic mixture is a mixture where the liquid phase components are completely miscible, whereas the solid components are not [41].

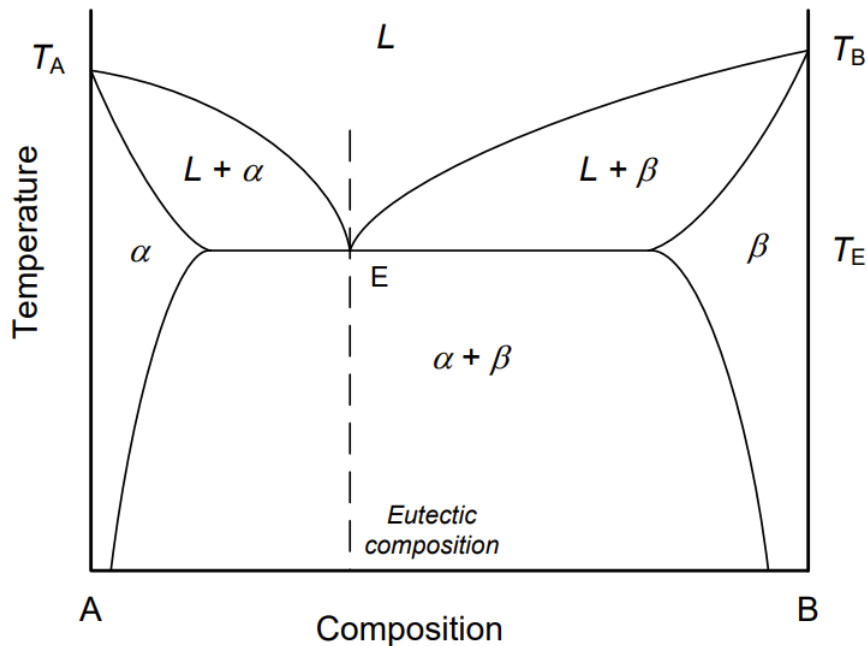


Figure 5.3: Diagram of a general binary eutectic mixture. [42]

This figure can be used to find out what kind of liquid/solid composition a specific mix of A and B have at a given temperature. An important point worth mentioning here is the *eutectic point*. This is the point where the liquid (containing miscible A and B) are in equilibrium with their solid counterparts,  $\alpha$  and  $\beta$ , respectively. The temperature of the mixture at this point,  $T_E$  is known as the eutectic temperature, and is the lowest temperature at which the liquid phase is stable. Two other concepts of interest are the terms *liquidus* and *solidus*. These terms refer to the lines in the diagram that show the composition dependence between the liquid (L) and semi-solid transition (L +  $\alpha$ ) and semi-solid (L +  $\alpha$ ) and solid transition ( $\alpha$ ), respectively. Based on these concepts, it is apparent that a phase diagram can provide very useful information about binary mixtures. But what about more complex mixtures?

[42]

## 5.2.1 Ternary phase diagrams

Very often, real world processes contain more than one or two compounds. Often, chemical process data regarding crystallization and solidification behaviour in a complex process are shown as several different, co-dependent, *ternary* phase diagrams. Ternary diagrams

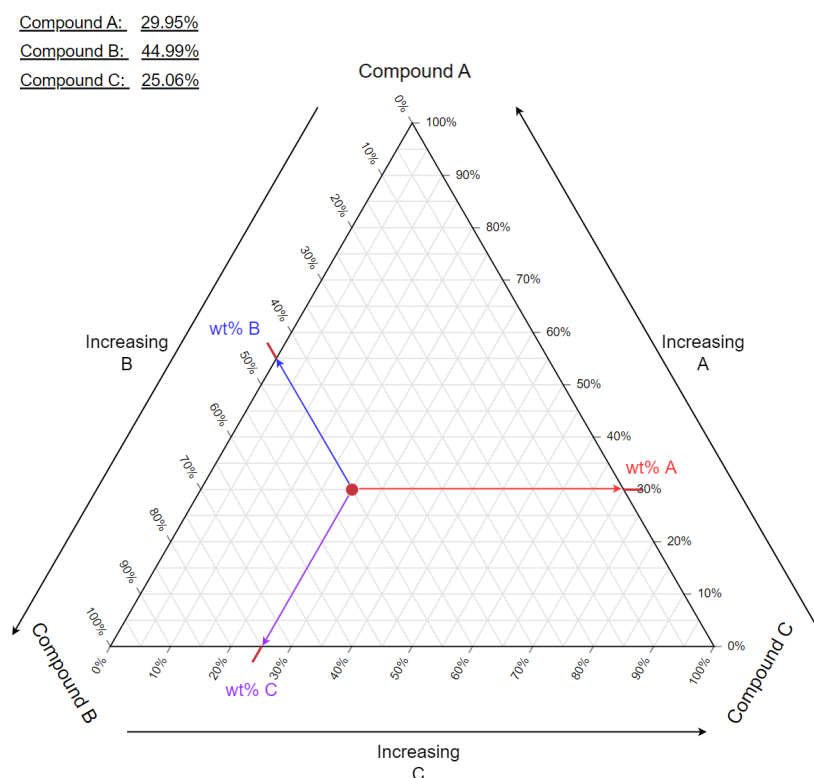


Figure 5.4: General ternary diagram. Figure shows calculation of weight-% at a single point. Layout created using TernaryPlot.com.

are shaped as triangles, with each corner representing 0 or 100% of ... *something*, usually weight percentages (when assessing chemical processes). A general layout is shown in figure 5.4. In a simple ternary diagram, the percentage of each compound can be read by drawing a line from the point of interest to the lower value, following the triangle. Such figures can also be enhanced by plotting in the different phase transitions and different temperature isotherms. These isotherms and graphs are all dependent on how the three compounds behave on their own, and together. This means that there is no *one-ternary-phase-diagram* to fit them all. Each diagram has to be specifically designed based on chemical and thermodynamic data. One such design will be the topic of the next section.

[43]



### 5.2.2 Crystallization of NPK fertilizer

In the NPK prilling process, ammonium nitrate ( $\text{NH}_4\text{NO}_3$ , abbrev. AN) and mono-ammonium phosphate ( $\text{NH}_4\text{H}_2\text{PO}_4$ , abbrev. MAP) are the two most important components. These two components account for most of the crystallization, and are therefore extremely important to evaluate. In addition, it is essential to consider moisture content in the form of  $\text{H}_2\text{O}$ . Water has been found to drastically alter the behaviour of crystallization processes, and needs to be taken into consideration. This gives rise to AN-MAP-Water phase diagram. Stokka, P. (1986) [20] explains that:

*For an evaporated NP-liquor with  $\text{pH} (1 + 9) < 5.0$ , the ternary system AN-MAP-Water describes the melt.*

Following, this becomes an extremely important tool in assessing what goes on inside a particle during cooling in the tower. The AN-MAP-Water phase diagram that is being used in this report is a collaboration between P. Stokka, O. Borge and S. E. Kjekshus, and was made digital by the author of this thesis. The phase diagram basis is given in Appendix C. Based on the composition of AN, MAP and water, the fraction between crystallized and liquid components change accordingly. This can be interpreted for the different products to be used in a model that calculates a heat capacity for use in the heat transfer model or to assess the degrees of crystallization, given a certain temperature or temperature interval. This is all well and good for evaluating an evaporated NP-liquor, but the system becomes more complicated when assessing additives and salt compositions [44]. As mentioned in chapter 2, most additives can be considered to be inert [12]. One compound that is proven to influence the crystallization, however, is potassium chloride (KCl) [21]. The amount of KCl that is dissolved will be a function of the melts temperature and composition. The dissolved KCl will present itself as  $\text{K}^+$  and  $\text{Cl}^-$  ions, alongside ammonium- ( $\text{NH}_4^+$ ) and nitrate-ions ( $\text{NO}_3^-$ ). This means that instead of considering a ternary phase diagram composed of:

AN – MAP – Water

there would be a need to consider a "six-sided" phase diagram composed of:

$\text{NH}_4\text{NO}_3 - \text{NH}_4\text{Cl} - \text{KNO}_3 - \text{KCl} - \text{MAP} - \text{Water}$

Such a system would be very complicated, and rely on several different factors, such as initial composition, temperature and residence time. Thermodynamic models containing this information are in many cases proprietary or difficult to obtain.

[21]

Programs such as FactSage or Aspen could potentially be used to produce such data, but was considered too time-consuming for the purpose of this thesis. Therefore, the simplified AN-MAP-Water phase diagram will be used to assess the heat capacities and crystallization behavior of all compounds.

### 5.3 Stokkas heat capacity model

Optimally, the information conveyed by the phase diagrams would be automatically read, and its data treated in real time in a computational model. This is unfortunately way beyond the technical skills of the author. Luckily, Stokka, P. (1989) developed a way of turning the phase changes into a way of describing the heat capacity of the system. To do this, Stokka uses the principles of a steep crystallization curve. This kind of crystallization curve is characterized by a large fraction of the crystallization happening in a tiny temperature interval. Such a curve is shown in figure 5.5.

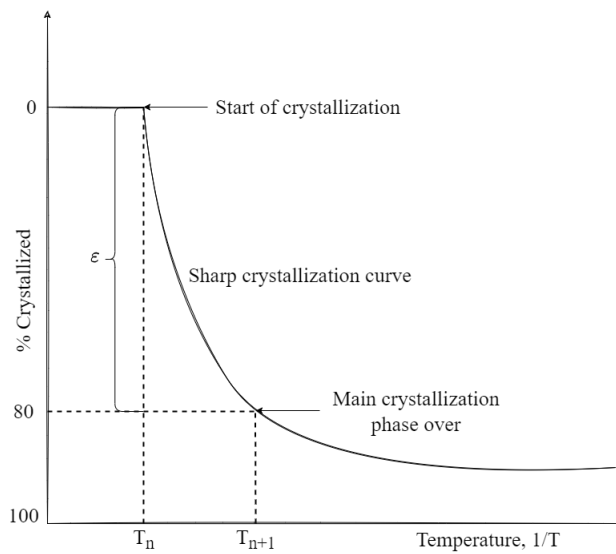


Figure 5.5: Characteristics of a steep crystallization curve.  $T_n$  and  $T_{n+1}$  are temperatures at the start and end of a specific interval.  $\epsilon$  is the degree of crystallization within that interval.

Instead of treating the chemical system as continuously changing, it is helpful to approximate it as constant within certain temperature intervals:

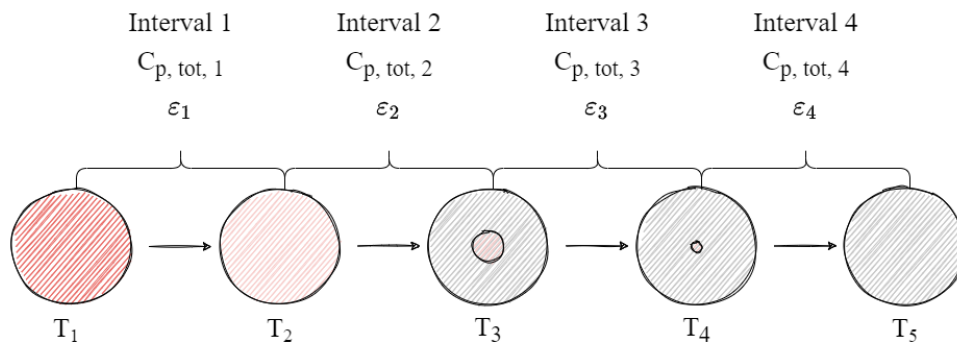


Figure 5.6: Breakdown of heat capacity intervals and degrees of crystallization for a particle.

The heat that is released from the particle in that interval, is therefore only a function of the heat capacity of the components in that interval. A figure depicting the breakdown of these intervals is shown in figure 5.6 on page 66. The number of intervals are equal for all products that are analyzed. The difference comes from the temperature breakpoints, and the heat capacities within those intervals. The heat intervals can be defined as:

Table 5.1: Temperature breakpoints used in determining the intervals, and subsequent heat capacities for different NPK products.

Symbol	Value	Interval	Degree of crystallization
$T_1$	Mixer temperature		
		1	$\epsilon_1$
$T_2$	Start of main crystallization		
		2	$\epsilon_2$
$T_3$	Start of rest crystallization		
		3	$\epsilon_3$
$T_4$	End of rest crystallization		
		4	$\epsilon_4$
$T_5$	End temperature		

The heat capacity itself is calculated by using the following equation [6]:

$$C_{p_{tot,int}} = \frac{m_{cr}\epsilon Q_{cr}}{m_{tot}(T_n - T_{n-1})} + (C_{p_s}\bar{x}_s + C_{p_l}\bar{x}_l) \quad (5.3)$$

Equation 5.3 considers a total heat capacity by taking both the heat of crystallization, and solid/liquid heat capacities of the components. The equation might be easier to understand if it is broken up into:

$$C_{p_{tot,int}} = C_{p_{cr}} + C_{p_{s+l}} \quad (5.4)$$

where the heat capacity of the crystallizing mass can be written in the form of equation 5.1 :

$$\Delta H = C_p \Delta T \implies C_p = \frac{\Delta H}{\Delta T} \quad (5.5)$$

Where the enthalpy change during phase transition can be written as (equation and 5.2):

$$\Delta H_{fus,vap} = \frac{Q_{cr}}{m_{tot}} \quad (5.6)$$

Combination gives:

$$C_p = \frac{Q_{cr}}{m_{tot}\Delta T} \quad (5.7)$$

However, in our case, it is not the total mass that is crystallized, only a fraction. Therefore, the heat capacity for the fraction of mass that is actually crystallized can be calculated by multiplying the equation by the mass available for crystallization ( $m_{cr}$ ), and the degree of crystallization (fraction of crystallization/kg,  $\epsilon$ ):

$$C_{p_{cr}} = \frac{Q_{cr}}{m_{tot}\Delta T} m_{cr} \epsilon \quad (5.8)$$

Which is identical to the term used in equation 5.3. The heat capacities of the solid and liquid compounds that do not undergo phase changes are calculated using a general weighted formula:

$$X_{A+B+C} = X_A \frac{A}{A+B+C} + X_B \frac{B}{A+B+C} + X_C \frac{C}{A+B+C} \quad (5.9)$$

In order to use equation 5.3 for different products, the following information needs to be known:

- Temperature values for determining the intervals ( $T_1$  to  $T_5$  in table 5.1)
- Heat of crystallization (related to  $\Delta H_{fus}$ )
- Heat capacities of all solid and liquid components
- Mass fractions of solids and liquids in each interval

### 5.3.1 Application to the conduction model

The calculated heat capacity might seem straight-forward to apply, since it is only used when calculating the Fourier number,  $Fo$ . However, the model automatically needs to account for the interval changes, which means that the calculated temperatures needs to be checked, and if the temperature has reached the start of a new interval, then the heat capacity needs to be changed before the calculations continue. A simple flowchart of the conduction model calculations can be seen in figure 5.7 below.

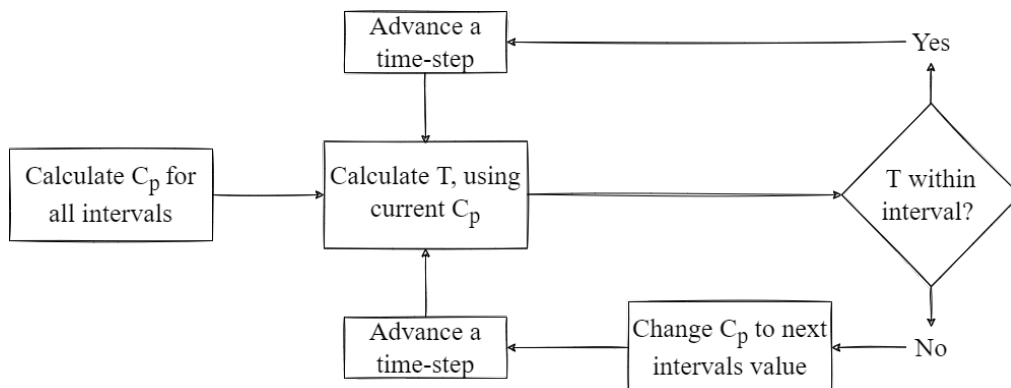


Figure 5.7: Flowchart depicting the general calculation procedure, accounting for Stokka's heat capacity model.

## 6 Results

The following chapter aims to provide the results of the heat- and mass transfer model. The core model itself is created using parameters given by Stokka, P. (1989) for product type NPK 21-4-10, using data obtained 05/06-84 [6]. The model was then tested using a set of parameters with ranging values, to see how the temperature values responded. Following this, the model was adapted for use with other NPK products, with a focus on four specified focus grades. The velocities and particle trajectories of the updated model will be shown first, followed by the base composite Ytterbø model, compared to Stokka's model. After that, the results from the sensitivity analysis will be presented. Lastly, the results from the adapted model will be shown.

### 6.1 Velocity model

A standalone version of the velocity model was run using particle diameter as the main variable. The parameters for the model can be seen in table 6.1:

Table 6.1: Parameters and values used for assessing a standalone velocity model.

Parameter	Symbol	Value	Unit
Particle density	$\rho_p$	1754	$\frac{\text{kg}}{\text{m}^3}$
Air density	$\rho_{air}$	1.19	$\frac{\text{kg}}{\text{m}^3}$
Tower height	h	43.4	m
Centrifuge speed	RPM	390	$\frac{1}{\text{min}}$
Inner centrifuge radius	$r_i$	0.265	m
Ejection radius	r	0.45	m
Prill bucket angle	$\theta$	20	°
Drag coefficient	$C_d$	0.44	-
Time	t	20	s
Step-time	dt	0.01	s

The results are shown in three forms:

1. Prill trajectories, x- and y-coordinates
2. Vertical position of prill as a function of time
3. Velocity profiles

### 6.1.1 Prill trajectories

The prill trajectories results are plotted as distance from tower top and distance from tower bottom. In this section, the lowest point of the prilling bucket is considered to be  $y = 0$ . If a particle is ejected in the middle of the prill bucket, it will start at  $y > 0$ . The bottom of the tower will therefore be at:

$$y_{bottom} = -h = -43.4 \text{ m}$$

The resulting particle trajectories are shown in figure 6.1.

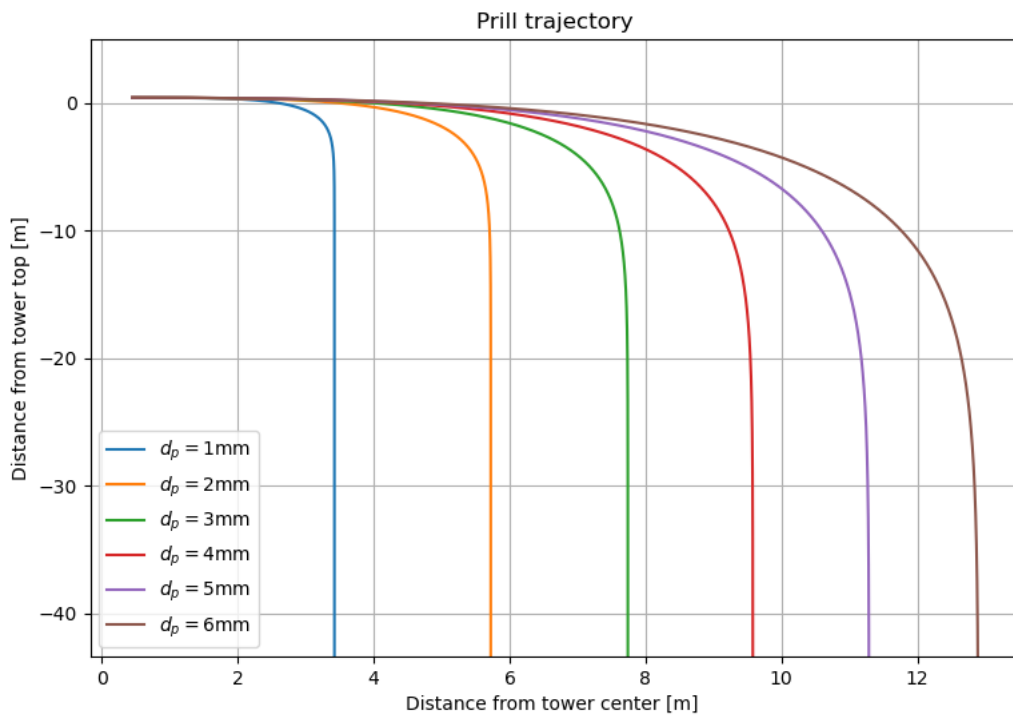


Figure 6.1: Particle trajectories for different particle diameters.

The results show a vast difference in the distance from the tower center, which is expected. Particles with a size over  $\approx 4.2$  mm will be flung more than 10 m in the x-direction. The larger the particle, the larger the distance from the tower wall becomes. Prill trajectories were also determined for varying centrifuge speeds, while keeping the particle diameter constant. The results are seen in figure 6.2.

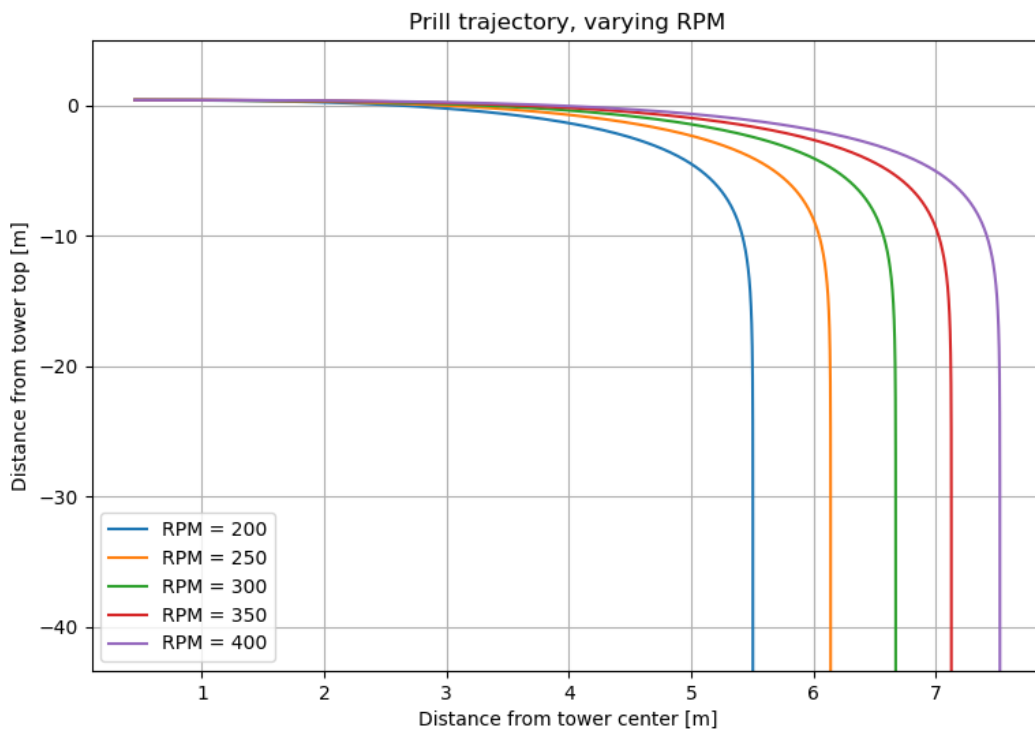


Figure 6.2: Prill trajectories as a function of centrifuge speeds (RPM). The prill diameter in this simulation was kept constant at  $d_p = 2.85$  mm.

### 6.1.2 Vertical position and velocity profiles

In this section, the particles vertical height, and velocity profiles are plotted as functions of time. This allows easy access to a given particle sizes time-of-flight, in order to determine how long the particles descent is, and when it reaches its terminal velocity. The length of the descent will directly impact the amount of heat transfer in the upcoming model. Results are shown in figure 6.3 and 6.4 on page 72.

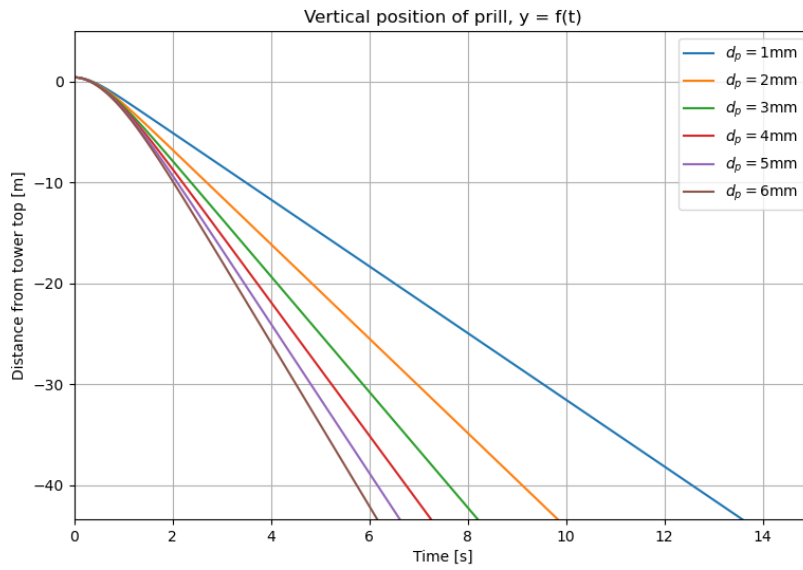


Figure 6.3: Vertical position for different particle diameters as a function of time.

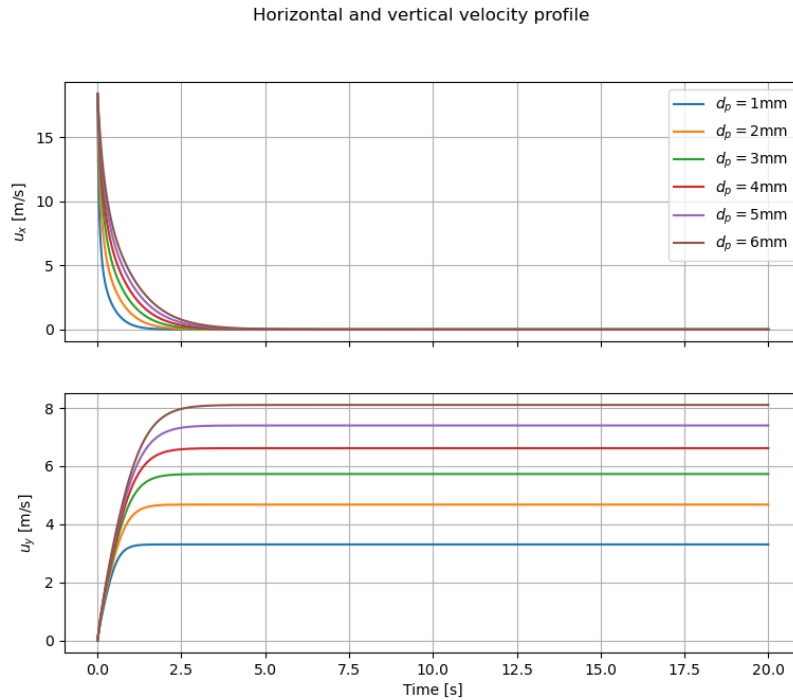


Figure 6.4: Horizontal ( $u_x$ ) and vertical ( $u_y$ ) velocity profiles for different particle diameters.



Figure 6.3 shows that a particles time-of-flight varies from just over 6 seconds for the largest particles to around 13.8 seconds for the smallest ones. By looking at the velocity profiles in figure 6.4, it is clear that larger particles take longer to reach their terminal velocities, and that the horizontal velocity continues to change even after the vertical velocity has stabilised. After a maximum of 5 seconds, all simulated particles will have reached their terminal velocities which range from 3.3-8.1 m\s.

### 6.1.3 Time-of-flight and terminal velocity

The following data was collected during model simulation:

Table 6.2: Time-of-flight and terminal velocities for 6 different particle diameters using an updated velocity model.

$d_p$ [mm]	$u_t$ [ $\frac{m}{s}$ ]	T.O.F [s]
1.00	3.31	13.58
2.00	4.68	9.83
3.00	5.73	8.22
4.00	6.62	7.26
5.00	7.40	6.62
6.00	8.10	6.16

The values in table 6.1.3 can be compared with calculations using Stokka's velocity model assumptions. The equations for calculating these values are given in chapter 3.1. An adapted model was run using Python, with identical parameters as those given in section 6.1. T.O.F, acceleration velocity and terminal velocity was calculated using Excel. The results are as follows:

Table 6.3: Time-of-flight, terminal velocities and acceleration velocities for 6 different particle diameters using a Python adapted version of Stokka's BASIC program model.

$d_p$ [mm]	$u_s$ [ $\frac{m}{s}$ ]	$u_v$ [ $\frac{m}{s}$ ]	T.O.F [s]
1	3.30	2.20	14.64
2	4.67	3.11	10.37
3	5.72	3.81	8.46
4	6.60	4.40	7.32
5	7.38	4.92	6.56
6	8.08	5.39	5.98

The values of the updated and original model were compared, and fitted to a trend-line. The results are shown in figure 6.5. The trend-lines serve to provide an easy tool to assess T.O.F's for different particle sizes. Their formula and  $R^2$  value are given below:

$$\text{Stokka's model} \longrightarrow \text{T.O.F} = 14.686d_p^{-0.5} : R^2 = 1$$

$$\text{Updated model} \longrightarrow \text{T.O.F} = 13.461d_p^{-0.442} : R^2 = 0.9993$$

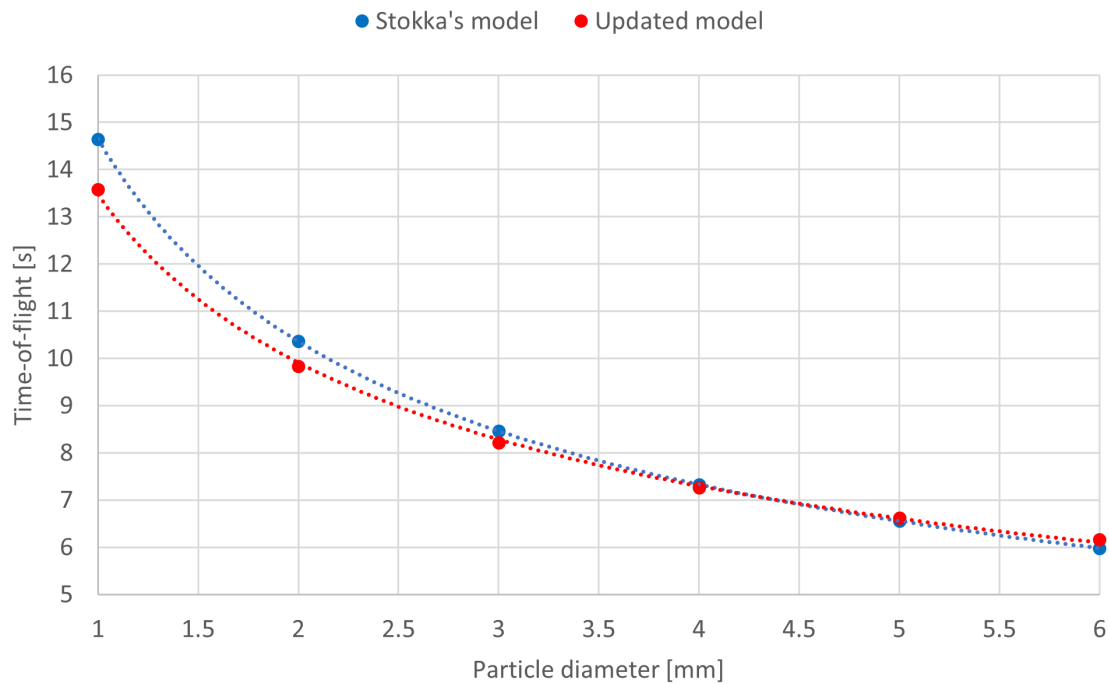


Figure 6.5: Comparison of time-of-flight for different particle sizes between Stokka's original velocity model and the updated velocity model.

## 6.2 Composite Ytterbø model

In this section, the results for the updated conduction model will be presented. Stokka's original model was adapted from BASIC to Python by the author and that program, along with Stokkas own measurements will be used as a guideline to validate the composite Ytterbø model, which is the main work of this thesis. The program will be run using measurements and data provided by Stokka himself in his 1986 paper. Any additional information that the updated model requires, was chosen based on reasonable assumptions and additional data provided by representative(s) at Yara International. The velocity section of the model was run using identical parameters to those shown in table 6.1. The

base parameters that are required for running the model are included in table 6.4, while the most important calculated parameters are shown in table 6.5.

Table 6.4: Simulation parameters for running the composite Ytterbø model. These parameters are essential parameters upon which further necessary parameters were calculated.

Parameters categories									
Particle		Tower		Operating		Crystallization		Simulation	
$d_p$	2.85 mm	r	12 m	$\dot{m}_{slurry}$	$1.06e5 \frac{\text{kg}}{\text{h}}$	$T_1$	$125^\circ\text{C}$	time	10 s
$k$	$1 \frac{\text{W}}{\text{m}\cdot\text{K}}$	h	43.4 m	$\dot{m}_{air}$	$1.16e6 \frac{\text{kg}}{\text{h}}$	$T_2$	$120^\circ\text{C}$	step	0.01
				$T_\infty$	$22^\circ\text{C}$	$T_3$	$100^\circ\text{C}$		
				$p_{air}$	101325 Pa	$T_4$	$22^\circ\text{C}$		
				$T_i$	$130^\circ\text{C}$	$Q_{cr}$	$108 \frac{\text{kJ}}{\text{kg}}$		
				RPM	390				

Based on the assumption that T.O.F for a particle with a diameter = 2.85 mm would land slightly above 8 seconds, running the model with a simulation time of 10 s was deemed appropriate. A step time of 0.01 seconds was chosen, resulting in a total of 100 iterations of the code per second of simulation. Choosing a sufficiently low step time is important to increase the accuracy of the simulation models, and is especially important considering that the step time is directly influencing the accuracy of the forward-euler method, which was used in calculating the velocities (see section 3.2.1). A time-step of 0.01 s was therefore seen to provide a good middle-ground between accuracy and computational speed.

Table 6.5: Simulation parameters calculated using composite Ytterbø model. These parameters are calculated as a part of finding the temperature of a prill as a function of time and radius.

Calculated parameters							
Particle properties		Air properties		Heat transfer		Crystallization	
$\rho_p$	$1747 \frac{\text{kg}}{\text{m}^3}$	$\rho_\infty$	$1.192 \frac{\text{kg}}{\text{m}^3}$	Bi	0.695	$c_{p1}$	$1742 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$r_{crit}$	0.83 mm	$c_p$	$1007 \frac{\text{J}}{\text{kg}\cdot\text{K}}$	$\bar{h}$	$244 \frac{\text{W}}{\text{m}^2\cdot\text{K}}$	$c_{p2}$	$14566 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
T.O.F	8.16 s	$\mu$	$182e-7 \frac{\text{N}\cdot\text{s}}{\text{m}^2}$			$c_{p3}$	$2504 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
		$k$	$25.9e-3 \frac{\text{W}}{\text{m}\cdot\text{K}}$			$c_{p4}$	$1752 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
		$\alpha$	$21.9e-6 \frac{\text{m}^2}{\text{s}}$				
		$u_\infty$	$0.712 \frac{\text{m}}{\text{s}}$				

Before producing a correct graph, the Biot number has to be calculated. Once the correct

parameters are entered, and  $Bi$  is calculated, the simulation is stopped, and the roots for the Biot number are calculated, according to equation 4.27. The roots were calculated for the following equation:

$$1 - \zeta_n \cot(\zeta_n) - 0.5446836972029505 = 0 \quad (6.1)$$

Calculations was done using `radial_eq_root_finder.py`, which can be found in Appendix E.2. All decimals were included to increase accuracy, and any positive roots that were found between 0.5 and 200 were included. A the graph of equation 6.1, highlighting the first four positive roots is shown in figure 6.6.

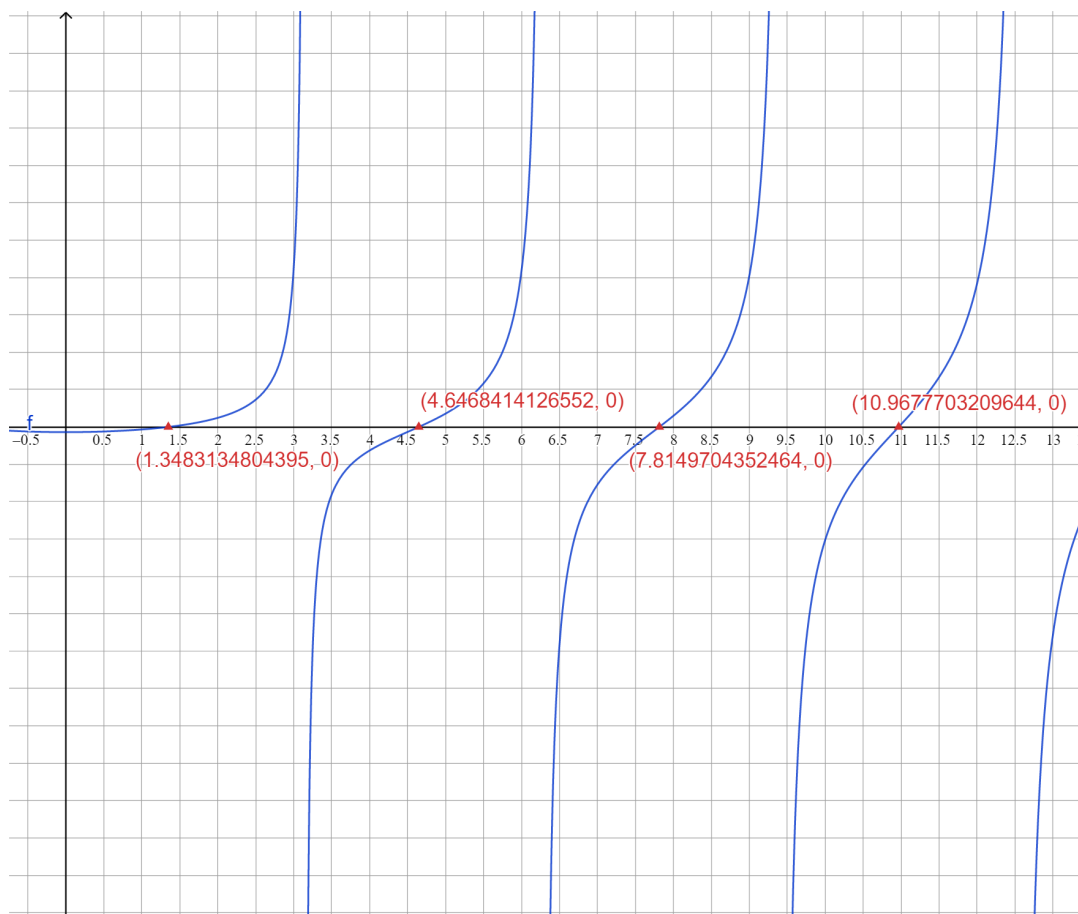


Figure 6.6: Graph showing the equation  $1 - \zeta_n \cot(\zeta_n) - 0.5446836972029505 = 0$ . The first four positive roots and their values are shown with a red triangle. This figure was made using GeoGebra's online graphing tool.

A total of 63 roots were found in the interval  $[0.5-200]$ , leading to 63 different variations of equations 4.27 and 4.26. These were all summed up according to equation 4.25 to produce one single term for  $\theta^*$  per step time, for a given radius. Equation 4.25 was combined with

equation 4.22 to calculate the temperature,  $T$ . The temperature plot for three different radii are shown below in figure 6.7.

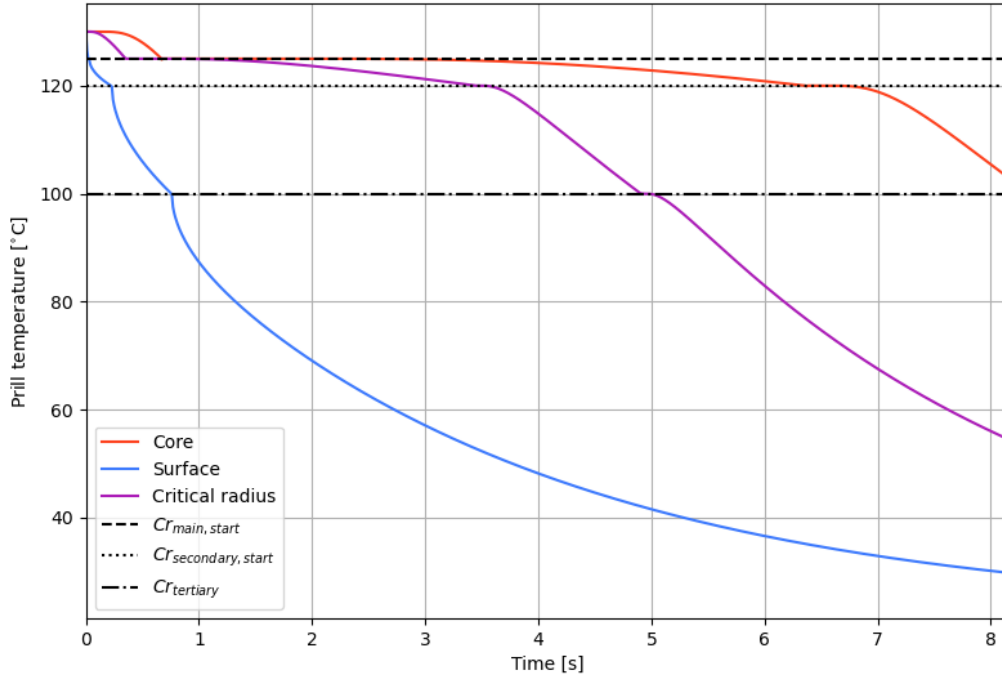


Figure 6.7: Temperature development during the descent of a single prill at three different radii.

In figure 6.7, the red line shows the core temperature, blue line shows the surface temperature, while the purple line shows the critical temperature ( $T$  at  $r = r_{crit}$ ). The stippled lines signify the start/end of a crystallization phase. The main crystallization phases for NPK 21-4-10 are given in the same setup as in table 5.1. For this particular product, crystallization begins at  $125^{\circ}\text{C}$ . The start of the main crystallization, rest crystallization (secondary) and finalizing crystallization (tertiary) are plotted in the figure. A comparison between the initial and final temperature can be seen in table 6.6.

Table 6.6: Initial, final and delta temperatures for the three radii.

	$T_{initial}$ [ $^{\circ}\text{C}$ ]	$T_{final}$ [ $^{\circ}\text{C}$ ]	$\Delta T$ [ $^{\circ}\text{C}$ ]
Core	130	102.8	27.2
Critical	130	67.5	62.5
Surface	130	35.5	94.5

It is important to highlight here that the final temperature is considered to be achieved when the particle hits the tower bottom. Any further cooling that occurs between the time it takes the particle to hit the ground, and the remaining simulation time are not considered.

### 6.2.1 Validation of updated model

The model is validated through comparison with Stokka's original model, along with assessing the model's accuracy, given Stokka's own measurements. Heat that is removed from the prills during descent are considered to be the only source of heat that heats up the air between inlet and outlet, and the air temperature increase can therefore be measured, and together with the air mass flow, total heat removed per hour can be calculated. First, equation 5.1 is used to calculate the heat removed by the air:

$$\dot{Q} = \dot{m}c_p\Delta T \implies \dot{Q}_{air} = \dot{m}_{air}c_{p_{air}}(T_{air_{out}} - T_{air_{in}})$$

On the day of 05/06-84, Stokka measured an air temperature in of 22°C, and a temperature out of 37.8°C. The heat capacity of air was calculated by the model using linear interpolation, and is shown in table 6.5. The mass flow of air was also measured, and is given in table 6.4. This leads to the following:

$$\dot{Q}_{air} = 1160000 \frac{\text{kg}}{\text{h}} 1.007 \frac{\text{kJ}}{\text{kg} \cdot \text{K}} (37.8 \text{ K} - 22 \text{ K}) = 18454518.18 \frac{\text{kJ}}{\text{h}}$$

This can be directly compared to the updated model by using the dimensionless heat loss (equations 4.29 to 4.31) to relate the surface temperature to heat loss through convection. In the updated model, this is done in the following way:

- Calculate surface temperature in intervals
- Calculate maximum possible heat transfer in each interval (equation 4.29), using each intervals initial temperature and heat capacity.
- Calculate fraction of heat lost in an interval (equation 4.30)
- Multiply fraction of heat lost ( $\frac{Q}{Q_o}$ ) with maximum possible heat loss in interval.
- After model has finished, sum up the heat lost ( $Q$ ) in each interval.
- Calculate mass flow of particles per hour.
- Multiply heat lost per particle with mass flow of particles to receive heat lost per hour.

Running model with parameters given in table 6.4 gave the following results:

$$\text{Heat released} = 18213519.47 \frac{\text{kJ}}{\text{h}} \text{ (Composite Ytterbø)}$$

Which produces an error of:

$$E = \frac{18213519.47 - 18454518.18}{18454518.18} \times 100 \% \approx -1.30 \%$$

Stokka's original model gave the following results:

$$\text{Heat released} = 16430000 \frac{\text{kJ}}{\text{h}} \text{ (Stokka)}$$

which yields an error of:

$$E = \frac{16430000 - 18454518.18}{18454518.18} \times 100 \% \approx -10.98 \%$$

### 6.3 Sensitivity analysis

After model implementation, and validation, a sensitivity analysis was run to determine which (of the selected) parameters were considered to have the greatest impact on the temperatures in the prill. The results of the model, as given in section 6.2, were considered the reference. Parameters were altered one at a time, while the remaining parameters remained constant. Often, changing the parameter caused the system to obtain a slightly different Bi-number. To produce accurate results, the Bi-number, and relevant roots were evaluated again before continuing the analysis. The temperature of the prill was analysed at three different sections:

- Core temperature ( $r = 0.01r_o$ )
- Critical temperature ( $r = r_{crit}$ )
- Surface temperature ( $r = r_o$ )

Six different parameters were selected for analysis based on the models capabilities. These parameters were:

1. Particle diameter,  $d_{p50}$
2. Slurry temperature,  $T_i$
3. Air temperature,  $T_\infty$
4. Air pressure,  $p_{air}$
5. Thermal conductivity of the particle,  $k$
6. Particle density,  $\rho_p$

These cases are further divided into two subsections:

- Independent parameters
- Dependent parameters

Independent parameters are characterized by their ability to remain detached from the rest of the system. For example, a change in air temperature does not inherently affect the particle composition in the mixer. However, the particle density and thermal conductivity **are** connected to composition. They are therefore considered to be dependent attributes. A summary of the cases (1-6), their ranges, and reference values are shown in table 6.7:

Table 6.7: Case numbers, parameters, units, ranges and reference values used in the sensitivity analysis of NPK 21-4-10.

Case	Parameter	Unit	Range	Reference
1	Particle diameter	mm	[1.5-4.0]	2.85
2	Slurry temperature	°C	[126-140]	130
3	Air temperature	°C	[0-25]	22
4	Air pressure	Pa	[95000-105000]	101325
5	Thermal conductivity	$\frac{W}{m^2 \cdot K}$	[0.6-1.2]	1
6	Particle density	$\frac{kg}{m^3}$	[1600-2000]	1747

The model was run between 4-6 times in the given ranges (one run per value) to produce reasonable comparisons. For each run, all temperatures were logged, and compared to the reference. Two plots were made for each radius, one that showed the temperature development, and one that showed the temperature difference development, compared to the reference value. The temperature difference in this case is given as:

$$\Delta T = T_{\text{run X, case Y, radius Z}} - T_{\text{ref}}$$

In general, for all variables, the  $\Delta T$  development can be summarized as:

- Core temperature:  $\Delta T$  starts out low, spikes as  $t \rightarrow 6.5$  s,
- Critical temperature:  $\Delta T$  starts out low, spikes as  $t \rightarrow 3.5$  s, followed by a small decrease before increasing again between  $t = [5.5-6.5]$  s
- Surface temperature:  $\Delta T$  starts increasing rapidly after simulation start, two local spikes within  $t = [0.5-1.5]$  s. After these spikes,  $\Delta T$  stabilizes.

An example of the temperature development and temperature difference development graph is shown in figure 6.8 below. A summary of all the sensitivity analysis data is



provided in Appendix B.1, while the data from the individual model runs can be found in Appendix B.2

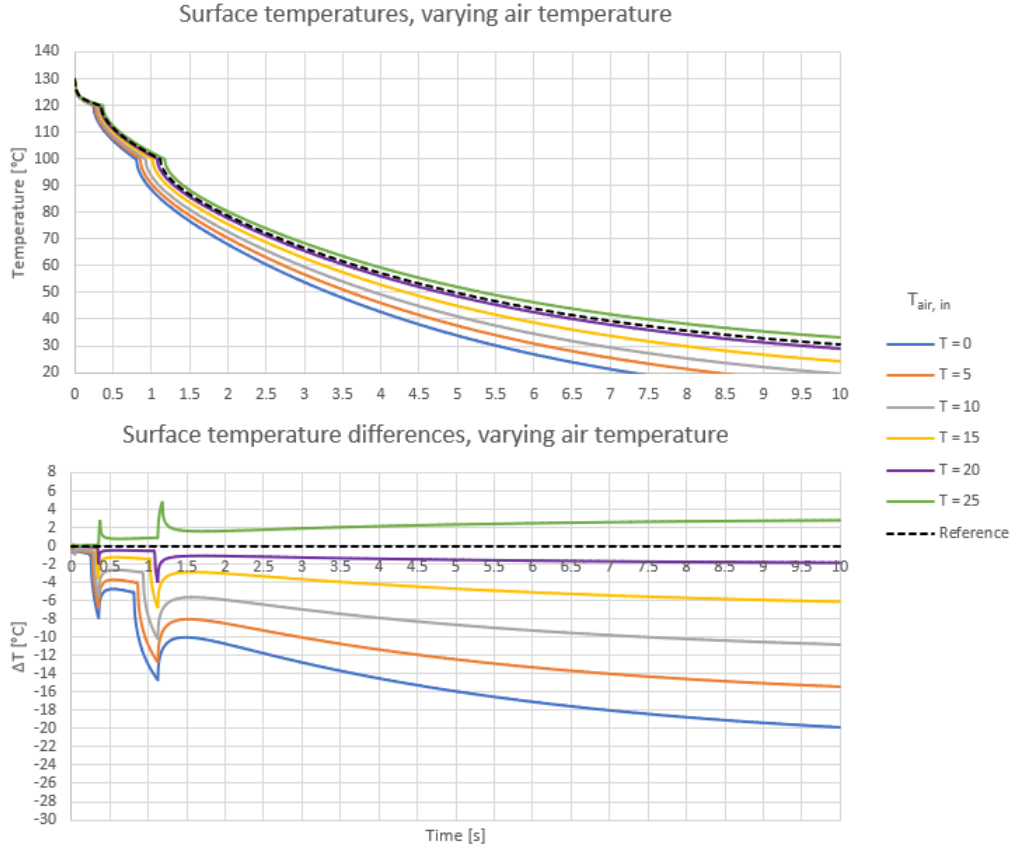


Figure 6.8: Figure showing the temperature- and temperature difference development on the surface of a prill as a result of varying the ambient air temperature in the composite Ytterbø model. Black dotted lines represent the reference value, which, in the case of ambient air temperature is equal to a simulation with  $T_{air,in} = 22^\circ\text{C}$

The temperature differences were averaged for each radius over the total simulation time according to the following equation:

$$\Delta\bar{T} = \frac{1}{N_t} \sum_{i=1}^{N_t} \Delta T_i \quad (6.2)$$

Where  $N_t$  is the number of iterations in the simulation/run. The maximum change in temperature differences across the ranges were calculated by subtracting the largest from the smallest value, and by dividing the value on the parameter range, it was possible to obtain *average expected temperature difference at given radii per unit increase/decrease*. These are the values have been chosen to represent the models sensitivity to a change in different parameters. A summary of the results are given in table 6.8.

Table 6.8: Average expected temperature difference at given radii per unit increase/decrease for 6 different parameters.

Case	Parameter	Unit	$T_{core}$	$T_{crit}$	$T_{surface}$
1	Particle diameter	$\frac{\Delta T}{\text{mm}}$	24.363	25.395	11.606
2	Slurry temperature	$\frac{\Delta T}{\text{K}}$	0.289	0.377	0.157
3	Air temperature	$\frac{\Delta T}{\text{K}}$	0.238	0.341	0.680
4	Air pressure	$\frac{\Delta T}{\text{kPa}}$	0.477	0.130	0.098
5	Thermal conductivity	$\frac{\Delta T}{\frac{\text{W}}{\text{m}^2 \cdot \text{K}}}$	17.764	27.729	7.159
6	Particle density	$\frac{\Delta T}{\frac{\text{kg}}{\text{m}^3}}$	0.013	0.026	0.012

This allows us to create a preliminary sensitivity ranking based on the different radii. The temperature sensitivity is ranked from 1 (most sensitive) to 6 (least sensitive) for the different radii. The results of this ranking is shown in table 6.9.

Table 6.9: Sensitivity ranking of different model parameters based on radius

Ranking	Core temperature	Critical temperature	Surface temperature
1	Particle diameter	Thermal conductivity	Particle diameter
2	Thermal conductivity	Particle diameter	Thermal conductivity
3	Air pressure	Slurry temperature	Air temperature
4	Slurry temperature	Air temperature	Slurry temperature
5	Air temperature	Air pressure	Air pressure
6	Particle density	Particle density	Particle density

The average differences of each run within one case was also plotted against each other, in order to see whether an increase in the parameter would lead to a linear/polynomial increase in the expected temperature difference. This was done to provide a simplistic tool for evaluating expected temperatures at different parameter values, given a basis in the model. All graphs were plotted using Excel, and trend-lines were fitted. The trend-line equations, graphs and  $R^2$  values are all given in Appendix B.

### 6.3.1 Example: Calculating new critical temperature

Consider the following example: NPK 21-4-10 is being produced on a day where the average air inlet temperature is 10°C. Assume that the quality of the product is dependent

on the critical radius temperature. What particle diameter could we now produce to equal the reference models temperatures?

This problem can be tackled using the experimental data in two ways. One can begin by looking at table 6.8, and see that the average expected temperature difference, per degree (for air temperature) is 0.341 for the critical radius. We use the reference model values to calculate total expected drop:

$$\Delta T = 0.341 \Delta T_{air} = 0.341 \frac{\Delta T}{K} (T_{air_{current}} - T_{air_{reference}}) = 0.341 \frac{\Delta T}{K} (10 - 22) K$$

$$\Delta T = -4.092 \implies T_{crit_{current}} - T_{crit_{reference}} = -4.092 \implies T_{crit_{current}} = -4.092 + T_{crit_{reference}}$$

This means (using the critical temperature as given in table 6.6) that the new expected critical radius temperature is:

$$T_{crit_{current}} = (-4.092 + 67.5)^\circ C = 63.408^\circ C \quad (6.3)$$

### 1. Expected average difference (EAD) method

This new temperature can be useful on its own, or, the temperature difference could be used to calculate how much of an increase in particle diameter the process could handle, given the new conditions ( $T_{air}$ ). From table 6.8, it is shown that the temperature at the critical radius increases by an average of:

$$\frac{\Delta T}{mm} = 24.363 \frac{K}{mm}$$

The following equation can then be used:

$$d_{p_{new}} = d_{p_{ref}} - \frac{\Delta T}{\frac{\Delta T}{mm}}$$

Inserting values yields:

$$d_{p_{new}} = 2.85 \text{ mm} - \frac{(-4.092 \text{ K})}{24.363 \frac{K}{mm}} = 2.85 \text{ mm} + 0.168 \text{ mm} \approx 3.02 \text{ mm}$$

In other words, running the Composite Ytterbø model with an ambient air temperature of 10°C, and an average particle diameter of 3.02 mm should yield similar critical temperatures as running the model with an ambient air temperature of 22°C and an average particle diameter of 2.85 mm.

## 2. Trend-line method

Using the air temperature summary in Appendix B.1, for the critical temperature as a function of air temperature, the following trend-line equation can be used:

$$\Delta T = 0.3552T_{air} - 7.6231, R^2 = 0.986$$

Values are inserted the estimated critical temperature difference can be calculated, using the new air temperature:

$$\Delta T = 0.3552 \times 10 - 7.6231 = -4.080^\circ\text{C}$$

This means that going from 22 to 10°C, we can expect a critical temperature drop of -4.08 °C. In order to calculate the increased diameter, the sign of the temperature difference has to be changed (we are going from a lower critical temperature, up to reference):

$$\Delta T = 4.080^\circ\text{C}$$

Using the particle diameter summary in Appendix B.1, the correlation between particle diameter and temperature difference is given by the following equation:

$$\Delta T = -5.9065x^2 + 58.316x - 117.94, R^2 = 0.9989$$

Plotting the intersection between the equation above, and  $y = 4.080$  gives us an x-value ( $d_p$ ) of:

$$d_p = 3.01 \text{ mm}$$

## Validation of calculation example

Running the model with an ambient air temperature of 10°C, and particle diameters as calculated in section 6.3.1 using EAD and trend-line method, 3.02 and 3.01mm respectively, gives us the following result:

Table 6.10: Calculated critical temperatures after finding new particle diameter through the EAD and trend-line method.

Method	$d_p$ [mm]	T.O.F [s]	$T_{crit}$ [°C]
EAD	3.02	8.03	67.071
Trend	3.01	8.05	66.192

Comparing the values of the critical temperatures in table 6.10 with the original critical temperature of 67.469°C, the error for each method can be calculated:

$$E = \frac{67.071 - 67.5}{67.5} \times 100 \% \approx -0.59 \%, \text{ EAD}$$

$$E = \frac{66.192 - 67.5}{67.5} \times 100 \% \approx -1.89 \%, \text{ Trend-line method}$$

## 6.4 Model adaptation for other products

An attempt was made at adapting the composite Ytterbø model for several different products. The core model remained the same, but some new elements were taken into consideration. The composition formula was changed to reflect each individual product, however, only the main constituents (other than AN and MAP) were taken into account. By main constituents, it is here implied that their weight fraction is  $\geq 1\%$ . All other parameters were kept as shown in table 6.4. This lead to the following considerations:

- Fraction of KCl (MOP) in NP-melt
- Fraction of  $K_2SO_4$  (SOP) in NP-melt
- Fraction of Kieserite in NP-melt
- Fraction of Dolomite in NP-melt

The purpose of this addition was to provide a better measure of product particle density, and a more accurate T.O.F calculation. In addition, new crystallization degrees were extrapolated based on the solid fractions of the different products at different temperatures (see Appendix D.1-D.4). Instead of defining main and rest crystallization areas based on the percentage of crystallization, and developing the heat capacities around these areas, the focus was on specific temperature intervals, making the phase diagram easier to use. The intervals were constructed as follows:

- Interval 1: Mixer temperature  $\rightarrow$  Initial crystallization temperature
- Interval 2: Initial crystallization temperature  $\rightarrow 120^\circ\text{C}$
- Interval 3:  $120^\circ\text{C} \rightarrow 100^\circ\text{C}$
- Interval 4:  $100^\circ\text{C} \rightarrow 50^\circ\text{C}$

Seeing as how it is beneficial for the energy consumption to keep crystallization from occurring in the mixer, while keeping the temperature as low as possible,  $T_{mixer}$  was set to:

$$T_{mixer} = T_{cr} + 1^\circ\text{C}$$

The adapted model was used for analyzing the heat release, prill temperature development and total crystallization of NPK 25-2-6 (MOP), NPK 21-4-10 S+B, NPK 21-06-13S and NPK 25-07-07 (MOP). The heat release and prill temperature development was calculated by the composite Ytterbø model, while the total crystallization was estimated by using the phase diagrams for the different products, given in Appendix D.1-D.4. When assessing the total crystallization, it is assumed that 1 % of the volume remains liquid, regardless of core temperature. The total crystallization degree can then be estimated as:

$$\epsilon_{tot} = \bar{X}_s|_{T=T_{core}(T.O.F)} - 0.01 \quad (6.4)$$

The results are presented in separate sections below.

### 6.4.1 NPK 25-2-6 (MOP)

The temperature development of a 2.85 mm diameter prill of NPK 25-2-6 (MOP) is shown in figure 6.9 below:

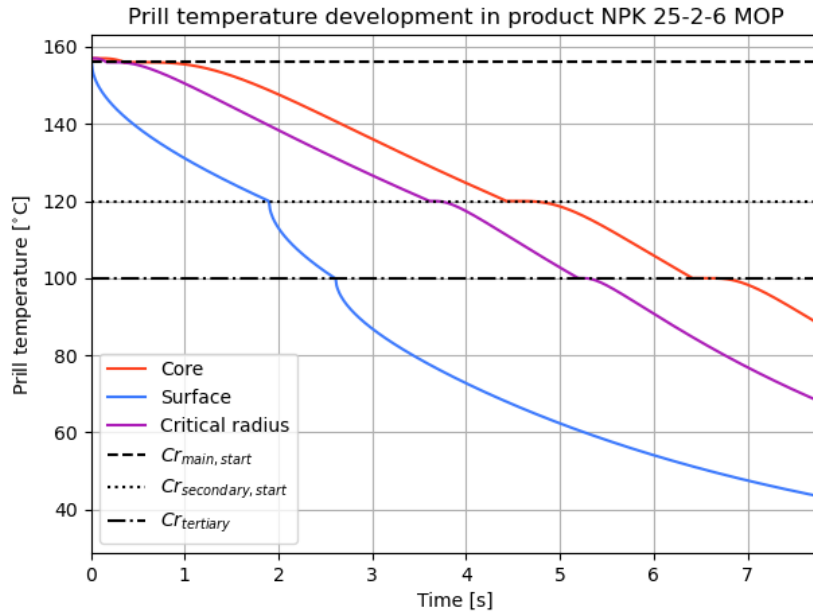


Figure 6.9: Temperature development of a 2.85 mm prill of NPK 25-2-6. Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line).

The total crystallization degree was calculated using equation 6.4 and Appendix D.1:

$$\varepsilon_{tot, \text{NPK 25-2-6}} = 0.951 - 0.01 = 0.941 = 94.1 \%$$

Key values from the simulation can be seen in table 6.11.

Table 6.11: Calculated parameters from running the composite Ytterbø model with NPK 25-2-6 (MOP) as product.

Calculated parameters									
Particle properties		Heat transfer			Interval data				
$\rho_p$	$1910 \frac{\text{kg}}{\text{m}^3}$	Bi	0.545	$T_i$	$157^\circ\text{C}$	$\varepsilon_1$	0	$C_{p1}$	$1636 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
T.O.F	7.78 s	$\bar{h}$	$191.23 \frac{\text{W}}{\text{m}^2\cdot\text{K}}$	$T_1$	$156^\circ\text{C}$	$\varepsilon_2$	0.87	$C_{p2}$	$3499 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{core}$	$87.86^\circ\text{C}$	$Q_{released}$	$23.27 \times 10^6 \frac{\text{kJ}}{\text{h}}$	$T_2$	$120^\circ\text{C}$	$\varepsilon_3$	0.92	$C_{p3}$	$2097 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{crit}$	$67.89^\circ\text{C}$			$T_3$	$100^\circ\text{C}$	$\varepsilon_4$	1	$C_{p4}$	$1652 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{surface}$	$43.34^\circ\text{C}$			$T_4$	$50^\circ\text{C}$				

### 6.4.2 NPK 21-4-10 S+B

The temperature development of a prill of a 2.85 mm diameter NPK 21-4-10 S+B is shown in figure 6.10 below:

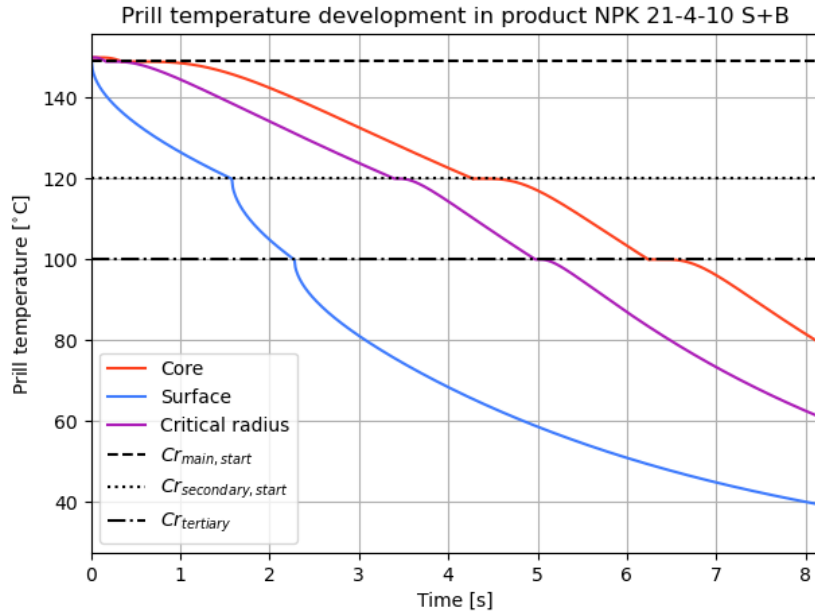


Figure 6.10: Temperature development of a 2.85 mm diameter prill of NPK 21-4-10 S+B. Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line).

The total crystallization degree was calculated using equation 6.4 and Appendix D.2:

$$\epsilon_{tot, NPK\ 25-2-6} = 0.960 - 0.01 = 0.950 = 95.0 \%$$

Key values from the simulation can be seen in table 6.12.

Table 6.12: Calculated parameters from running the composite Ytterbø model with NPK 21-4-10 S+B as product.

Calculated parameters									
Particle properties		Heat transfer			Interval data				
$\rho_p$	$1710 \frac{\text{kg}}{\text{m}^3}$	Bi	0.543	$T_i$	149°C	$\epsilon_1$	0	$C_{p1}$	$1768 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
T.O.F	8.17 s	$\bar{h}$	$190.38 \frac{\text{W}}{\text{m}^2\cdot\text{K}}$	$T_1$	148°C	$\epsilon_2$	0.86	$C_{p2}$	$4288 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{core}$	79.33°C	$Q_{released}$	$23.90 \times 10^6 \frac{\text{kJ}}{\text{h}}$	$T_2$	120°C	$\epsilon_3$	0.94	$C_{p3}$	$2327 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{crit}$	60.90°C			$T_3$	100°C	$\epsilon_4$	1	$C_{p4}$	$1783 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{surface}$	39.26°C			$T_4$	50°C				

### 6.4.3 NPK 21-06-13S

The temperature development of a prill of a 2.85 mm diameter NPK 21-06-13S is shown in figure 6.11 below:

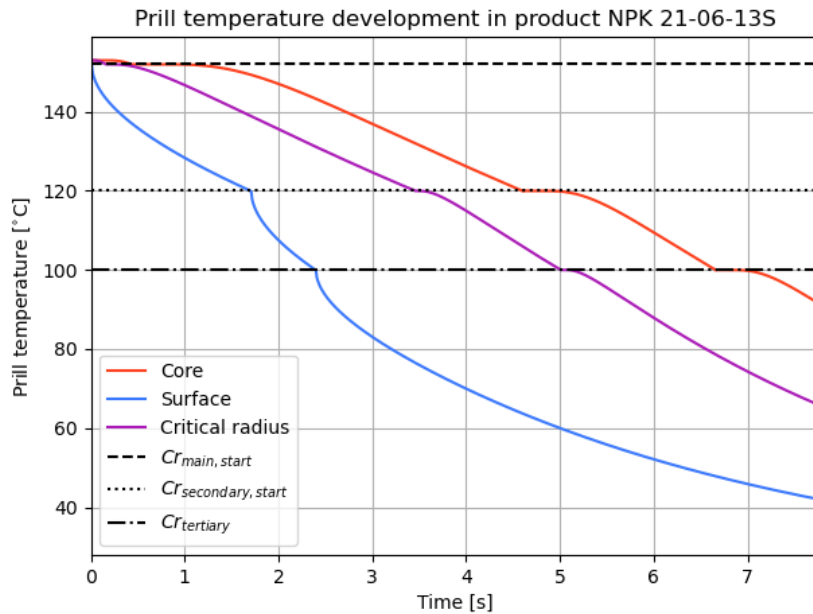


Figure 6.11: Temperature development of a 2.85 mm prill of NPK 21-06-13S. Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line).

The total crystallization degree was calculated using equation 6.4 and Appendix D.3:

$$\epsilon_{tot, \text{NPK 21-06-13}} = 0.953 - 0.01 = 0.943 = 94.3 \%$$

Key values from the simulation can be seen in table 6.13.

Table 6.13: Calculated parameters from running the composite Ytterbø model with NPK 21-06-13S as product.

Calculated parameters									
Particle properties		Heat transfer			Interval data				
$\rho_p$	$1910 \frac{\text{kg}}{\text{m}^3}$	Bi	0.553	$T_i$	$153^\circ\text{C}$	$\epsilon_1$	0	$C_{p1}$	$1636 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
T.O.F	7.78 s	$\bar{h}$	$194.23 \frac{\text{W}}{\text{m}^2\cdot\text{K}}$	$T_1$	$152^\circ\text{C}$	$\epsilon_2$	0.87	$C_{p2}$	$3707 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{core}$	$91.20^\circ\text{C}$	$Q_{released}$	$22.49 \times 10^6 \frac{\text{kJ}}{\text{h}}$	$T_2$	$120^\circ\text{C}$	$\epsilon_3$	0.93	$C_{p3}$	$2143 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{crit}$	$65.63^\circ\text{C}$			$T_3$	$100^\circ\text{C}$	$\epsilon_4$	1	$C_{p4}$	$1649 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{surface}$	$41.91^\circ\text{C}$			$T_4$	$50^\circ\text{C}$				



#### 6.4.4 NPK 25-07-07 (MOP)

The temperature development of a 2.85 mm diameter prill of NPK 25-07-07 (MOP) is shown in figure 6.12 below:

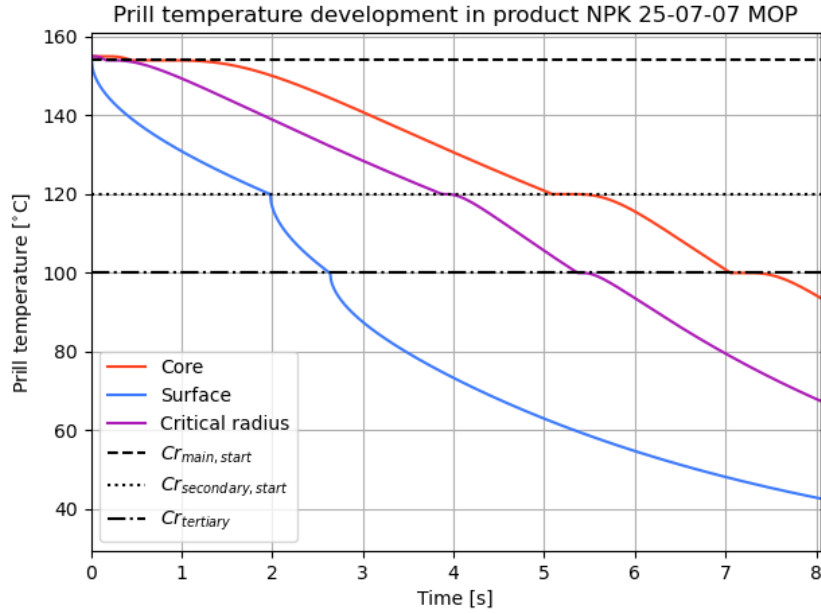


Figure 6.12: Temperature development of a 2.85 mm prill of NPK 25-07-07 (MOP). Temperature developments are shown for the core (red line), critical radius (purple line) and surface (blue line).

The total crystallization degree was calculated using equation 6.4 and Appendix D.4:

$$\epsilon_{tot, NPK\ 25-07-07} = 0.944 - 0.01 = 0.934 = 93.4\ \%$$

Key values from the simulation can be seen in table 6.14.

Table 6.14: Calculated parameters from running the composite Ytterbø model with NPK 25-07-07 (MOP) as product.

Calculated parameters									
Particle properties		Heat transfer			Interval data				
$\rho_p$	$1767 \frac{\text{kg}}{\text{m}^3}$	Bi	0.546	$T_i$	$157^\circ\text{C}$	$\epsilon_1$	0	$C_{p1}$	$1843 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
T.O.F	8.05 s	$\bar{h}$	$191.52 \frac{\text{W}}{\text{m}^2\cdot\text{K}}$	$T_1$	$156^\circ\text{C}$	$\epsilon_2$	0.925	$C_{p2}$	$4330 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{core}$	$93.59^\circ\text{C}$	$Q_{released}$	$25.84 \times 10^6 \frac{\text{kJ}}{\text{h}}$	$T_2$	$120^\circ\text{C}$	$\epsilon_3$	0.955	$C_{p3}$	$2170 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{crit}$	$67.44^\circ\text{C}$			$T_3$	$100^\circ\text{C}$	$\epsilon_4$	1	$C_{p4}$	$1849 \frac{\text{J}}{\text{kg}\cdot\text{K}}$
$T_{surface}$	$42.66^\circ\text{C}$			$T_4$	$50^\circ\text{C}$				



## 7 Discussion

In this section, the results presented in chapter 6 will be discussed. The topics will be addressed in the same order as they are presented in chapter 6. This section will encompass the following:

1. The updated velocity model
2. The updated heat transfer model (composite Ytterbø)
3. Sensitivity analysis
4. NPK product analysis

### 7.1 The updated velocity model

The calculated prill trajectories seems to accurately reflect particle behaviour in the prilling tower. Given that the radius of the tower is approximately equal to 12m, then particles of size 5.5-6 mm are at risk of wall collision. The rotation speed of the centrifuge was tested, and for a particle of diameter 2.85 mm, the horizontal distance vary from 5.7 to 7.8 m between 200 and 400 RPM. The particle sizes also seem to greatly affect the distance, which is as expected. The same can be said for the vertical position and velocity profiles of the prills. Time-of-flight variations are directly proportional to particle sizes, and the variations are quite large. It is observed that the particle velocity more than doubles in the particle size range [1-6 mm]. Comparing these data to Stokkas older model, the results were quite surprising. Despite several simplifying assumptions, the calculated T.O.F values compared to their respective particle diameters showed quite similar results to the updated model. An important thing of note here is that Stokka explicitly writes that *"it is assumed that the acceleration period is present in the first 10m of the tower, given particle sizes of [2.5-3.5 mm]"* [6]. When running Stokka's older model, with particle sizes from [1-6 mm], this assumption was made for every size. By looking at figure 6.1, this assumption is quite good for particles between 2 and 4 mm (as expected), but smaller particles attain their terminal velocity before the 10 m mark, and larger particles attain their terminal velocity after. By looking at the figure, the terminal velocity for particles of size  $d_p = 1$  mm is reached after ca. 4 m, while for particles of size  $d_p = 6$  mm, it is not achieved until it has fallen ca. 30 m. This is an obvious source of error, and would spread out the differences in the graphs seen in figure 6.5. However,

for the applicable size of 2.85 mm, both models are seen to give accurate results when compared to other calculations [7], [8], [45]. One flaw that encompasses both models, is the lack of air temperature variation. The air temperature is considered to be ambient throughout the tower, which is lower than what the actual temperature is. By the time the air has made it through to the top, it estimated that the temperature has increased around 14-17°C [6], which would cause a substantial air density difference throughout the tower, which would again affect the particle velocity. Based on this, it can be assumed that the particle velocities and time-of-flight values calculated by the model are higher than what they would realistically be.

## 7.2 Updated heat transfer model

Unlike the velocity model, which simply specified the force equations in more detail, the updated heat transfer model creates an entire new way of approaching the problem. It was initially assumed that the temperature of the particle could be seen as even, but that claim was proven to be false, given that the Biot-number did not satisfy the requirements of the LCM-method in any of the cases run in this thesis. This led to a combination between a newly developed heat conduction frame, and an older heat capacity method. While initially tough to combine and implement, the results show great promise. The calculated heat capacities are similar to the original model developed by Stokka, with exception of  $c_{p3}$  and  $c_{p4}$ . In Stokka's original model, these were not calculated using equation 5.3, but taken from I. M. Hugstmyr (1987) [46]. Due to the unavailability of heat capacity values for different products, all heat capacities were calculated using equation 5.3 when running the composite Ytterbø model.

Under the assumption that all heat taken up by the air is caused by heat release from prills during their descent, the margin of error is dramatically reduced. The updated model show improvements by reducing the error from 10.98 to 1.30 %. While the results are good in this particular case, it is tough to estimate the overall accuracy of the model. This is largely due to the simplification that is made in the heat capacity model. While providing an accurate end result (in the case of running the model using Stokka's parameters), it hampers the development of a more accurate model. Keeping the heat capacity constant within intervals makes other solutions of developing phase-change models quite difficult. Similar models of prilling towers have been made, including moisture content transfer and thermal phase changes through the Stefan model [47]. Farrokhanpanah et al. (2017) developed a SPH model for evaluating heat conduction with solidification and melting, which show promise [48]. Voller et al. (1987) also developed a numerical modelling scheme for phase transitions [49]. Although this particular knowledge is currently out of reach for the author, it is considerations to be made for future model upgrades. One improvement that was attempted was to include a more realistic model for the convection coefficient. In this model, the convection coefficient is taken as an average, regardless

of how the fluids velocity changes. One major obstacle in implementing this, is that the Biot number is dependent on the convection heat transfer. If the updated model presented one convection coefficient for every time step, then the Biot number would also change every time-step. This would again lead to new roots being produced, according to equation 4.27, which would cause the computational solve time to skyrocket. It is possible that this could be possible to circumvent by improved coding-capabilities, but that was beyond the authors reach. When assessing the complex nature of the NPK fertilizers and the prilling tower in general, it is understandable why the heat capacity and velocity simplifications were made. Most prilling tower models use urea, or pure ammonium nitrate. Implicitly, this means that the heat capacities of the prills can be more easily identified, without further modeling, which allows for a more complex framework (such as CFD modelling). The heat capacities of NPK fertilizers are more difficult to analyze, given that different products solidify at different temperatures, and in different orders. It is therefore highly recommended that any further development to this model is accompanied by relevant experimental and process testing. As it stands, however, this model successfully shows the implementation of a transient conduction model. The transient model also opens up for accurately evaluating the air temperature. Alamdari et al. (2000) [7] showed that using a transient conduction model to assess air temperature development in the tower provided very good correlations with plant data, compared to the lumped-capacitance method. Summarized, the transient conduction model can be considered a step in the right direction towards a product-dependent, accurate model of the NPK-3 prilling tower.

### 7.3 Sensitivity analysis

When assessing the sensitivity of the model, all parameters were kept constant, except the parameter that was being tested. The temperature development and temperature difference all show abrupt changes when a simulated prill goes from one heat capacity interval to another, which is expected when looking at the temperature graph (see figure 6.8). The general temperature differences were quite consistent during all parameters, however, there were some outliers. These outliers are almost exclusively found when looking at the core temperatures. For example, running the reference model with an ambient air temperature of 0°C, yields a higher core temperature after 10 s than running the model with  $T_{air,in} = 5^\circ\text{C}$  and  $T_{air,in} = 10^\circ\text{C}$ , which seems counter-intuitive. A similar issues was found when testing for particle density and ambient air pressure. In the model, altering the air temperature affects the convection heat transfer through the convection heat coefficient, and subsequently the Biot number, which affects the conduction equations. The particle density affects the particles velocity, which in turn affects the relative Nusselt number of the fluid flow, and consequently the heat convection and conduction through the Biot number. A reduction in air pressure (given a constant mass flow) will lead to reduced density, which again leads to increased velocity through a reduction in drag and

buoyancy forces. It is unsure why these parameters deviate from normal behaviour when assessing certain values. The deviations in core temperatures for air temperature, air pressure and particle density can be seen in Appendix B. Here trend-lines deviate from linear or polynomial behaviour, and show  $R^2$  values of 0.8377, 0.7745 and 0.8143 respectively. Since the problem is most often connected to the core calculations, it is reasonable to assume that a mistake has been made on the authors part when programming the model.

In general, the critical and surface temperatures behave as expected when running the model with different parameter values. The temperatures increase when particle size, slurry temperature, air temperature and particle density increases, and decreases when the ambient air pressure increases. As for the thermal conductivity, increased values of  $k$  results in a decreased critical temperature, and an increased surface temperature. This makes sense, as the gradient between the layers become less steep as the conductivity increases. Setting a significantly high conductivity value would be equal to assuming equal temperature throughout the prill, which would be equivalent to using the lumped-capacitance method.

As for the sensitivity ranking between the different parameters, the model is most sensitive to particle diameter and thermal conductivity. This could point to that in order to further develop this model, implementing a particle-size control through the centrifuge (by using Rayleigh's jet theory), could be helpful. Also, the thermal conductivity used in the base model is assumed to be  $1 \text{ W}\backslash(\text{m}\cdot\text{K})$ . As seen from table 6.8, an increase/decrease in the thermal conductivity of  $\pm 0.1 \text{ W}\backslash(\text{m}\cdot\text{K})$  could yield a temperature difference of up to  $\pm 2.7^\circ\text{C}$  when the particle hits the tower bottom. The thermal conductivity is dependent on the salt composition of the prill, and it would therefore be useful to do experimental testing or mathematical modelling to determine accurate thermal conductivities, based on different products. From table 6.8, one can also observe that while particle diameter and thermal conductivity rank highest for all radii, air temperature influence seem to increase as the radius increases, which is expected. On the other hand, the air pressure has the opposite effect, where its influence is strongest at the core. The only connection between the air pressure and core temperature is through the convection coefficient, Biot-number, and calculated roots. This seems to indicated that the core might be "over-sensitive" to changes that affect the convection conditions. One glaring "problem" with this ranking, is that it is based on specific units. Thermal conductivity can contribute to a temperature change that is 10x the magnitude of the slurry temperature *per unit*. If this is taken at face value, the numbers might seem misleading. It is not uncommon to alter the slurry temperature by more much than  $\pm 1^\circ\text{C}$ , however, ending up with a compound where the thermal conductivity is  $\pm 1 \text{ W}\backslash(\text{m}\cdot\text{K})$  is highly unlikely. Realistic thermal values could vary by up to  $\pm 0.4 \text{ W}\backslash(\text{m}\cdot\text{K})$ , depending on product composition [50]. This is why table 6.9 should be used with some caution. One major benefit of this system however, is the ability to "predict" what kind of changes we could see in prill temperatures, based on the process conditions. The use of *Expected Average Difference*-, or trend-line method can be

used to say something about how the prill temperature responds to process changes, based on the reference parameters used in this thesis. This could be of great value, and can be used (carefully) even to assess other product types. However, the model itself is built on top of a reference product, and any applications beyond this product will reduce the estimation accuracy. However, the basis of these methods can prove to be accurate tools for future process assessment when different products have successfully been characterised for use with the composite Ytterbø model.

## 7.4 Model adaptation and NPK product analysis

Running the composite Ytterbø model for the four focus grades produced significantly different graphs than the reference model (NPK 21-4-10). The main reason for this is the calculated  $C_p$  values, which were significantly lower in the second interval for the focus grades. Running the reference model gave a  $C_p$  value for interval two of ca. 14500 J/(kg·K), while the focus grades produced values ranging from 3499 J/(kg·K) to 4330 J/(kg·K). As seen in equation 5.3, the temperature interval is in the denominator. If the temperature interval where the main crystallization happens is large, then the heat capacity gets reduced. The calculations done on the heat capacity is identical to the calculation Stokka did for NP 23-23 in his report (1989) [6]. The only difference here is that the heat of crystallization is considered to be 108 kJ/kg, which is the determined value for MAP-AN-KCl melt phase. This value of 108 kJ/kg, is used for all products that were analyzed, even though KCl is not present in NPK 21-6-13S. This was simply due to lack of available measurements of heat of crystallization values for the different product compositions. These product models were not subject to validation, as opposed to the core section of the composite Ytterbø model. Validation would be required to use the model further, but can be accomplished. By looking at measured NSP values, or measure prill temperatures for a focus grade product, and inputting process parameters from that time into the model, one could get a reasonable way of validating the model. Unfortunately, calibrating the model was deemed to be too time-consuming for this thesis.

When assessing the results themselves, there are large differences in heat released. As mentioned previously, the heat released is directly influenced by the time-of-flight, and it is clear that the lower density focus grades (NPK 21-4-10 S+B and NPK 25-07-07) also release the most heat. As mentioned in section 4.2.3, the temperature of the particle at the critical radius at the time of impact could be a good implicit estimator as to whether a particle is fully crystallized or not. From the model results, it is seen that NPK 21-4-10 S+B has the lowest critical radius temperature of 60.9°C. The highest critical radius temperature is achieved with NPK 25-2-6, which reached 67.89°C. At this point it seems as though both these products should be fully crystallized by the time they reach the tower bottom, however, the particle size in question is 2.85 mm. In reality, the particle size distribution will be scattered, and include both smaller and larger particles.

The implicit result that can be drawn from these simulation, by using the critical radius temperature, is ranking the perceived difficulty of prilling. In this case, the product with the highest critical temperature would be the most difficult. Another interesting fact is that the model shows no absolute correlation between the temperature at the different radii. NPK 25-2-6 sports the highest critical radius temperature, but NPK 21-06-13S shows the highest core temperature. This would be equivalent to changing the thermal conductivity of the material directly, but instead, the model handles this through the salt composition, which is very promising. What the model does not account for, are particle size distributions. This would, as mentioned above, be very important in assessing the overall performance of a product, both by looking at the prilled particles themselves, but also the salt particle distribution. Previous work has determined that the particle size of the potassium chloride has to be critically selected [4], [15]. It was reported that too fine particles would be difficult to mix properly, and too large particles could cause issues with clogging of the holes in the prilling bucket. In this model, the only thing considered is the fraction of potassium chloride in the melt, and it's conversion factor which is assumed constant at 40 %. This leaves a large room for improvement, both when considering particle- and salt size distributions.

Due to the model producing such low heat capacities in comparison to the reference model, the temperature difference between the core and critical radius are much lower when looking at the focus grades. This meant that it was feasible to use the core temperature directly to verify to what extent crystallization had occurred. While running the model with reference parameters, all particles showed > 93 % crystallization, with NPK 21-4-10 having the highest degree of crystallization of 0.95, while NPK 25-07-07 had the lowest with 0.934. Taking into consideration that particles need to be ca. 80-85 % crystallized to survive the impact with the tower bottom, it is possible to conclude that all of the products above would survive, given a particle size of 2.85 mm. Considering the simplifications and assumptions made, it is hard to conclude how the chemical composition influence the heat transfer and crystallization. One assessment that could be made by looking at the product recipes (confidential), and comparing them to the calculated crystallization values is that NPK 21-4-10 has the lowest wt % of AN+MAP in the total particle composition, and show greatest crystallization. NPK 25-07-07 has the highest wt % of AN+MAP, and show least crystallization. This could be due to how the heat capacity and density is changed due to addition of other products, which would increase/decrease heat transfer and T.O.F. However, without further chemical and thermodynamical data regarding the specific focus grades, it is difficult to make conclusions.



## 8 Conclusion

An updated velocity model was designed to be implemented as part of a larger heat-and mass transfer model. The velocity model gave results that strongly resembled previously calculated values. Prill trajectories were calculated for different particle sizes, and the distance from the tower wall ranged from 3.7-13 m, with the smaller particles ending up the closest to the center. This distance was also calculated as a function of RPM, keeping the particle size constant at 2.85 mm. The results show distances from the tower wall ranging from 5.7-7.8 m, while varying the RPM from 200-400 respectively. The time-of-flight for different particle sizes were calculated using the values in table 6.1, and the results ranged from 13.58 s to 6.16 s, given particles in the size range 1-6 mm, with the smallest particle having the longest time-of-flight.

The velocity model was incorporated into the composite Ytterbø model, where the temperature distribution of a prill, and the total heat release was calculated and compared to an older reference model. The model was run using parameters and measurements taken from Stokka, made on 5/6-84. The composite Ytterbø model calculated a time-of-flight of 8.09 s. The temperature as the particle hit bottom was calculated for the core, critical radius and the surface. The temperatures were 102.8, 67.5 and 35.5°C respectively. The total heat released from the prills was calculated to be  $18.21 \times 10^6$  kJ/h, which, compared to the older models calculated value of  $16.43 \times 10^6$  kJ/h represents an error reduction from 10.89% to 1.30% when comparing to the reference value of  $18.33 \times 10^6$  kJ/h.

Performing a sensitivity analysis on the composite Ytterbø model, using a set of reference values showed that particle diameter and thermal conductivity had the overall largest impacts on the temperature distribution. Particle diameter influences were calculated to be 24.3 °C/mm for the core, 25.4 °C/mm for the critical radius and 11.606 °C/mm for the surface. Thermal conductivity influences were calculated to be  $17.8^\circ\text{C}\left(\frac{\text{W}}{\text{m}^2\text{K}}\right)$  for the core,  $27.7^\circ\text{C}\left(\frac{\text{W}}{\text{m}^2\text{K}}\right)$  for the critical radius and  $7.2^\circ\text{C}\left(\frac{\text{W}}{\text{m}^2\text{K}}\right)$  for the surface. Air temperature influence increases as radius increases, while air pressure influence decreases as radius increases. Particle density only affects prill temperature in a marginal way, but does significantly impact T.O.F, which increases total heat released. Two ways were developed to combine process parameter changes and temperature values; expected average difference method (EAD) and the trend-line method.

The composite Ytterbø model was adapted to simulate the prill temperature and crystallization degrees in four different focus grades. The model was updated based on the

products composition, which in turn altered the heat capacities and crystallization intervals. Remaining parameters were kept identical to the reference model. Out of the four focus grades, NPK 21-4-10 S+B presented the lowest core temperature of 79.3°C and highest crystallization percentage of 95 %. NPK 25-07-07 presented the highest core temperature of 93.6°C and lowest crystallization percentage of 93.4 %.

## 9 Further work

Having reviewed the composite Ytterbø model, its components and the results, there are a couple of changes that should be made in order to ensure proper functionality. These changes can be divided into two categories; model upgrades and product data. The model upgrades will be presented first, followed by the product data upgrades.

### 9.1 Model upgrades

- Implement Rayleigh's Jet Theory to predict particle sizes depending on centrifuge and liquid conditions.
  - This would allow for additional parameter investigation. Viscosity and centrifuge speed should be further evaluated.
- Implement air temperature changes in the tower.
  - Quick solution: Logarithmic-mean-temperature difference (LMTD) method. This can allow the model to account for the difference between inlet and outlet temperature, while neglecting any gradients inside the tower.
  - Robust solution: Implement heat- and mass-transfer across the surface of the particle to account for moisture and temperature changes in the air. Validation of such a model would require temperature logging at different heights.
- Extend velocity model into three dimension to account for spiraling movement.
- Calculate  $h$  for every time-step, instead of averaging it over the entire tower height.
  - Although this might seem like a simple solution, calculating a new convection coefficient each time-step will cause each time-step to have a corresponding Biot number. Running a model such as this, within a reasonable time-frame is only feasible if number of roots are heavily diminished. Assessment of how many roots would be needed to make the model accurate "enough" would be a reasonable next step.

## 9.2 Product data upgrades

As previously mentioned, this model makes many assumptions and simplifications in regards to specific product crystallization behavior. This could be eliminated by analyzing the complex chemistry and thermodynamics behind a given product. Hopefully, many of the points below can be addressed by using computational modelling of the system, through FactSage, Aspen or other software. Gathering the data below is **crucial** for accurate product assessment.

- Accurate assessments of the thermal conductivity of salt-mixes. The thermal conductivity greatly influences the temperature development in the prill, and it would therefore be beneficial to run experiments or mathematical models to estimate a more accurate conductivity of the solid salt compounds in the different products.
- Find accurate heat of crystallization values for different products, based on relevant phase diagrams, or computational models.
- Account for MOP, SOP, Poly-P and how they influence the crystallization parameters in specific products.

# Bibliography

- [1] T. Kjærgaard, *A Plant that Changed the World: the rise and fall of clover 1000–2000*, 2016. [Online]. Available: <https://thorkildkjaergaard.com/a-plant-that-changed-the-world-the-rise-and-fall-of-clover-1000-2000/>.
- [2] M. Mazoyer, *A history of world agriculture : From the neolithic age to the current crisis*, eng, New York, 2006.
- [3] N. Cherkasov, A. Ibhaddon and P. Fitzpatrick, ‘A review of the existing and alternative methods for greener nitrogen fixation,’ eng, *Chemical engineering and processing*, vol. 90, pp. 24–33, 2015, ISSN: 0255-2701.
- [4] F. T. Nielsson, ‘Manual of Fertilizer Processing,’ 2018, ISSN: 9781351433853 1351433857 9781351433846 1351433849. [Online]. Available: <https://doi.org/10.1201/9780203748183>.
- [5] *Yara porsgrunn*. [Online]. Available: <https://www.yara.no/om-yara/om-yara-norge/porsgrunn/>.
- [6] P. Stokka, ‘PRILLING. en enkel beregnings og forståelsesmodell for kjøling og krystallisering av prills i prilletårn,’ Unpublished Work, Norsk Hydro, 1989.
- [7] A. Alamdari, A. Jahanmiri and N. Rahmaniyan, ‘Mathematical modelling of urea prilling process,’ *Chemical Engineering Communications*, vol. 178, no. 1, pp. 185–198, 2000, ISSN: 0098-6445. DOI: 10.1080/00986440008912182. [Online]. Available: <https://doi.org/10.1080/00986440008912182>.
- [8] A. O. Gezerman, ‘Mathematical modeling for prilling processes in ammonium nitrate production,’ *Engineering Reports*, vol. 2, no. 6, e12173, 2020. DOI: <https://doi.org/10.1002/eng2.12173>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/eng2.12173>.
- [9] A. Mehrez, P. D. E. A. H. H. Ali, W. Zahra, S. Ookawara and M. Suzuki, ‘Study on Heat and Mass Transfer During Urea Prilling Process,’ *Int. J of Chemical Engineering and Applications*, vol. 3, pp. 347–353, 2012. DOI: 10.7763/IJCEA.2012.V3.216.
- [10] W. H. Ross, A. R. Merz and K. D. Jacob, ‘Preparation and Properties of the Ammonium Phosphates,’ *Industrial & Engineering Chemistry*, vol. 21, no. 3, pp. 286–289, 1929, ISSN: 0019-7866. DOI: 10.1021/ie50231a029. [Online]. Available: <https://doi.org/10.1021/ie50231a029>.

- [11] M. Gargouri, C. Chtara, P. Sharrock, A. Nzihou and H. Elfeki, 'Experimental Study of the Purification of an Industrial Fertilizer (Mono-Ammonium Phosphate) to Larger Scale Using an Experimental Design,' *International Journal of Materials Engineering*, vol. 2, pp. 32–37, 2012. DOI: 10.5923/j.ijme.20120204.01.
- [12] G. Kongshaug, B. A. Brentnall, K. Chaney *et al.*, 'Phosphate Fertilizers,' in *Ullmann's Encyclopedia of Industrial Chemistry*. 1991, vol. A19. DOI: [https://doi.org/10.1002/14356007.a19\\_421](https://doi.org/10.1002/14356007.a19_421). [Online]. Available: [https://onlinelibrary.wiley.com/doi/abs/10.1002/14356007.a19\\_421](https://onlinelibrary.wiley.com/doi/abs/10.1002/14356007.a19_421).
- [13] N. J. Tro, *Chemistry: A molecular approach*. Pearson Education, Inc, 2020.
- [14] H. Kiiski, 'Properties of Ammonium Nitrate based fertilisers,' 2009.
- [15] T. P. Hignett, *Fertilizer manual*, ser. Developments in plant and soil sciences. Dordrecht ; Boston: Nijhoff/W. Junk Publishers for the International Fertilizer Development Center ; Hingham, MA : Distributors for the U.S. and Canada, Kluwer Academic Publishers, 1985, vol. v. 15, ISBN: 9024731224.
- [16] 'Production of NPK fertilizer by the nitrophosphate route,' *Best Available Techniques for Pollution Prevention and Control in the European Fertilizer Industry*, vol. 7, 2000.
- [17] L. Rayleigh, 'On The Instability Of Jets,' *Proceedings of the London Mathematical Society*, vol. s1-10, no. 1, pp. 4–13, 1878. DOI: <https://doi.org/10.1112/plms/s1-10.1.4>. eprint: <https://londmathsoc.onlinelibrary.wiley.com/doi/pdf/10.1112/plms/s1-10.1.4>. [Online]. Available: <https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s1-10.1.4>.
- [18] G. Ricardo, D. Noriler, W. Martignoni and H. F. Meier, *Eulerian-Lagrangian Analysis of Multiphase Flow in Urea Prilling Process with Phase Changing*. 2013, vol. 32, pp. 2173–2178, ISBN: 978-88-95608-23-5. DOI: 10.3303/CET1332363.
- [19] R. J. Ribando, *Heat transfer tools*. New York: McGraw-Hill Publishing, 2001, ISBN: 0072479620 9780072479621 0072463287 9780072463286.
- [20] P. Stokka, 'FULLGJØDSEL. anvendelse av smeltekjemi for å forstå/beregne væskefaststofforhold i nitrofosfatprosessens saltsmelter.,' Unpublished Work, Norsk Hydro, 1985.
- [21] ———, 'FULLGJØDSEL. tidsavhengig viskositet i inndampede NP—smelter. viskositetsforhold i inndampede NP smelter tilsatt KCl.,' Unpublished Work, Norsk Hydro, 1986.
- [22] Y. Wu, C. Bao and Y. Zhou, 'An Innovated Tower-fluidized Bed Prilling Process,' *Chinese Journal of Chemical Engineering - CHINESE J CHEM ENG*, vol. 15, pp. 424–428, 2007. DOI: 10.1016/S1004-9541(07)60102-5.
- [23] B. S. Massey, *Mechanics of Fluids*, 6th ed. London, England: Van Nostrand Reinhold, 1989, ISBN: 0278000479 0412342804.

- [24] D. Willem, *Projectile motion with air resistance*, Electronic Article.
- [25] T. Bergman, L. Incropera and F. P., *Fundamentals of Heat and Mass Transfer*. Hoboken, NJ: Wiley, 2011, ISBN: 9780470501979 0470501979.
- [26] S. Linge and H. P. Langtangen, *Programming for Computations – Python*. Springer Open, 2020, ISBN: 978-3-030-16876-6. DOI: <https://doi.org/10.1007/978-3-030-16877-3>.
- [27] Y. A. Engel, *Heat transfer: A practical approach*. New York: McGraw-Hill, 2004, ISBN: 0071236449 9780071236447.
- [28] *Prandtl number - Heat Transfer Engineering and Design*, 2020. [Online]. Available: [https://www.engineersedge.com/heat\\_transfer/prandtl\\_number\\_13954.htm](https://www.engineersedge.com/heat_transfer/prandtl_number_13954.htm).
- [29] D. J. Acheson, *Elementary Fluid Dynamics*. Clarendon Press, 1990, p. 134.
- [30] W. Ranz and W. Marshall, 'Evaporation of drops,' *Chemical engineering progress*, vol. 48, pp. 141–146, 1952.
- [31] C. M. v. Land, 'Industrial crystallization of melts,' 2005, ISSN: 0203021134 9780203021132. [Online]. Available: <https://doi.org/10.1201/9780203021132>.
- [32] G. Recktenwald, *Transient, One-Dimensional Heat Conduction in a Convectively Cooled Sphere*, Electronic Article, 2006.
- [33] L. M. Jiji, *Heat Conduction*, Electronic Book, 2009.
- [34] S. N. Saleh, S. M. Ahmed, D. Al-mosuli and S. Barghi, 'Basic design methodology for a prilling tower,' *The Canadian Journal of Chemical Engineering*, vol. 93, no. 8, pp. 1403–1409, 2015, ISSN: 0008-4034. DOI: <https://doi.org/10.1002/cjce.22230>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cjce.22230>.
- [35] J. Taler and P. Oćłoń, 'Transient Heat Conduction in Sphere,' in 2014, pp. 6186–6198, ISBN: 978-94-007-2738-0. DOI: 10.1007/978-94-007-2739-7\_400.
- [36] J. VanSant and L. L. N. Laboratory, *Conduction Heat Transfer Solutions*. Lawrence Livermore National Laboratory, 1983. [Online]. Available: <https://books.google.no/books?id=JS1jPAAACAAJ>.
- [37] D. A. D. Polyanin, W. E. Schiesser and A. I. Zhurov, *Partial differential equation*, 2008. [Online]. Available: [http://www.scholarpedia.org/article/Partial\\_differential\\_equation](http://www.scholarpedia.org/article/Partial_differential_equation).
- [38] E. G. Wiggins, 'Transient Heat Conduction in a Sphere,' *ASEE CoED Journal*, vol. 1, no. 4, 2010.
- [39] D. D. Wagman, *The NBS tables of chemical thermodynamic properties : selected values for inorganic and C1 and C2 organic substances in SI units*, ser. Journal of physical and chemical reference data. Washington, D.C: Published by the American Chemical Society and the American Institute of Physics for the National Bureau of Standards, 1982, vol. 11(1982). Supplement ; 2, ISBN: 0883184176.

- [40] J. Smith, H. V. Ness, M. Abbott and M. Swihart, *Introduction to Chemical Engineering Thermodynamics*. McGraw-Hill Education, 2017, ISBN: 9781259921896.
- [41] J. Ulrich and T. Stelzer, ‘Melt Crystallization,’ in *Handbook of Industrial Crystallization*, A. Y. Lee, A. S. Myerson and D. Erdemir, Eds., 3rd ed. Cambridge: Cambridge University Press, 2019, pp. 266–289, ISBN: 9780521196185. DOI: DOI : 10 . 1017 / 9781139026949 . 009. [Online]. Available: <https://www.cambridge.org/core/books/handbook-of-industrial-crystallization/melt-crystallization/E84EAF65BEE0ACB09637BB6E99BE200E>.
- [42] V. B. John, *Understanding phase diagrams*. London: Macmillan, 1974, ISBN: 0333149890 9780333149898.
- [43] D. R. F. West, *Ternary equilibrium diagrams*, 2nd ed. London: Chapman and Hall, 1982, ISBN: 0412229706,0412228203.
- [44] I. Ymen, ‘Granuleringsteori för NPK-produkter del 4: Sammansättningen på den eutektiska lösningen  $\text{NH}_4\text{NO}_3\text{-NH}_4\text{Cl-NH}_4\text{H}_2\text{PO}_4\text{-KNO}_3\text{-H}_2\text{O}$ ,’ Tech. Rep., 1987.
- [45] M. Skydanenko, V. Sklabinskyi, A.-K. M. Nadhem and K. Nichvolodin, ‘Determination of granule (prill) movement modes in the prilling tower for mineral fertilizer production,’ *Technology audit and production reserves*, vol. 5, no. 3(61), pp. 6–9, 2021. DOI: 10 . 15587 / 2706 - 5448 . 2021 . 241142. [Online]. Available: <http://journals.uran.ua/tarp/article/view/241142>.
- [46] I. Hugstmyr, ‘Måling av spesifikk varmekapasitet på NP og NPK i temperaturintervallet 25-117°C,’ Unpublished Work, Norsk Hydro, 1987.
- [47] K. Krabbenhoft, L. Damkilde and M. Nazem, ‘An implicit mixed enthalpy–temperature method for phase-change problems,’ *Heat and Mass Transfer/Waerme- und Stoffuebertragung*, vol. 43, pp. 233–241, 2007. DOI: 10.1007/s00231-006-0090-1.
- [48] A. Farrokhpahanah, M. Bussmann and J. Mostaghimi, *New smoothed particle hydrodynamics (SPH) formulation for modeling heat conduction with solidification and melting*, 2017.
- [49] V. R. Voller and C. Prakash, ‘A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems,’ *International Journal of Heat and Mass Transfer*, vol. 30, no. 8, pp. 1709–1719, 1987, ISSN: 0017-9310. DOI: [https://doi.org/10.1016/0017-9310\(87\)90317-6](https://doi.org/10.1016/0017-9310(87)90317-6). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0017931087903176>.
- [50] T. Shengwei, G. Hui, Y. Jiankang and L. Bin, ‘Physicochemical Properties of Acidic Ammonium Phosphate Slurries,’ *Industrial & Engineering Chemistry Research*, vol. 43, no. 12, pp. 3194–3199, 2004, ISSN: 0888-5885. DOI: 10.1021/ie030839y. [Online]. Available: <https://doi.org/10.1021/ie030839y>.



# **Appendix A**

## **Task description**

## FMH606 Master's Thesis

**Title:** Understanding influential process parameters of NPK fertilizer production; A study on the prilling process in the NPK3 prilling tower (POR)

**USN supervisor:** Chandana Ratnayake

**External partners:** Jo-Ela Johansen (Yara Technology and Project), Fredrik Rodahl (Yara Technology and Project) and Franz Otto von Hafenbrädl (SINTEF)

### **Task background:**

Yara Porsgrunn produces and exports NPK fertilizer around the world. Production is based on the nitrophosphate route where phosphate rock is dissolved in nitric acid and cooled to crystallize and filter off CN (calcium nitrate) crystals. The remaining (mother) liquor is neutralized with ammonia, water evaporated, and AN (ammonium nitrate) is added to adjust N/P ratio in the product. NP-liquor is then mixed with other salts like MOP (potassium chloride), SOP (potassium sulphate), AS (ammonium sulphate), kieserite, micronutrients and fillers according to specific NPK product recipes. Particulation is achieved by prilling in high prill towers where droplets form and crystallize upon cooling with countercurrent ambient air as it falls to the prill tower bottom. Ideally all prills should be perfectly spherical. Some prills will however be miss-shaped due to several reasons.

For decades there has been extensive work and testing to understand relevant process parameters and factors influencing prilling and non-spherical particles. The challenges are, however, still present and attention required to optimize relevant processes. In order to increase capacity utilization and improve prill quality in existing NPK plants at Yara Porsgrunn it is important to enhance the understanding on how relevant process parameters influence the prilling process. Several process parameters and factors that influence or may influence prill product NSP directly are identified. Several critical factors are however expected to be product dependent.

The present study aims to enhance knowledge by establishing **prill tower mass & energy balance calculations** and focusing on **the influence of salt PSD** (particle size distribution) in sphericity of the final NPK products to support further optimization of the whole prilling process.

### **Task description:**

- Review previous study reports on prill tower mass & energy balance (Stokka, 1989) and propose appropriate upgrades based on available State-of-Art models/technologies.
- Investigate crystallization and propose associated heat of crystallization and Cp values for prilling of four specified focus grades to enable use in mass & energy balance modelling.
- Develop and program a mass & energy balance model for NPK3 prilling tower (POR) for the four specified focus grades with prill size and ambient air temperature as a variable.
- Assess product quality by analyzing particle size distribution and particle characteristics and compare measurements to values predicted by the model based on different prill tower loads and ambient temperatures for the four specified focus grades.

- Perform a sensitivity analysis to understand to what extent natural variations in chemical composition and salt PSD influence cooling limitation for the four specified focus grades.

**Student category:** PT (EET, EPE, IIA or PT students)

**The task is suitable for online students (not present at the campus):** No  
(The topic is reserved for Anders Aaland Ytterbø)

**Practical arrangements:**

Data samples from different measurement principles will be available from the NPK prill stack. SINTEF will provide access to powder characterisation test facilities and relevant training.

**Supervision:**

As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

**Signatures:**

Supervisor (date and signature): 30.01.22



J. Einar Jørgensen  
2/2-2022



Student (write clearly in all capitalized letters): Anders Ytterbø

Student (date and signature):



28.01.22



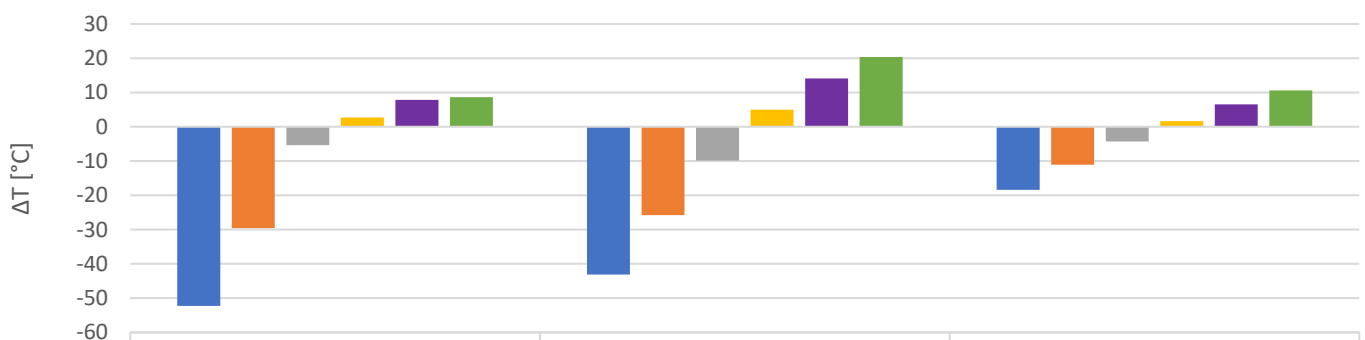
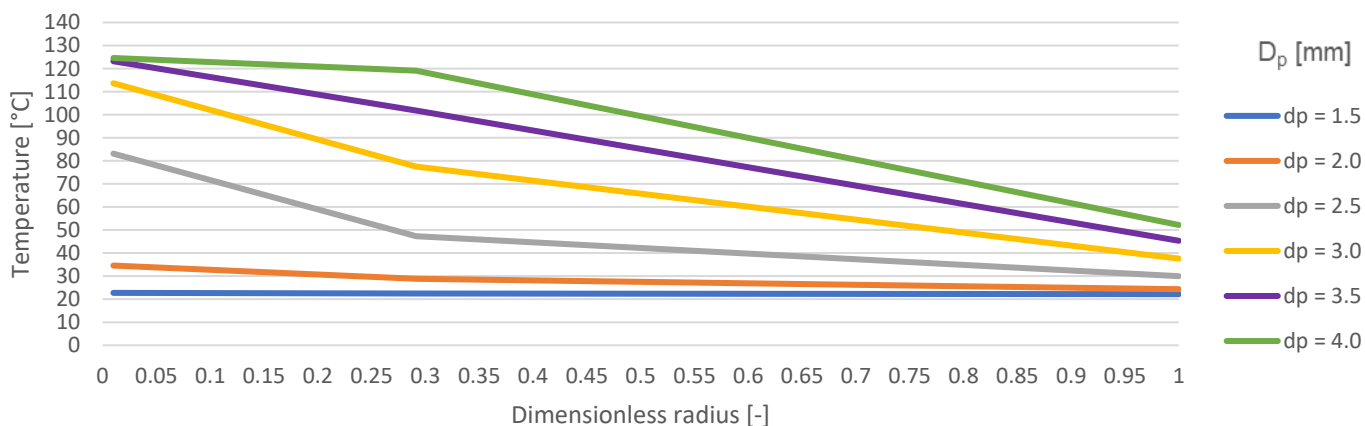
# **Appendix B**

## **Sensitivity analysis data**

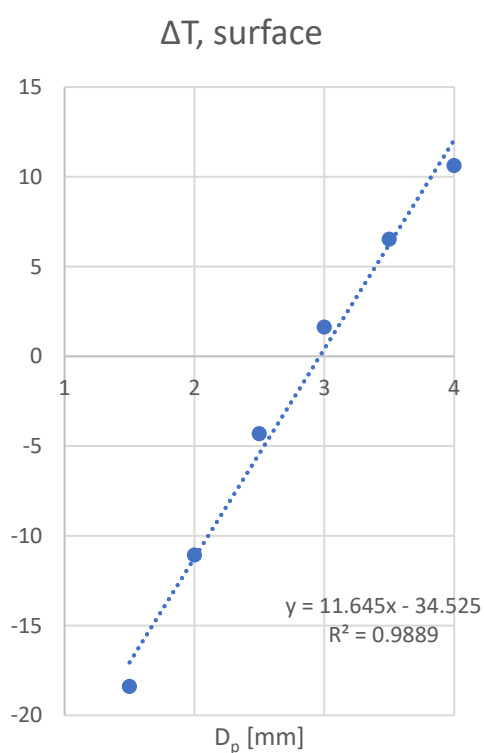
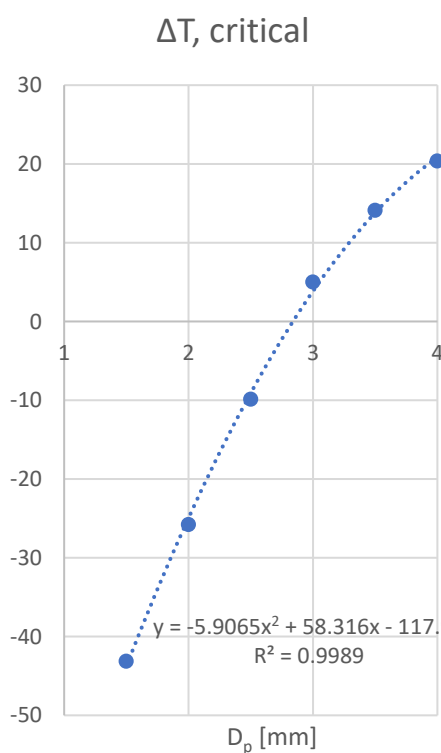
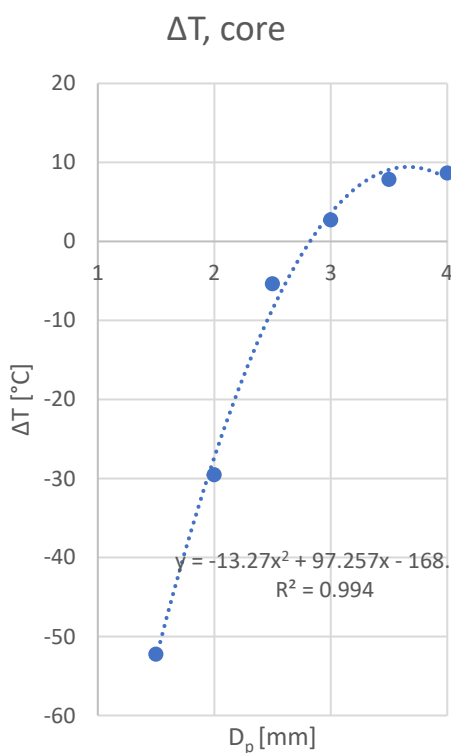
### **B.1 Summary data**

# Particle diameter

## Temperature profile, tower bottom

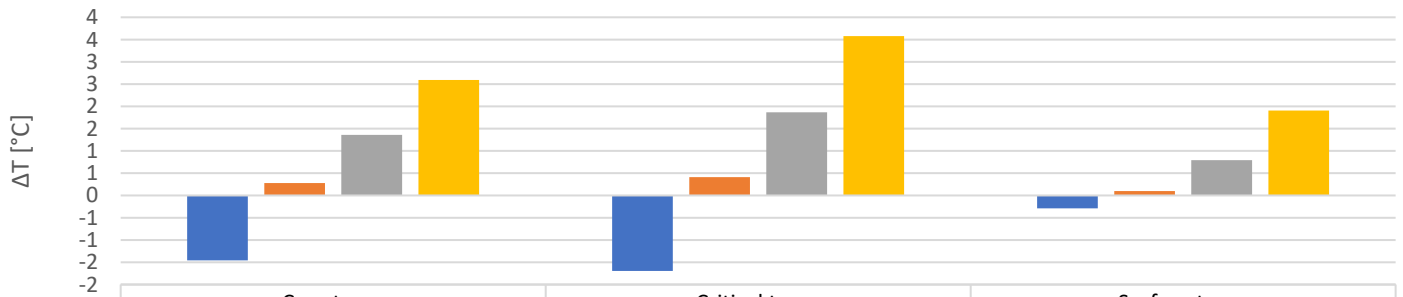
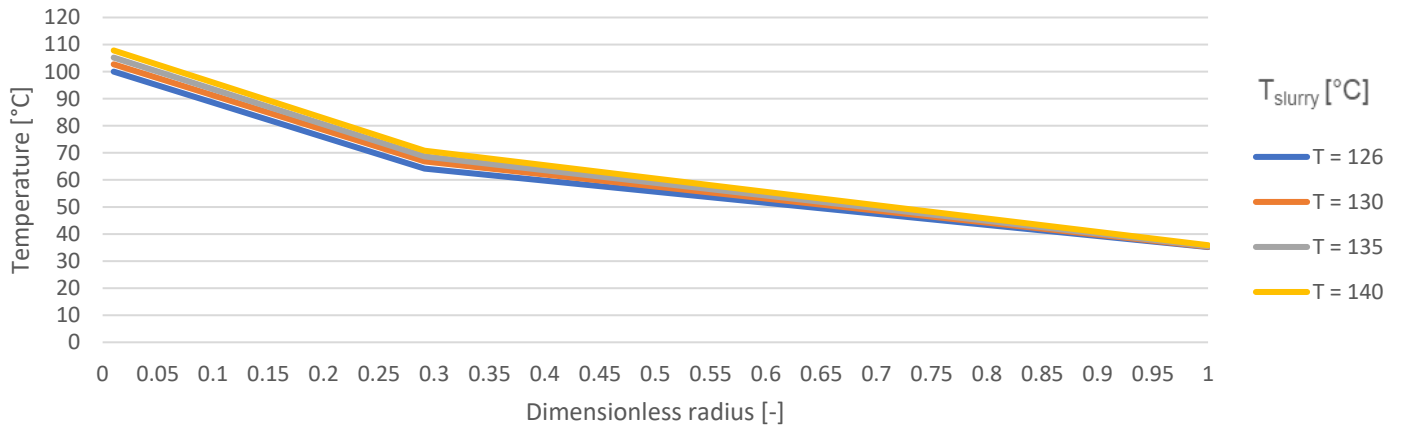


	Core temp	Critical temp	Surface temp
Dp = 1.5	-52.275	-43.127	-18.393
Dp = 2.0	-29.562	-25.787	-11.072
Dp = 2.5	-5.373	-9.869	-4.312
Dp = 3.0	2.698	5.000	1.627
Dp = 3.5	7.825	14.120	6.520
Dp = 4.0	8.632	20.361	10.622



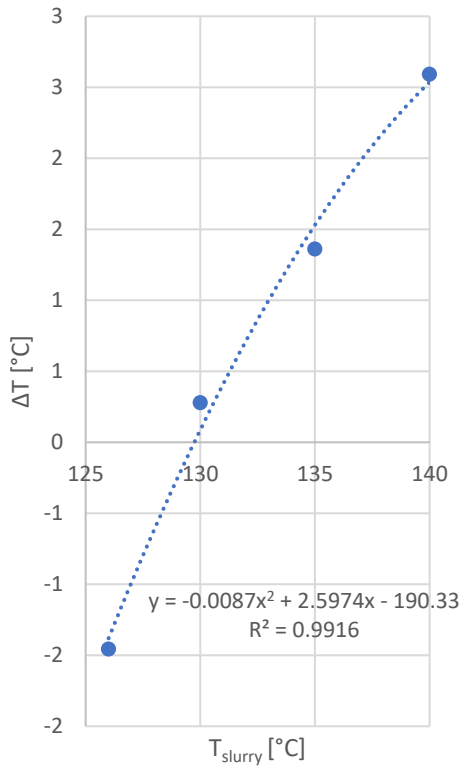
# Slurry temperature

Temperature profile, tower bottom

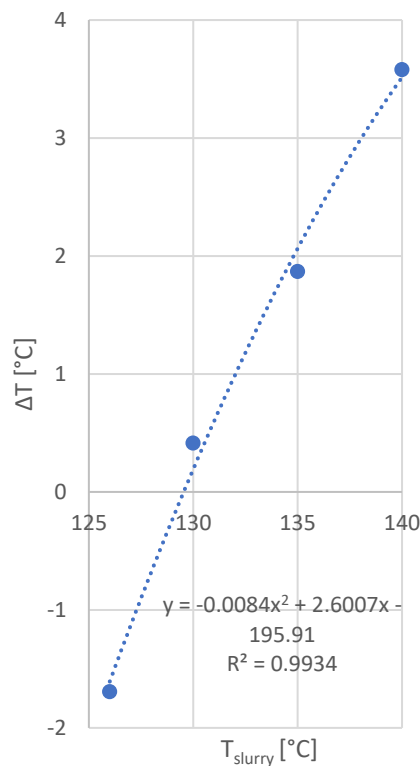


	Core temp	Critical temp	Surface temp
■ T = 126	-1.456	-1.695	-0.291
■ T = 130	0.278	0.413	0.099
■ T = 135	1.360	1.867	0.791
■ T = 140	2.591	3.578	1.907

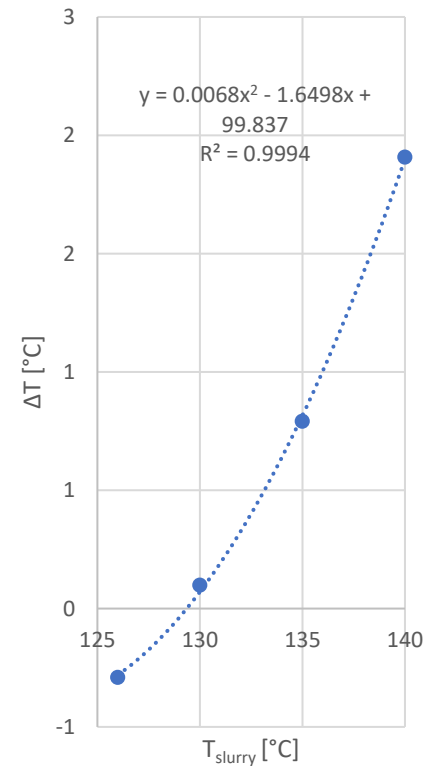
ΔT, core



ΔT, critical

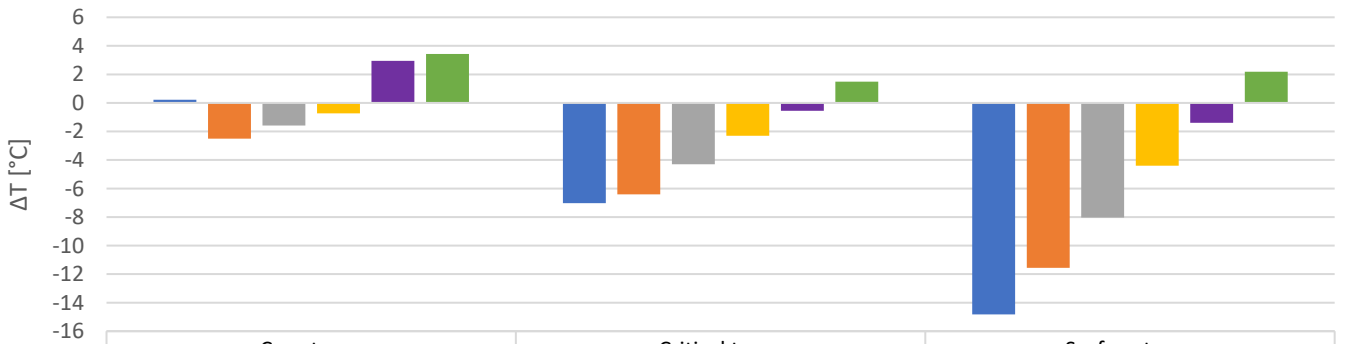
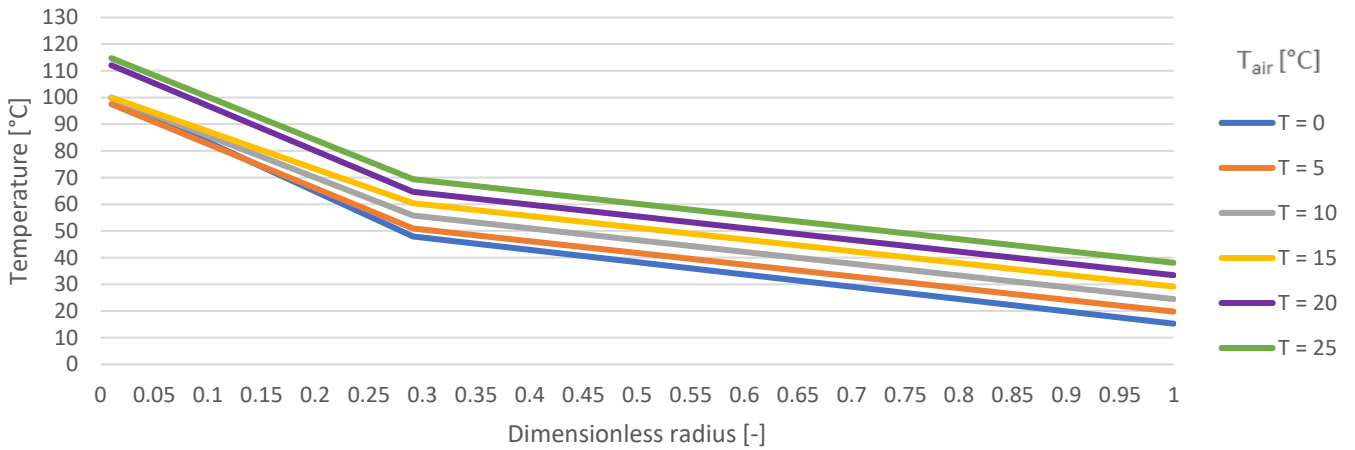


ΔT, surface

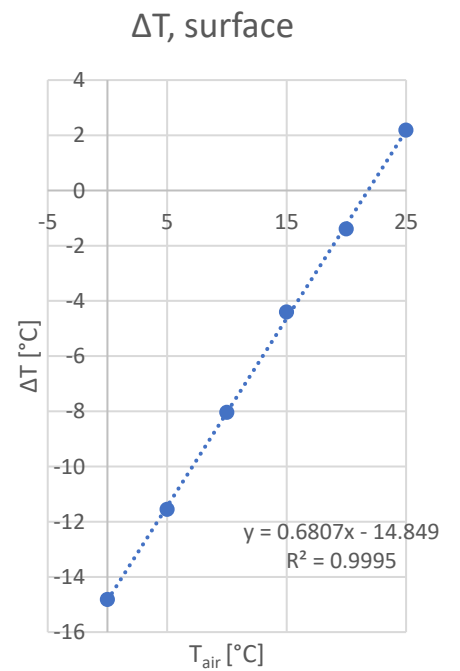
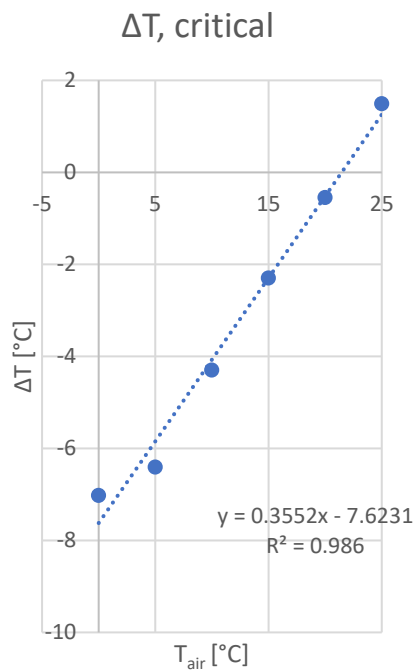
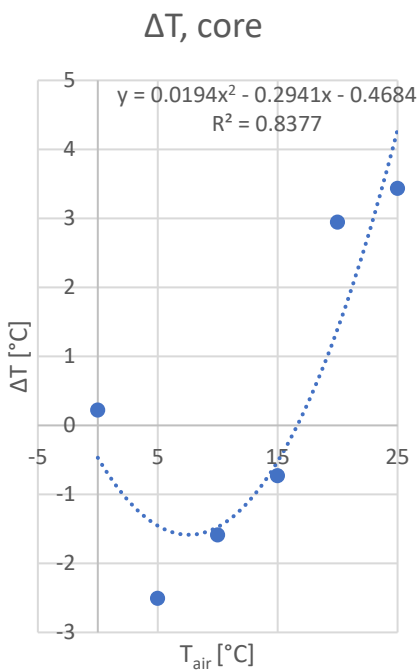


# Ambient air temperature

## Temperature profile, tower bottom



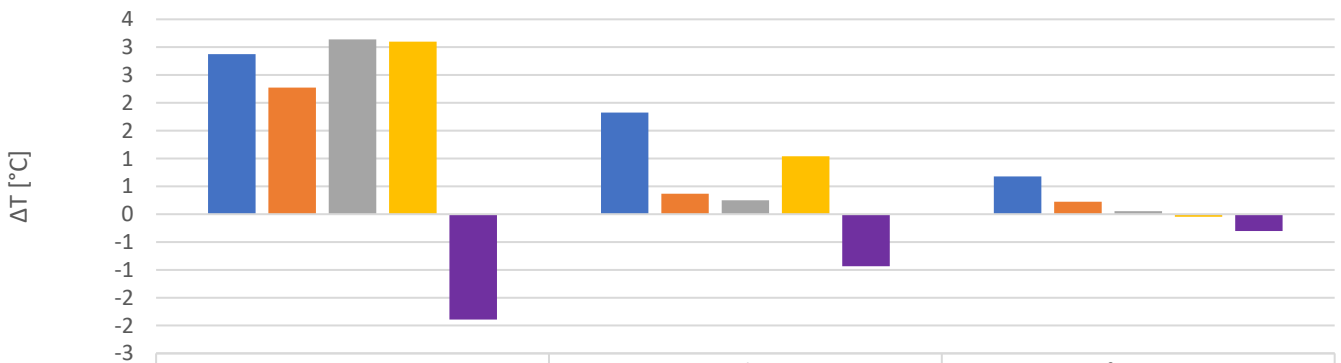
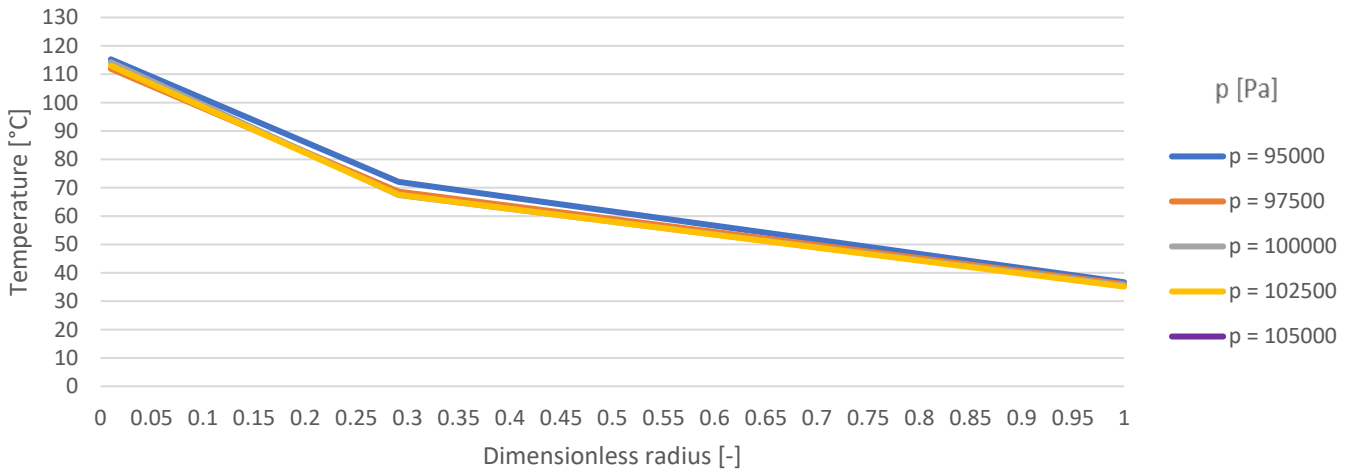
	Core temp	Critical temp	Surface temp
T = 0	0.220	-7.026	-14.820
T = 5	-2.508	-6.411	-11.558
T = 10	-1.589	-4.301	-8.041
T = 15	-0.732	-2.299	-4.405
T = 20	2.944	-0.549	-1.398
T = 25	3.433	1.488	2.182



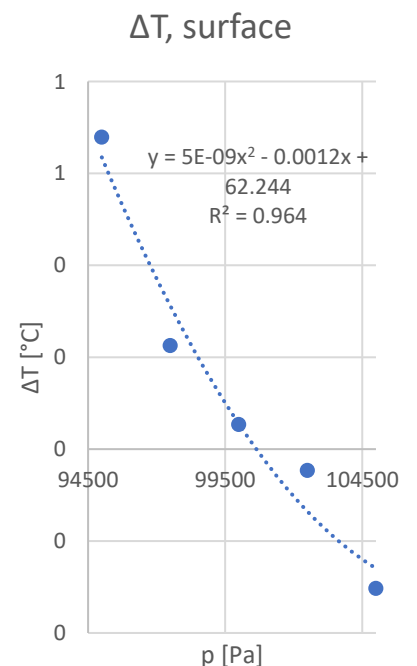
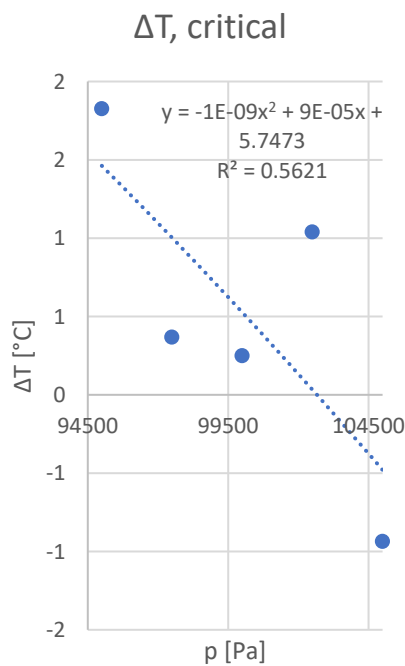
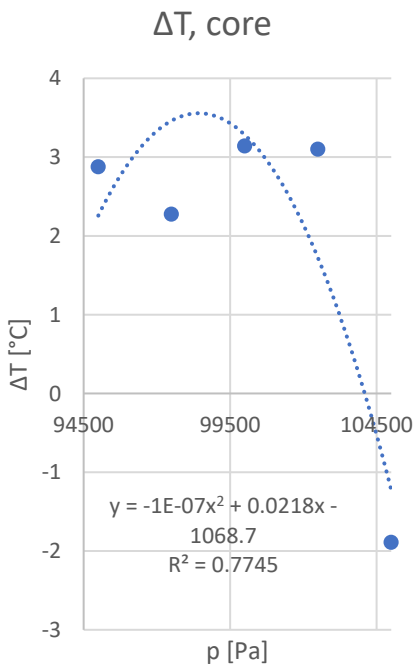


# Ambient air pressure

## Temperature profile, tower bottom

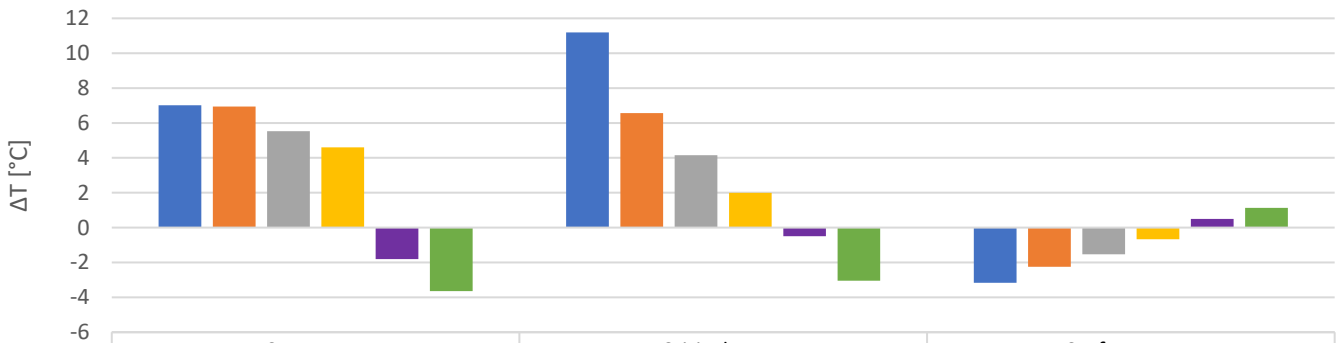
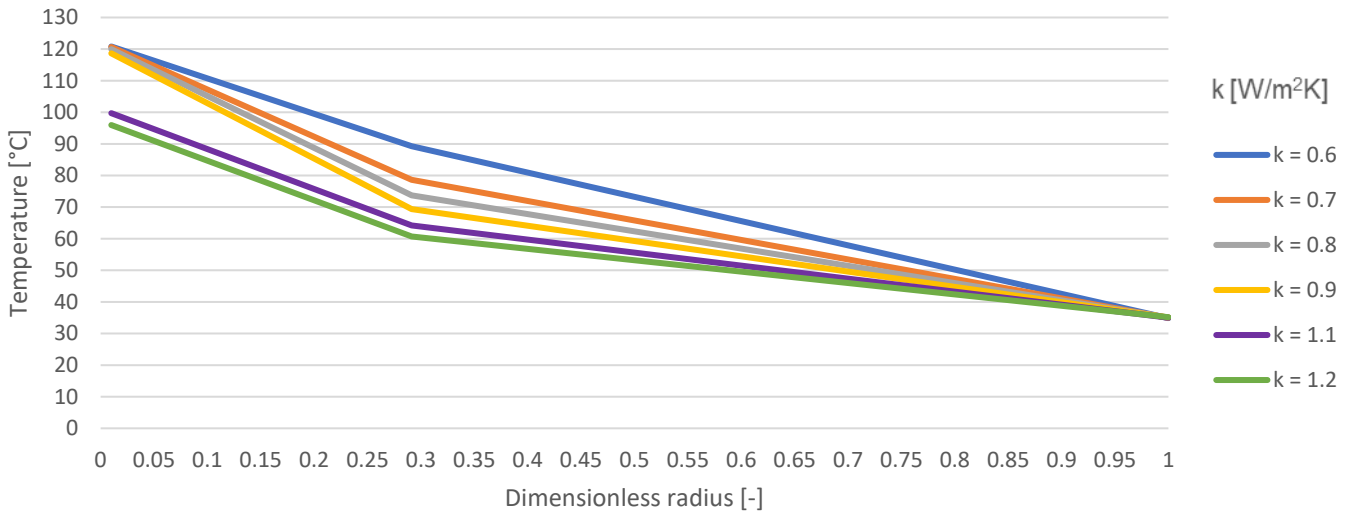


	Core temp	Critical temp	Surface temp
■ p = 95000	2.876	1.826	0.679
■ p = 97500	2.273	0.367	0.225
■ p = 100000	3.140	0.249	0.053
■ p = 102500	3.099	1.039	-0.047
■ p = 105000	-1.892	-0.936	-0.303

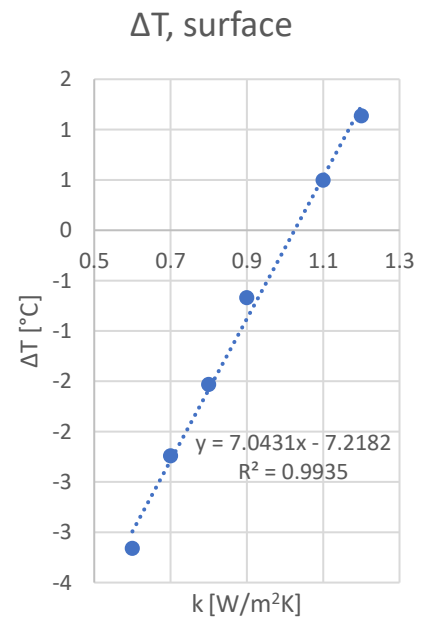
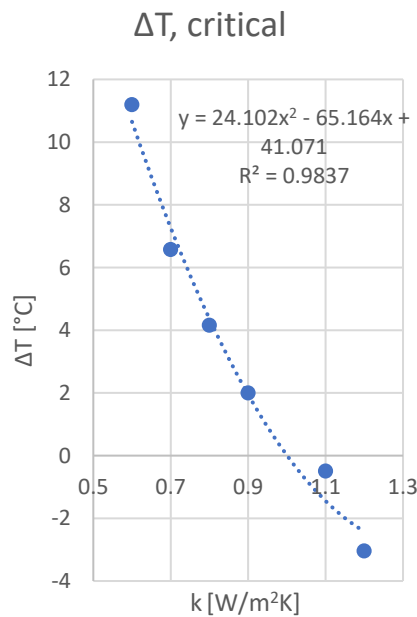
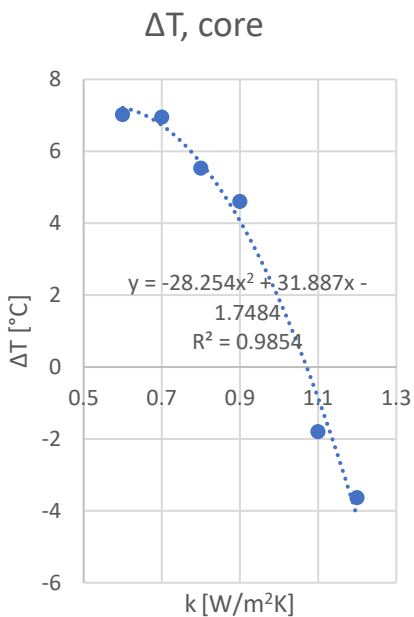


# Thermal conductivity

Temperature profile, tower bottom

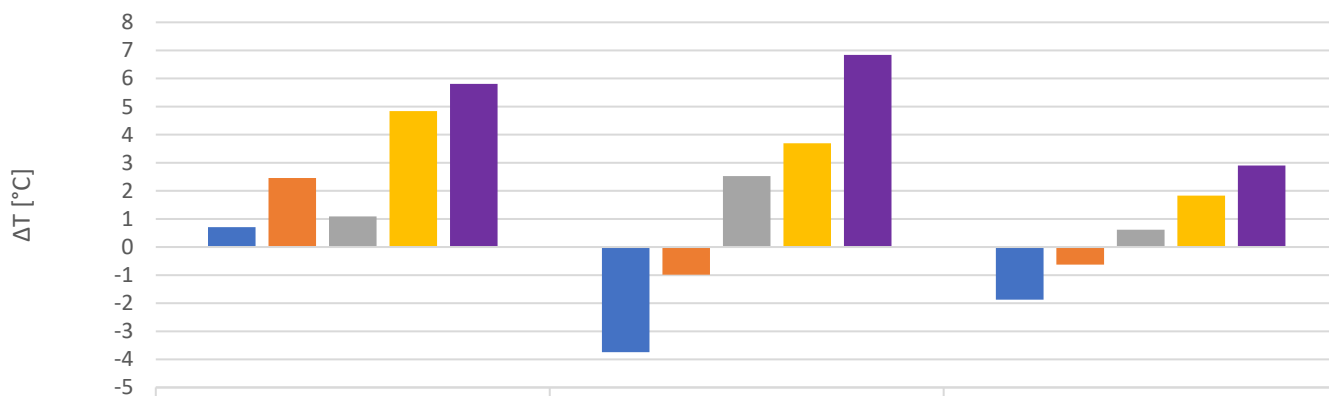
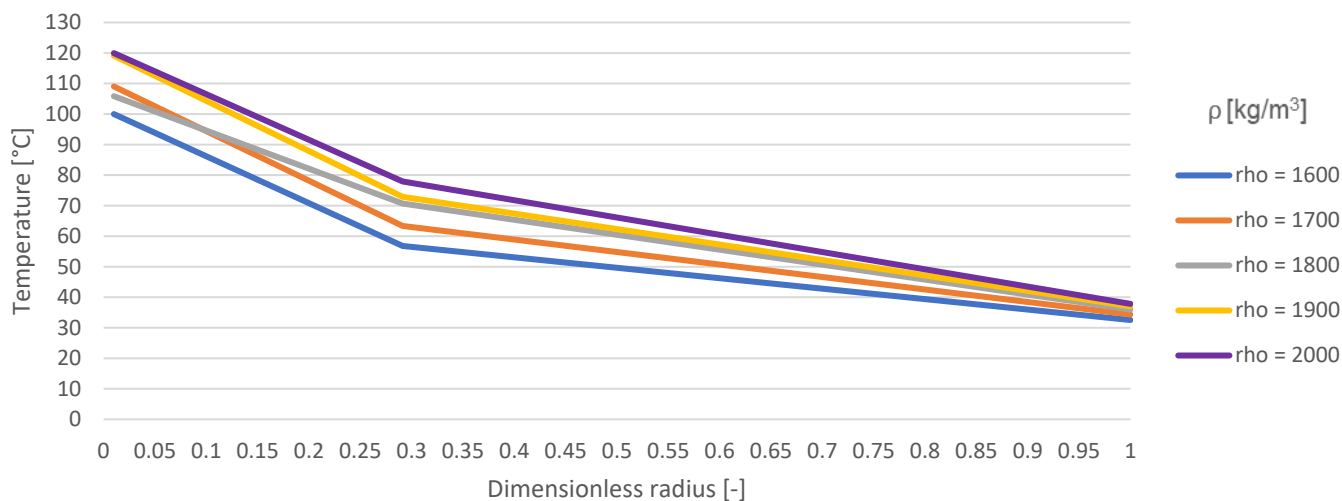


	Core temp	Critical temp	Surface temp
■ k = 0.6	7.019	11.190	-3.161
■ k = 0.7	6.946	6.568	-2.243
■ k = 0.8	5.527	4.153	-1.533
■ k = 0.9	4.600	1.995	-0.672
■ k = 1.1	-1.805	-0.496	0.495
■ k = 1.2	-3.639	-3.047	1.134



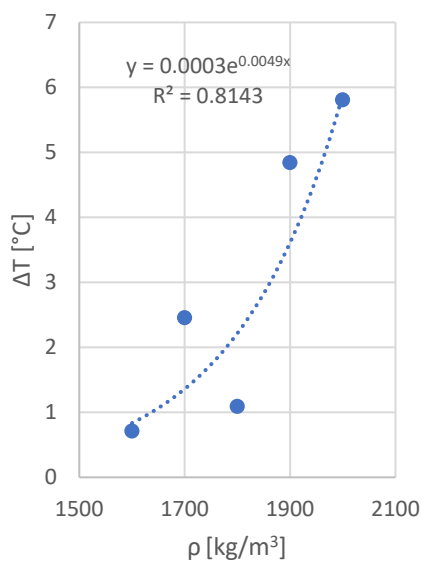
# Particle density

## Temperature profile, tower bottom

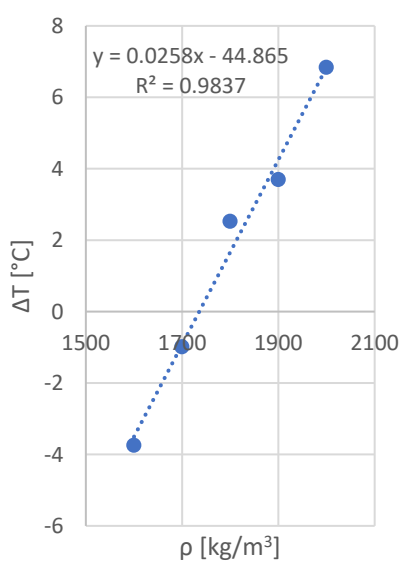


	Core temp	Critical temp	Surface temp
■ rho = 1600	0.710	-3.746	-1.872
■ rho = 1700	2.455	-0.987	-0.625
■ rho = 1800	1.092	2.523	0.619
■ rho = 1900	4.841	3.694	1.831
■ rho = 2000	5.807	6.838	2.905

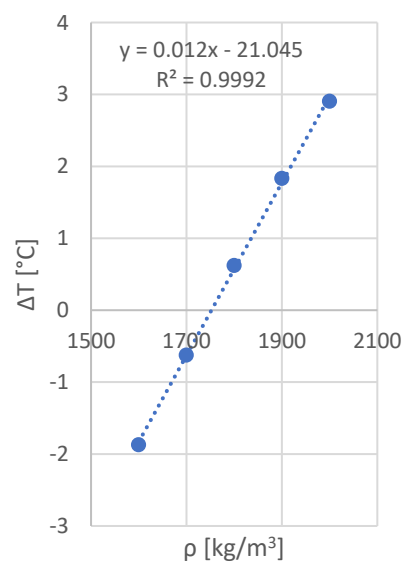
### ΔT, core



### ΔT, critical



### ΔT, surface

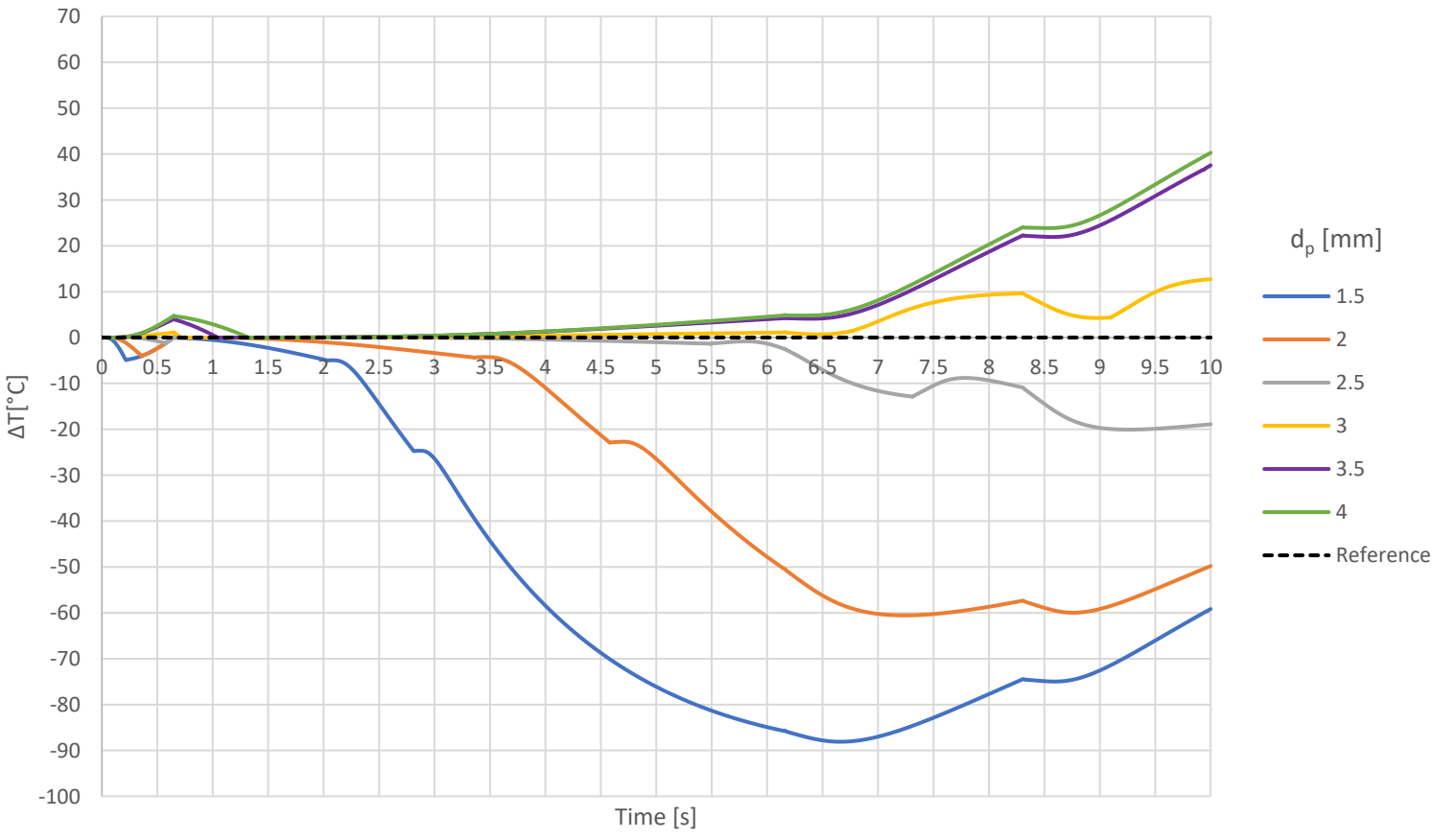
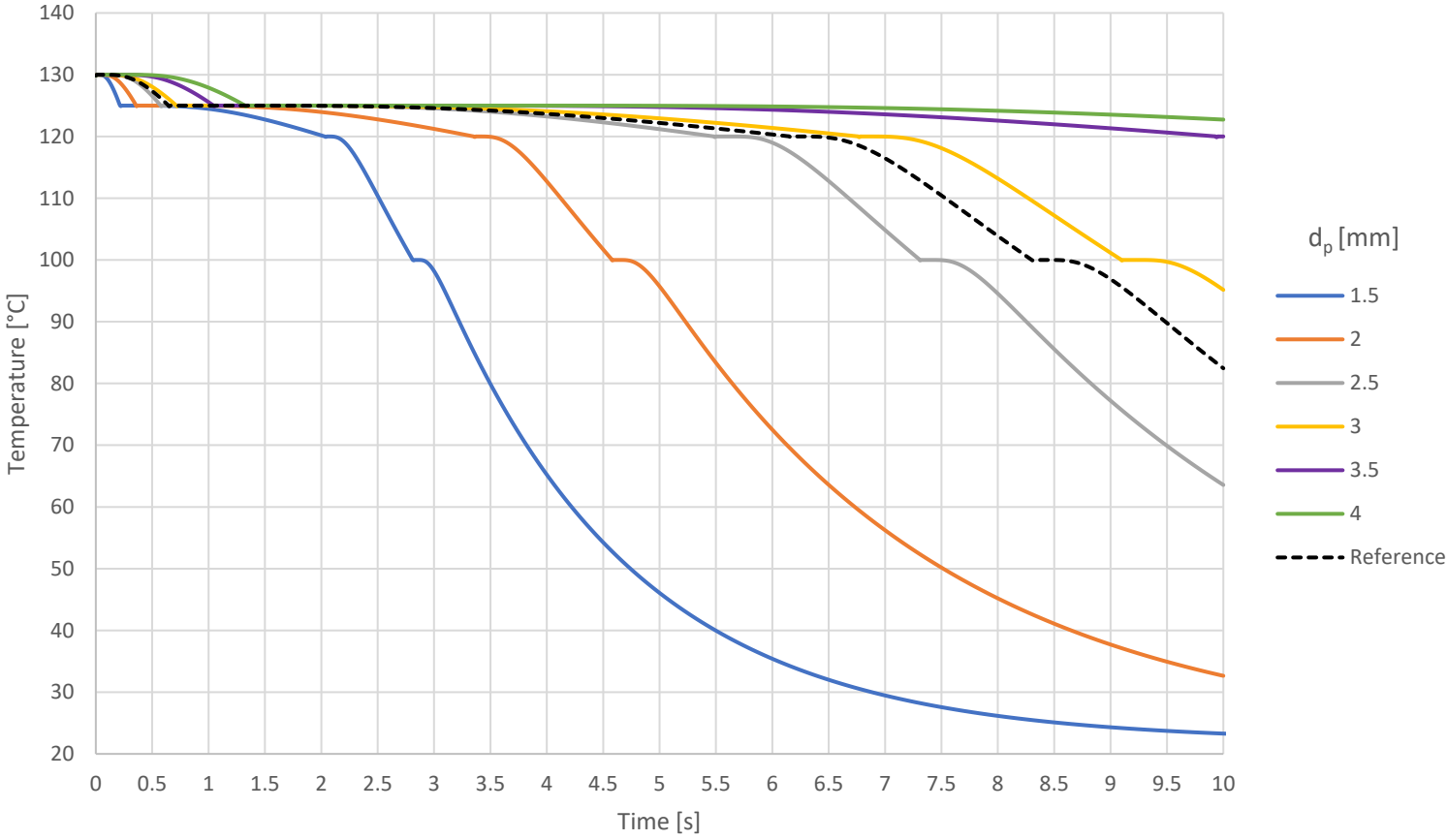


## **B.2 Individual run data**

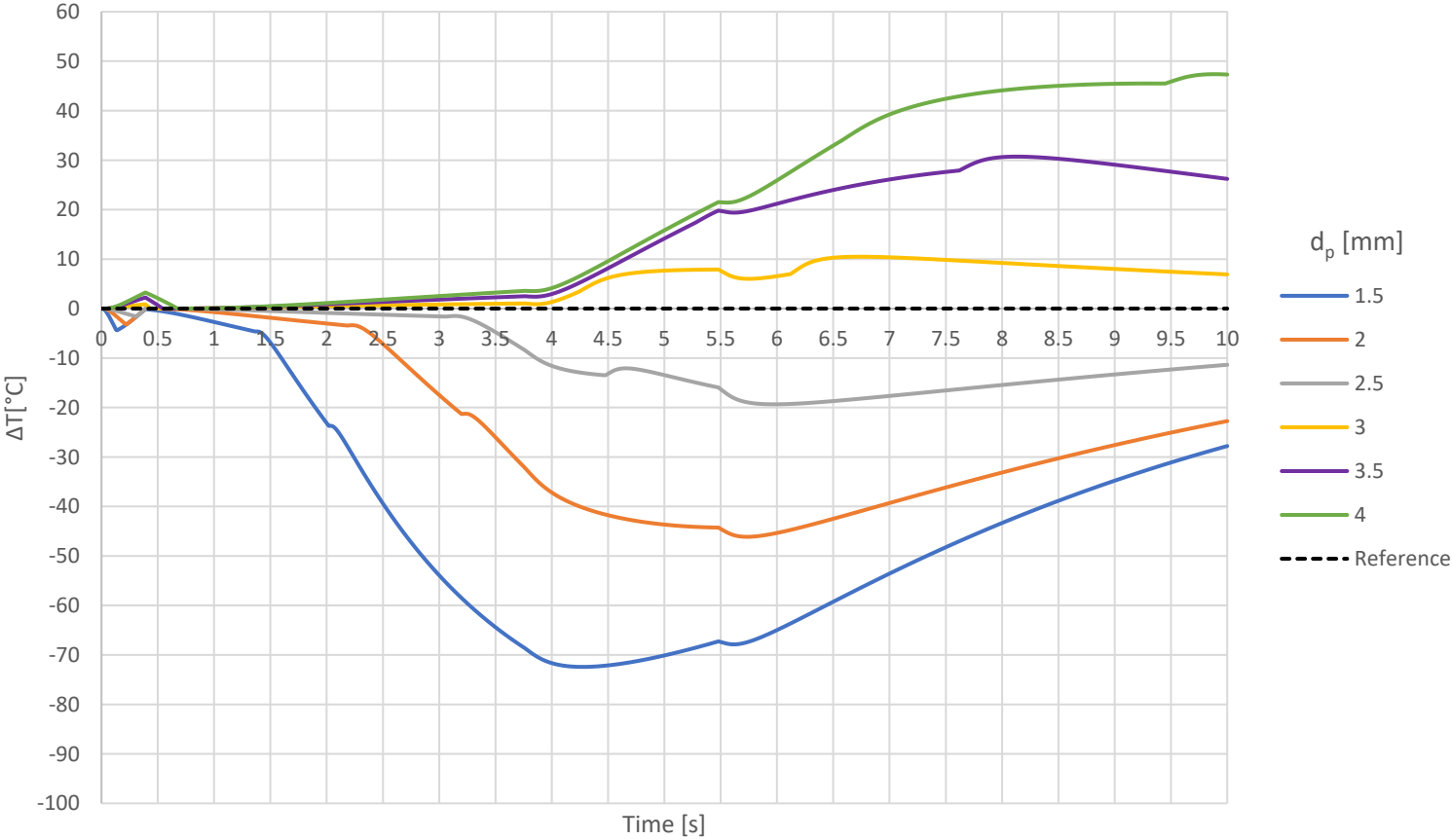
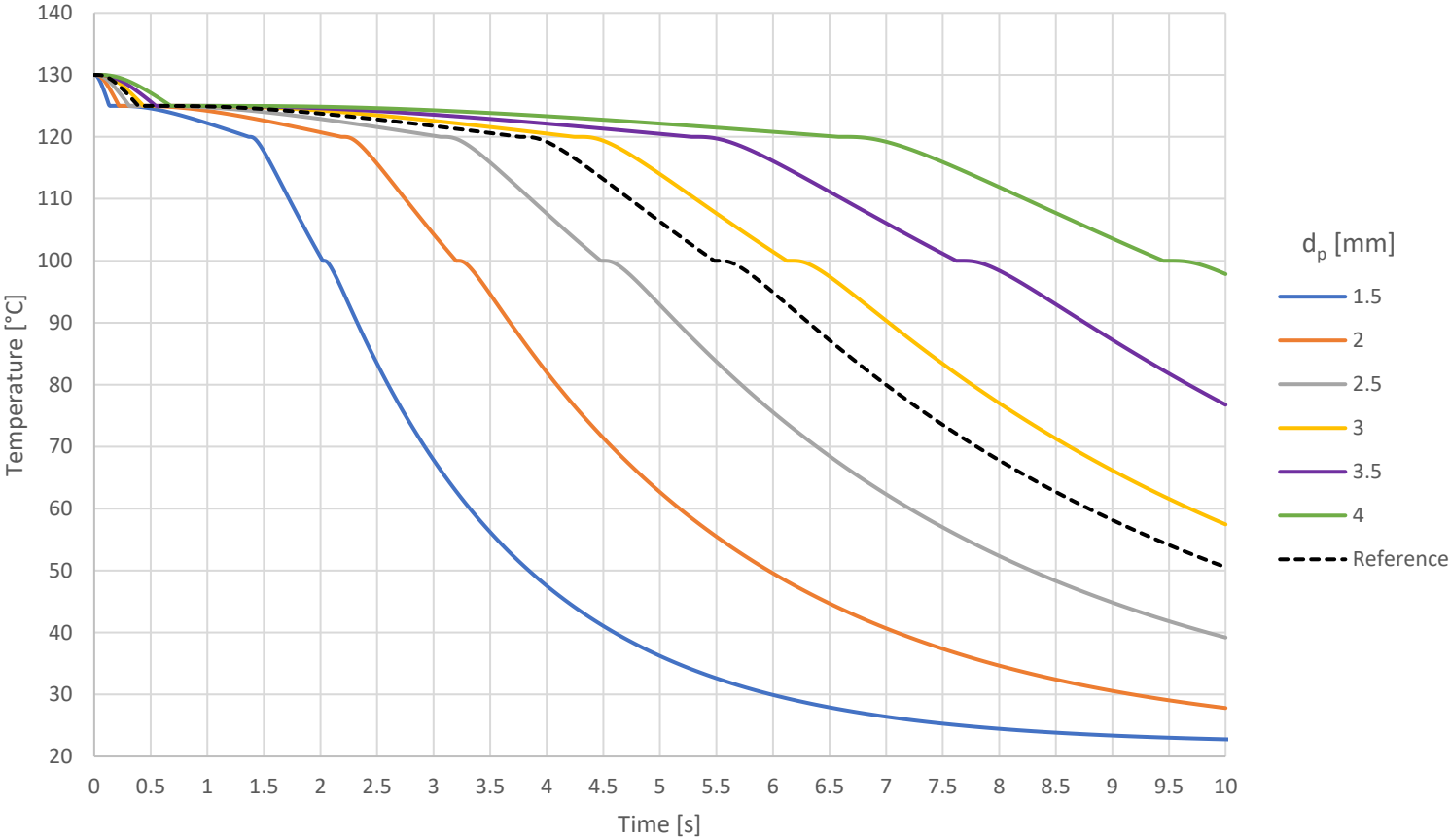
# Particle diameter

$d_{p,ref} = 2.85 \text{ mm}$

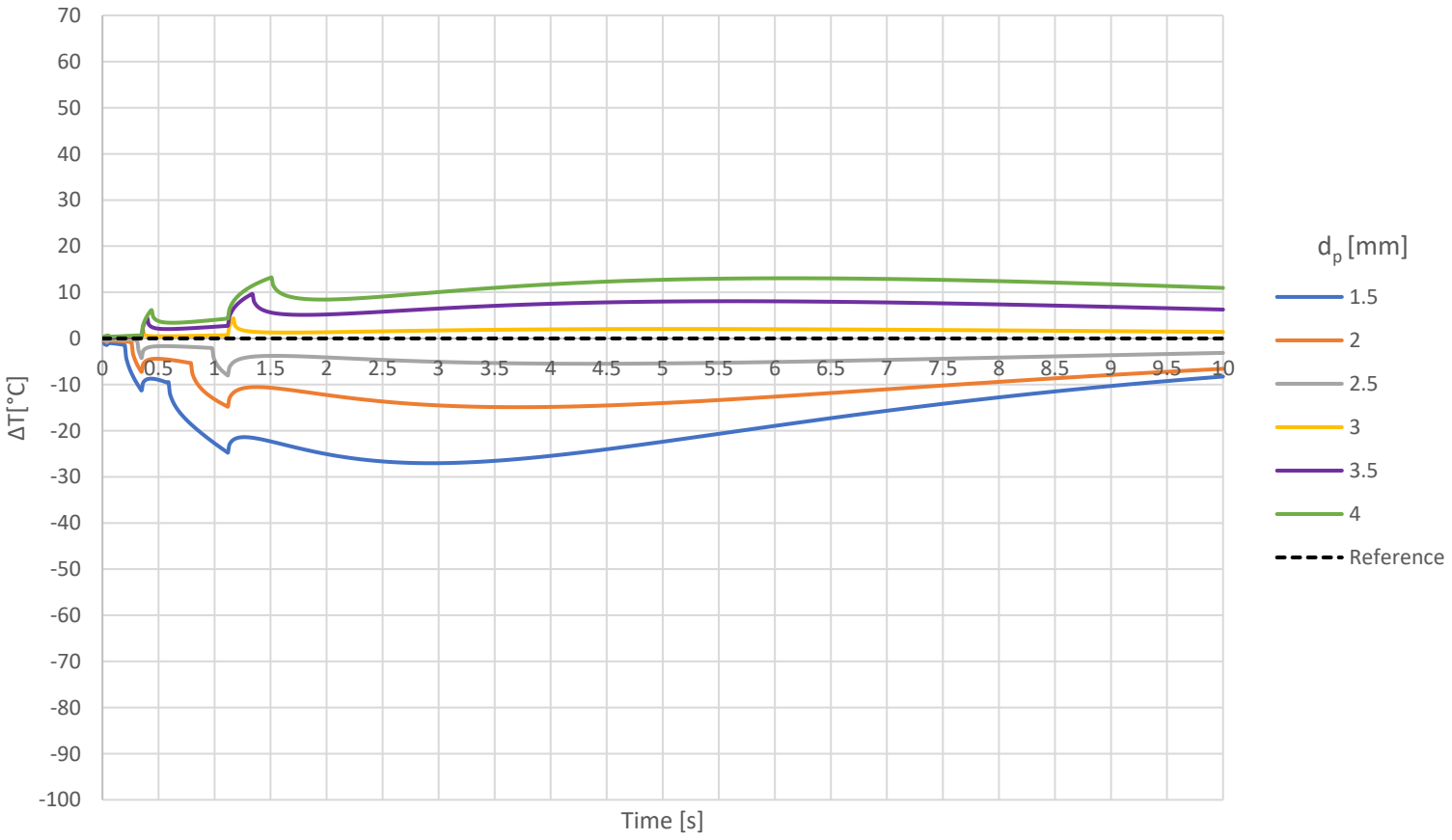
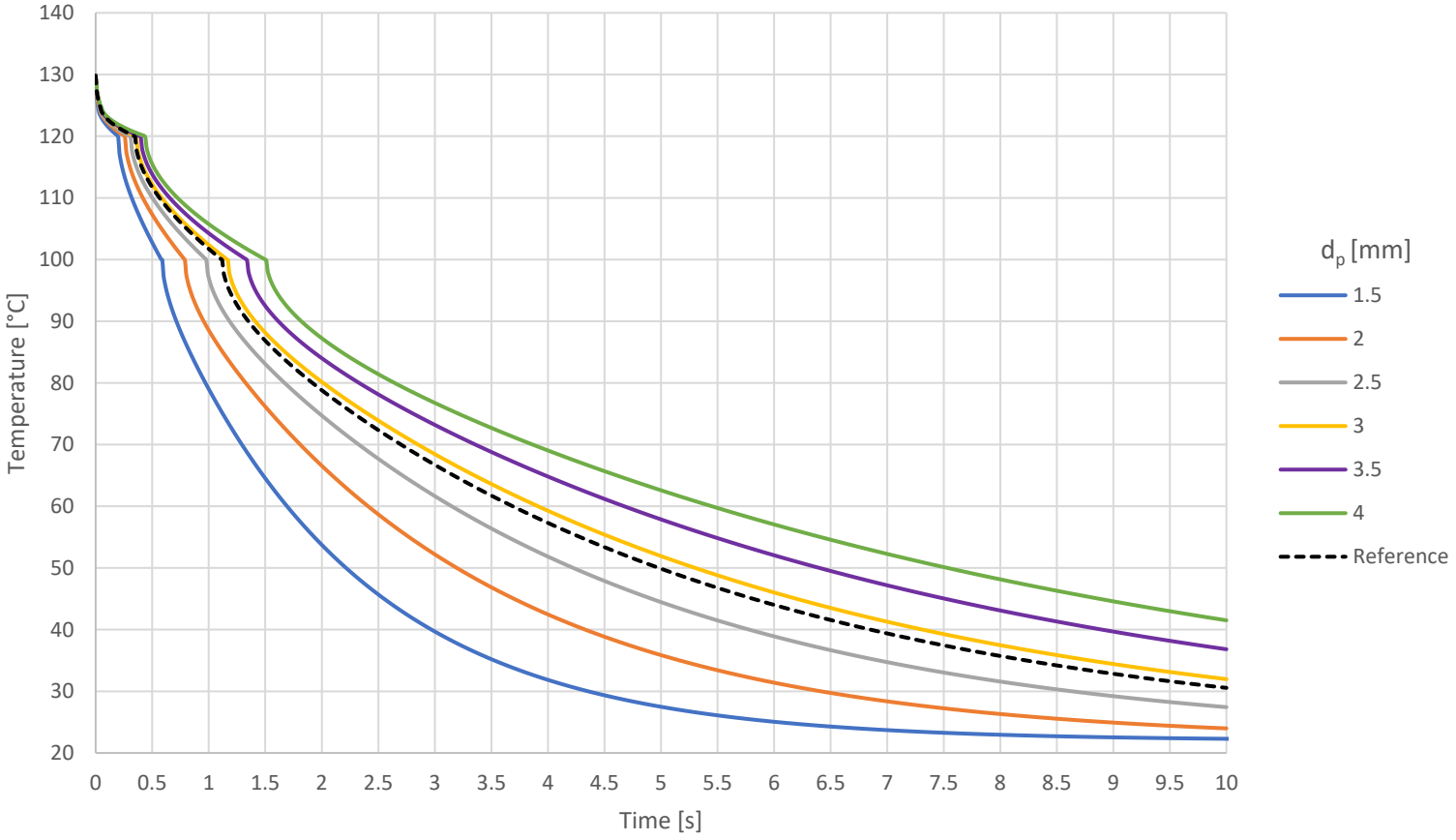
Core temperature developments (top) and temperature differences (bottom):



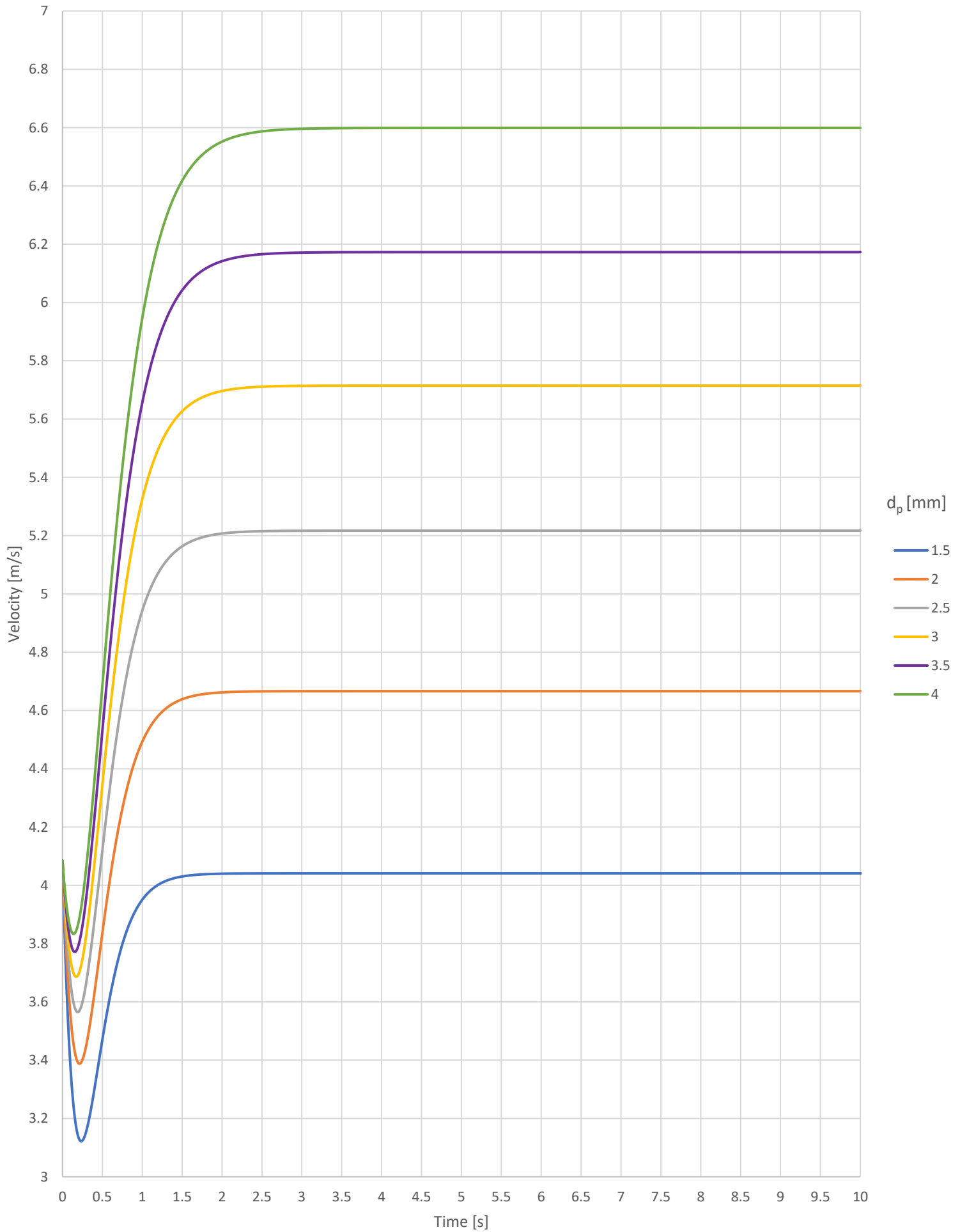
Critical temperature developments (top) and temperature differences (bottom):



Surface temperature developments (top) and temperature differences (bottom):



Total velocities:

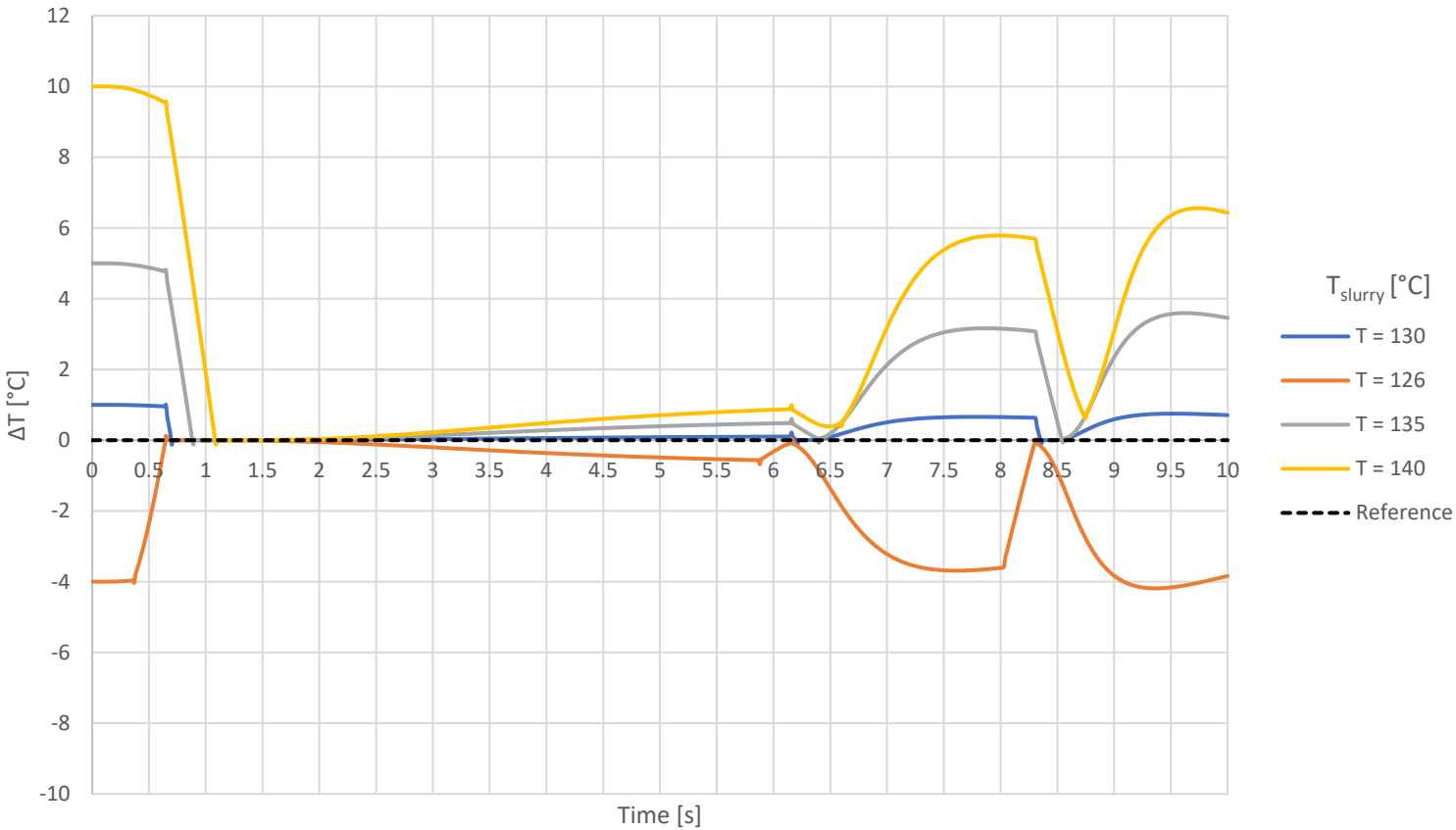
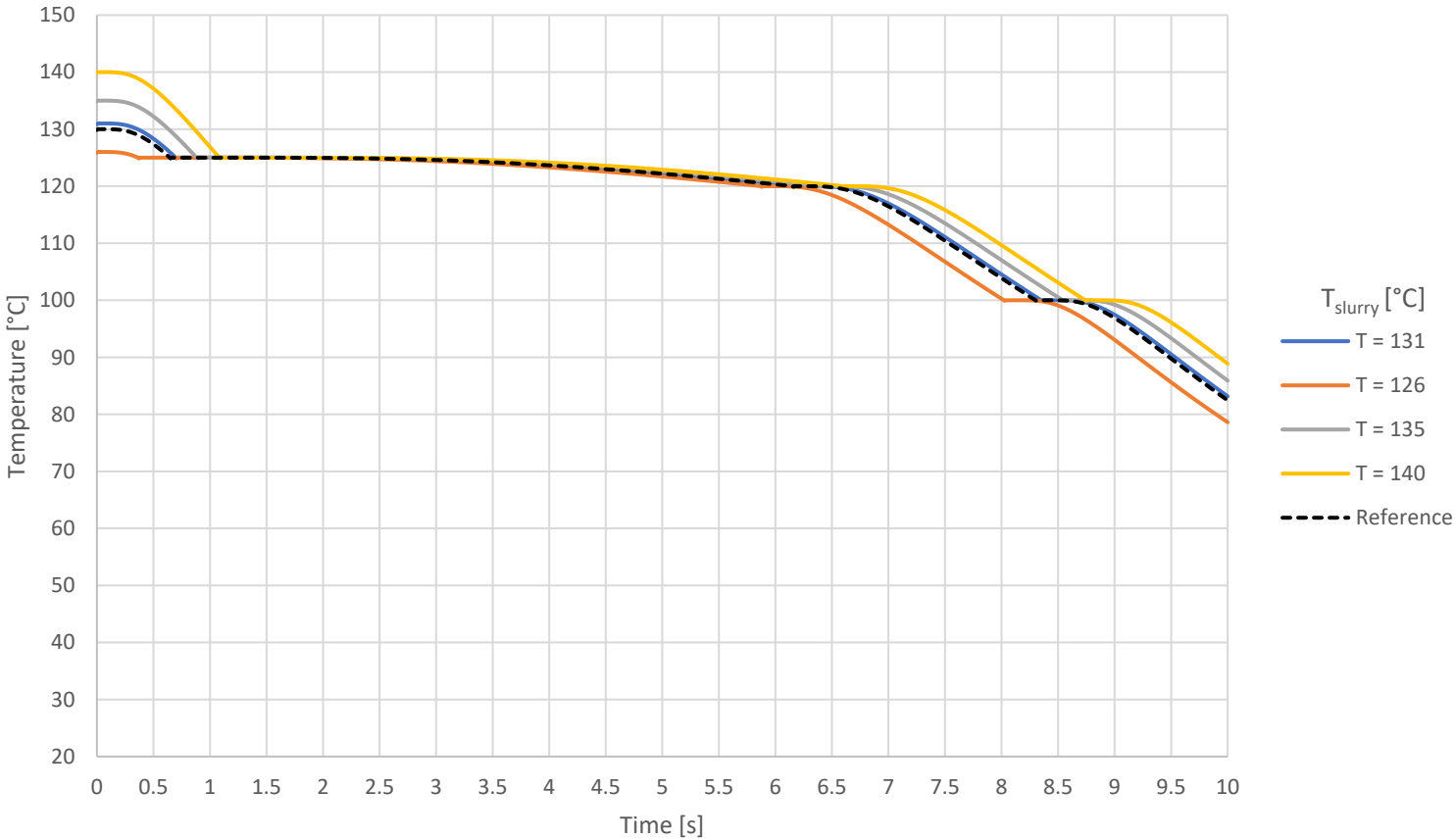




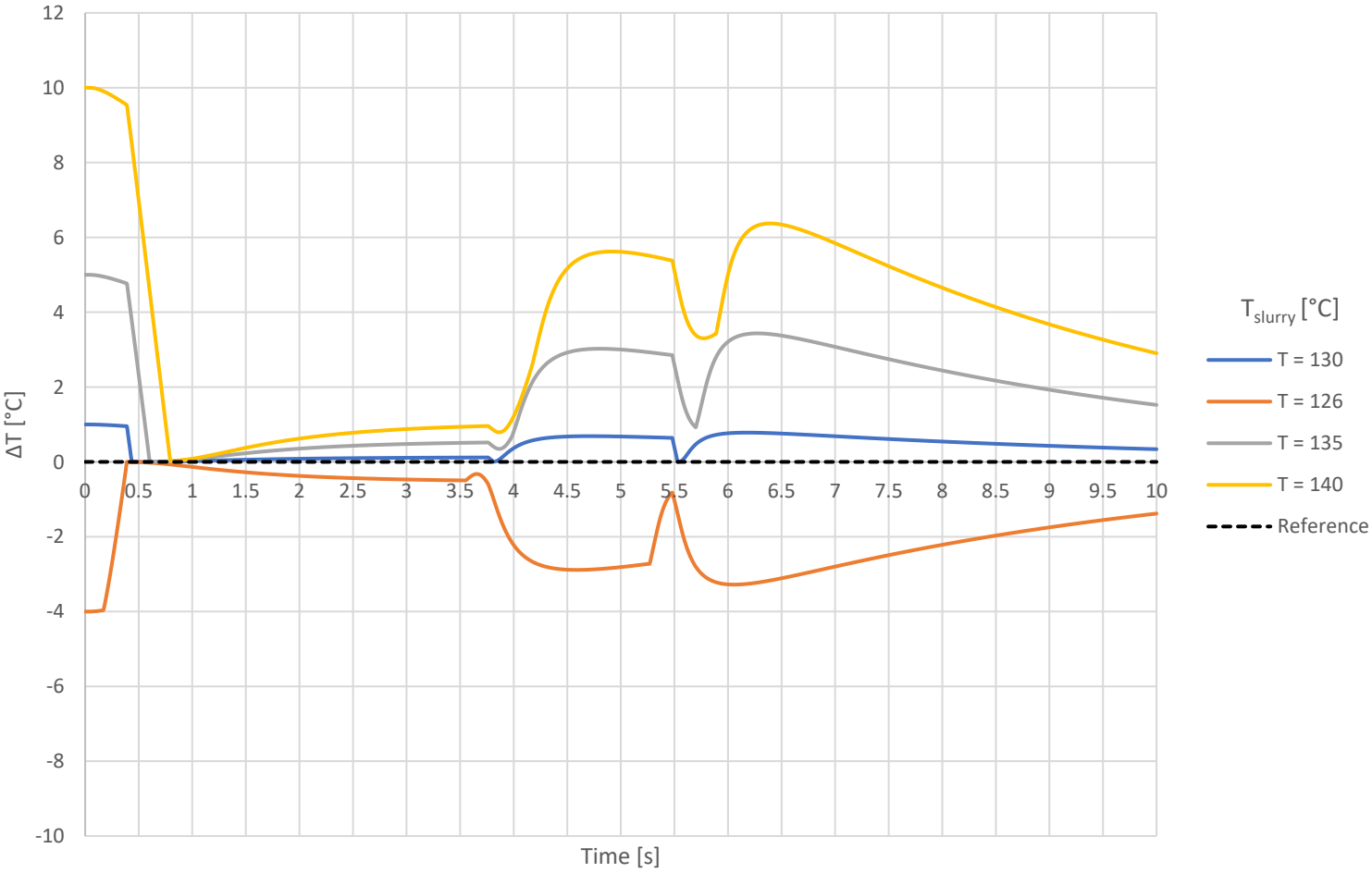
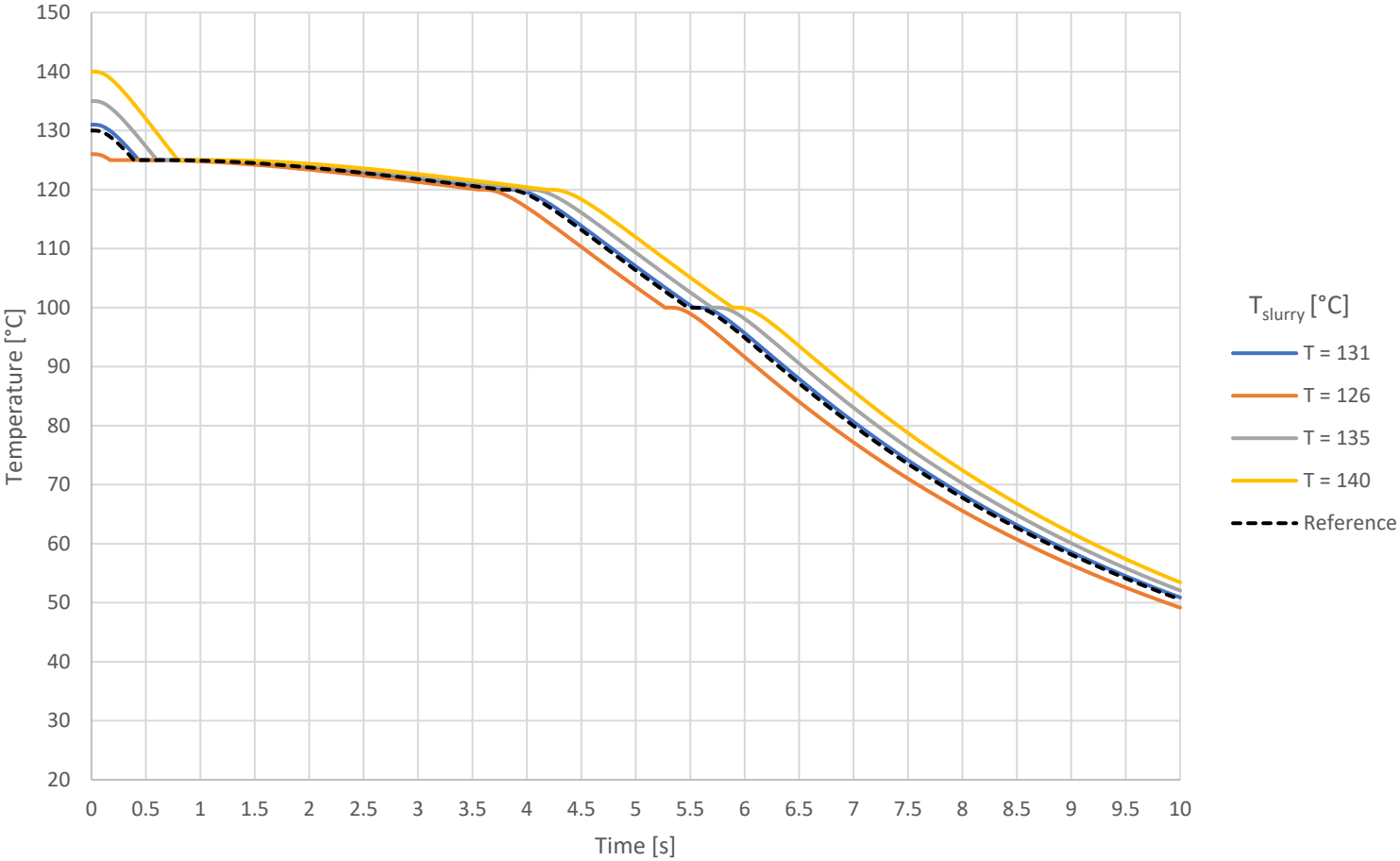
# Slurry temperature

$T_{\text{slurry, ref}} = 130^{\circ}\text{C}$

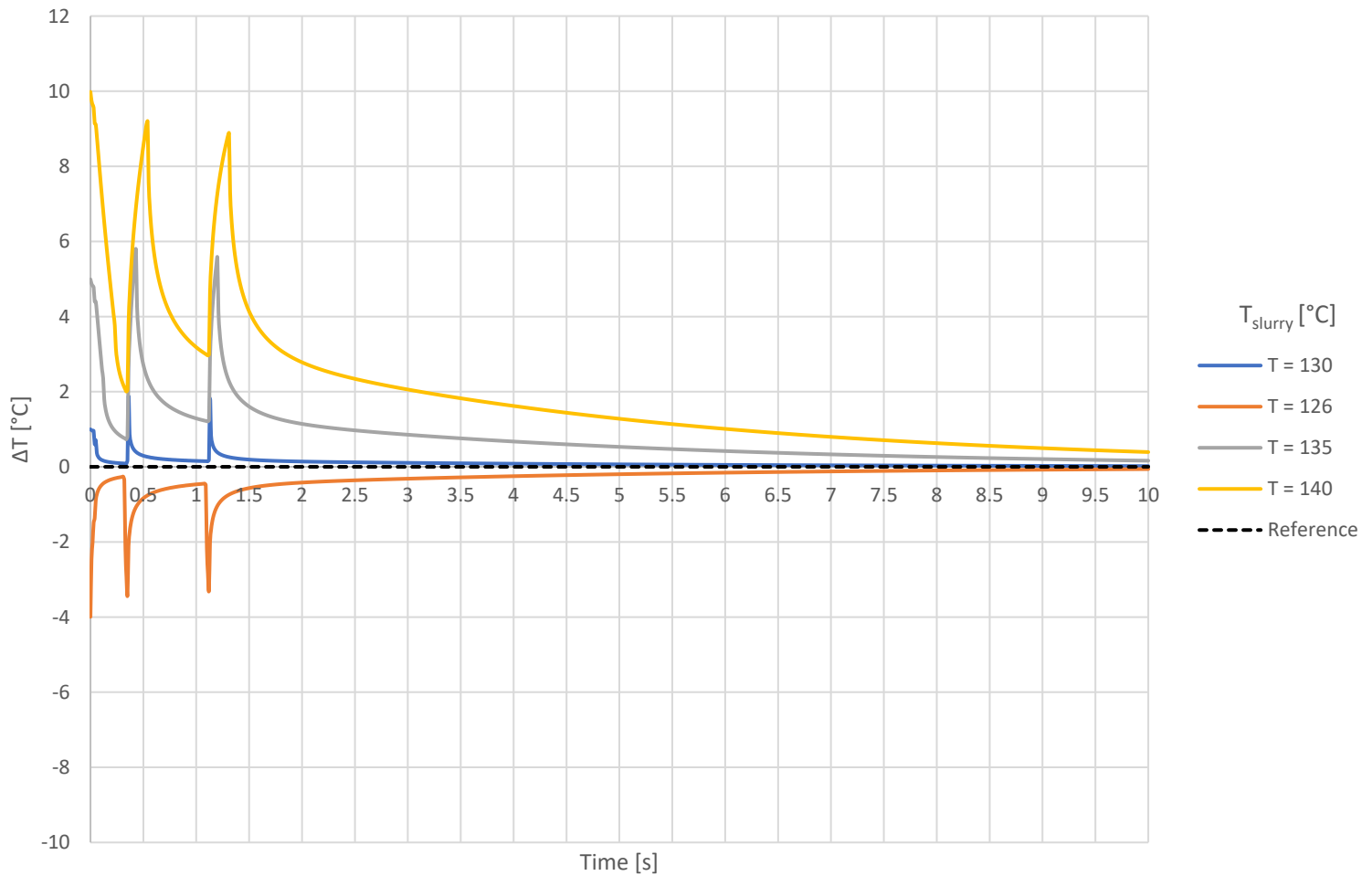
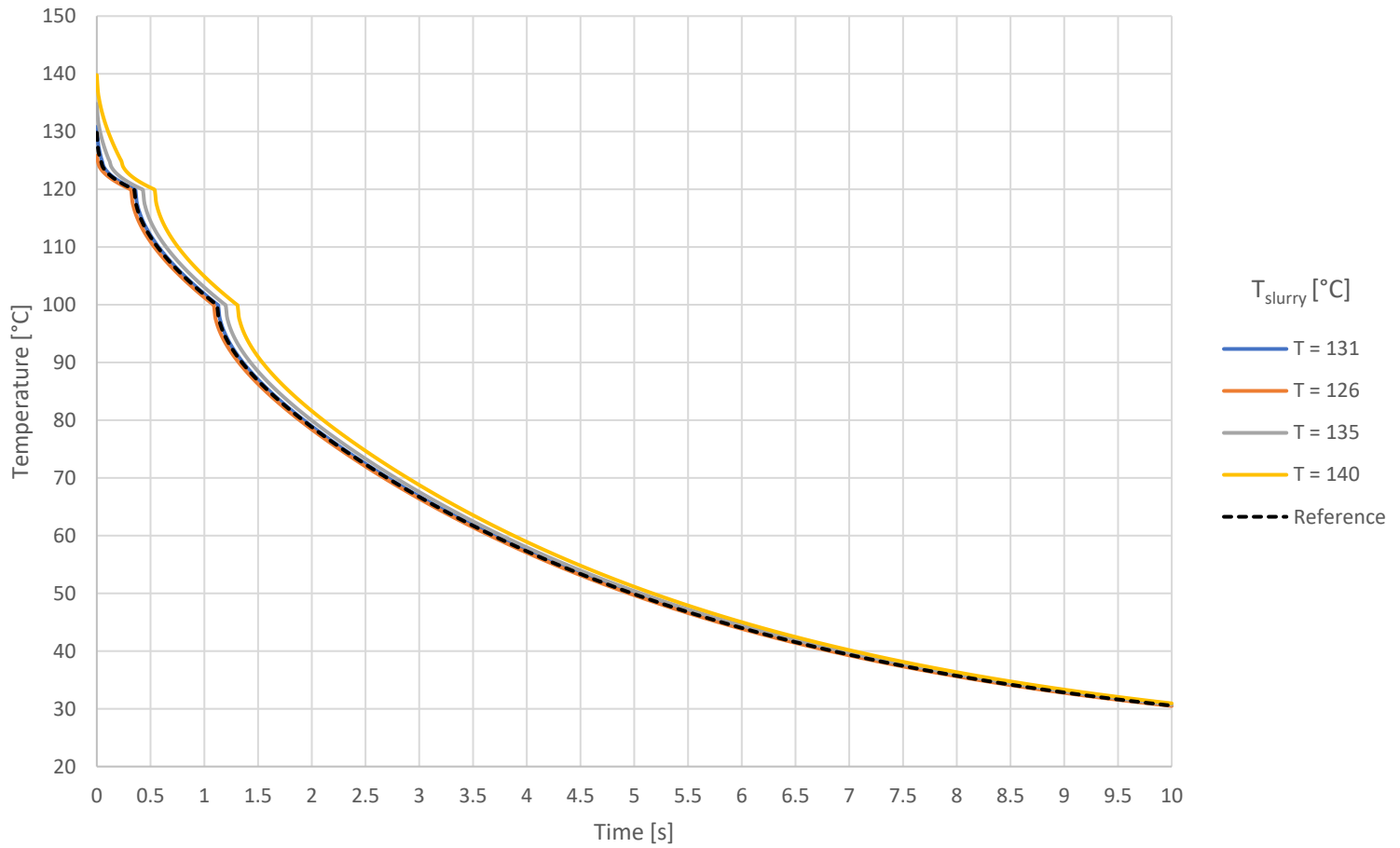
Core temperature developments (top) and temperature differences (bottom):



Critical temperature developments (top) and temperature differences (bottom):



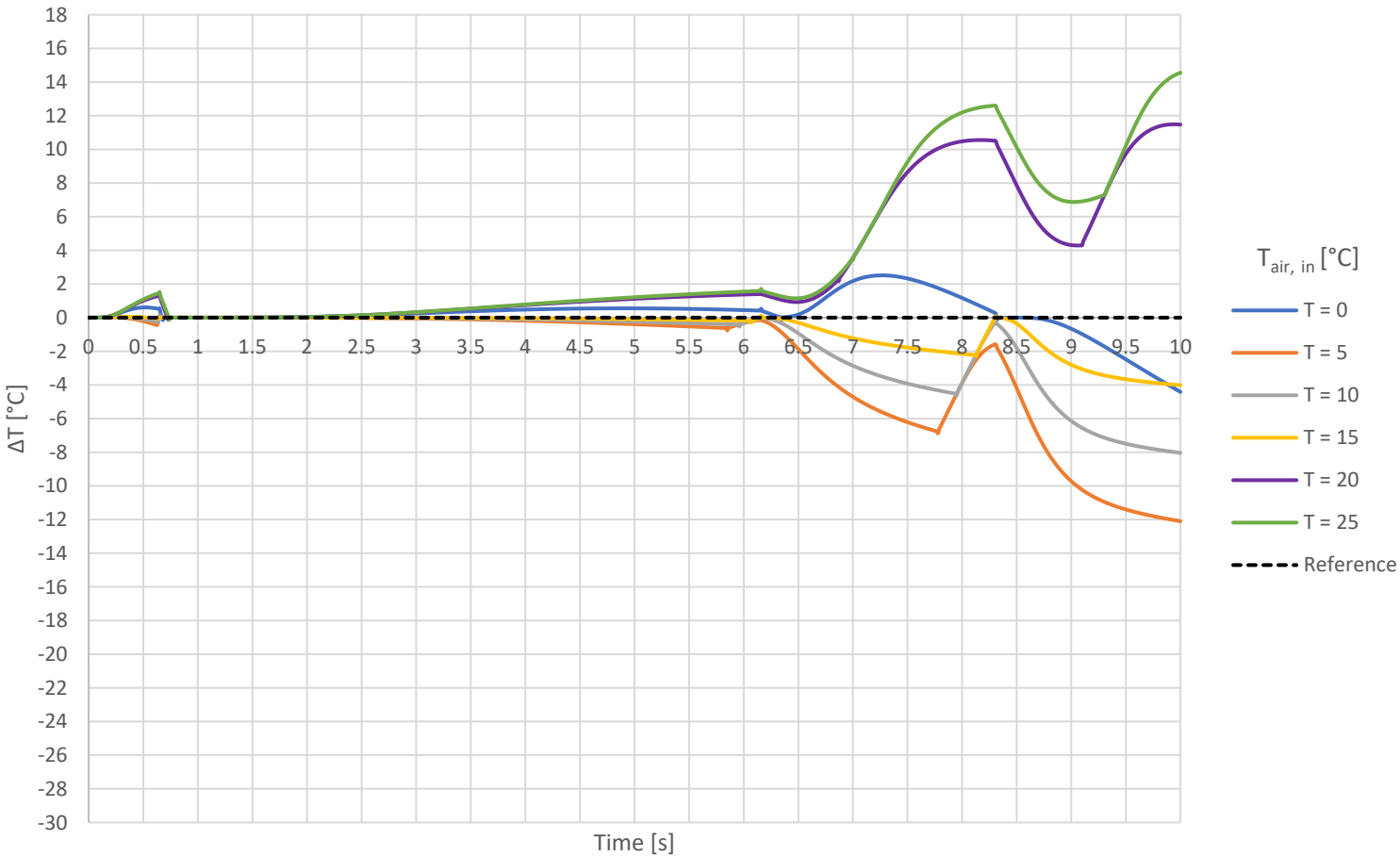
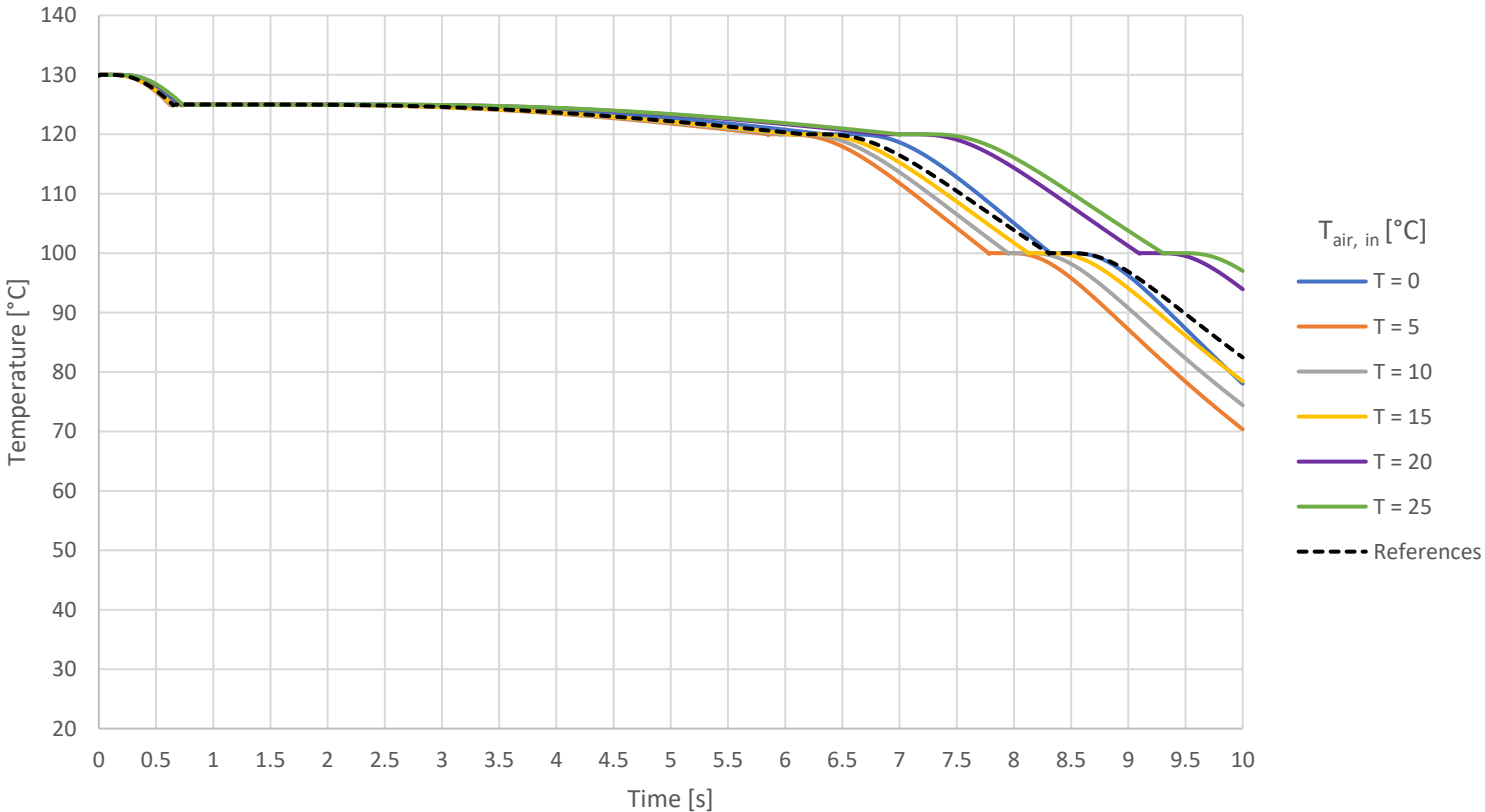
Surface temperature developments (top) and temperature differences (bottom):



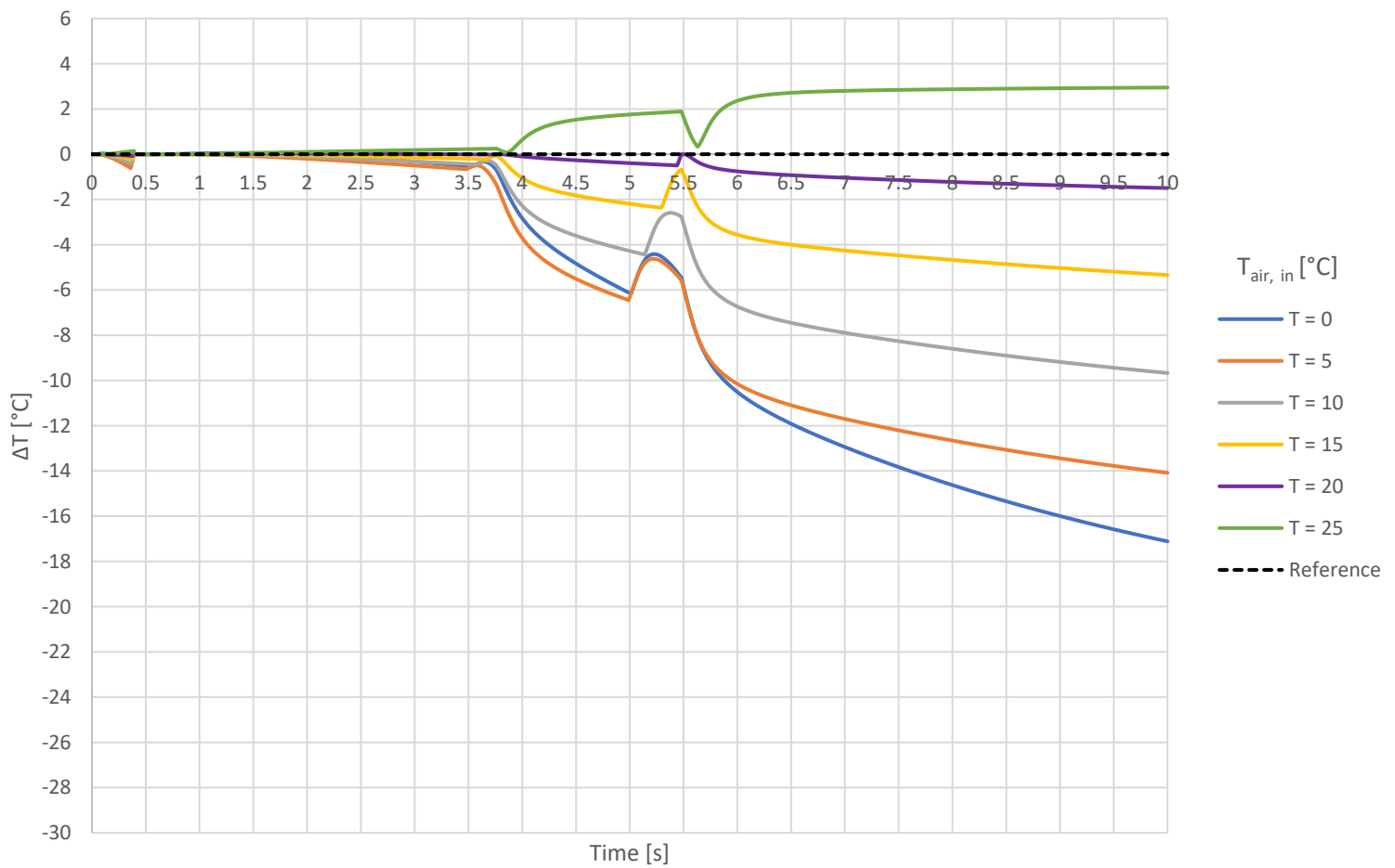
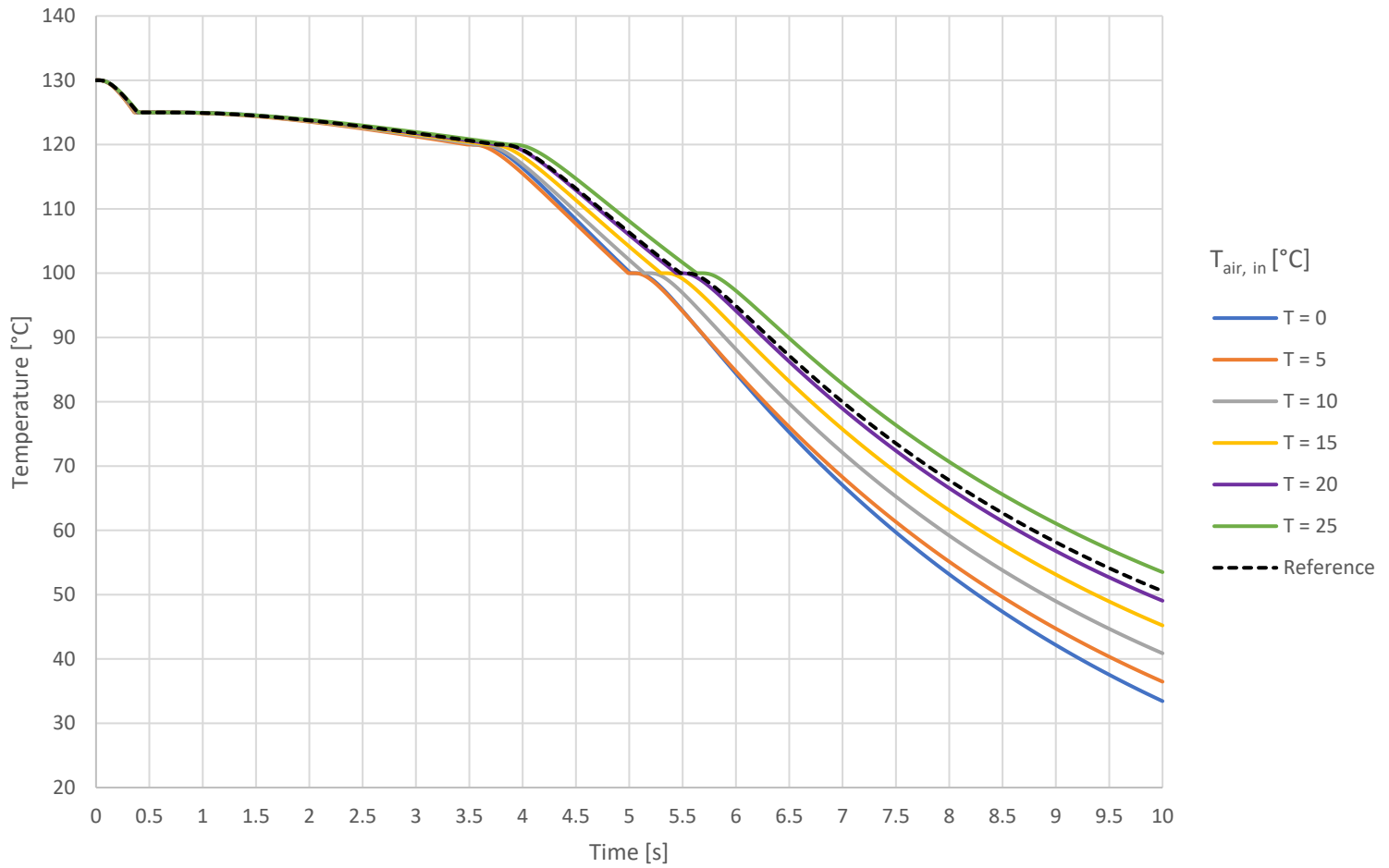
# Ambient air temperature

$T_{air, in, ref} = 22^{\circ}C$

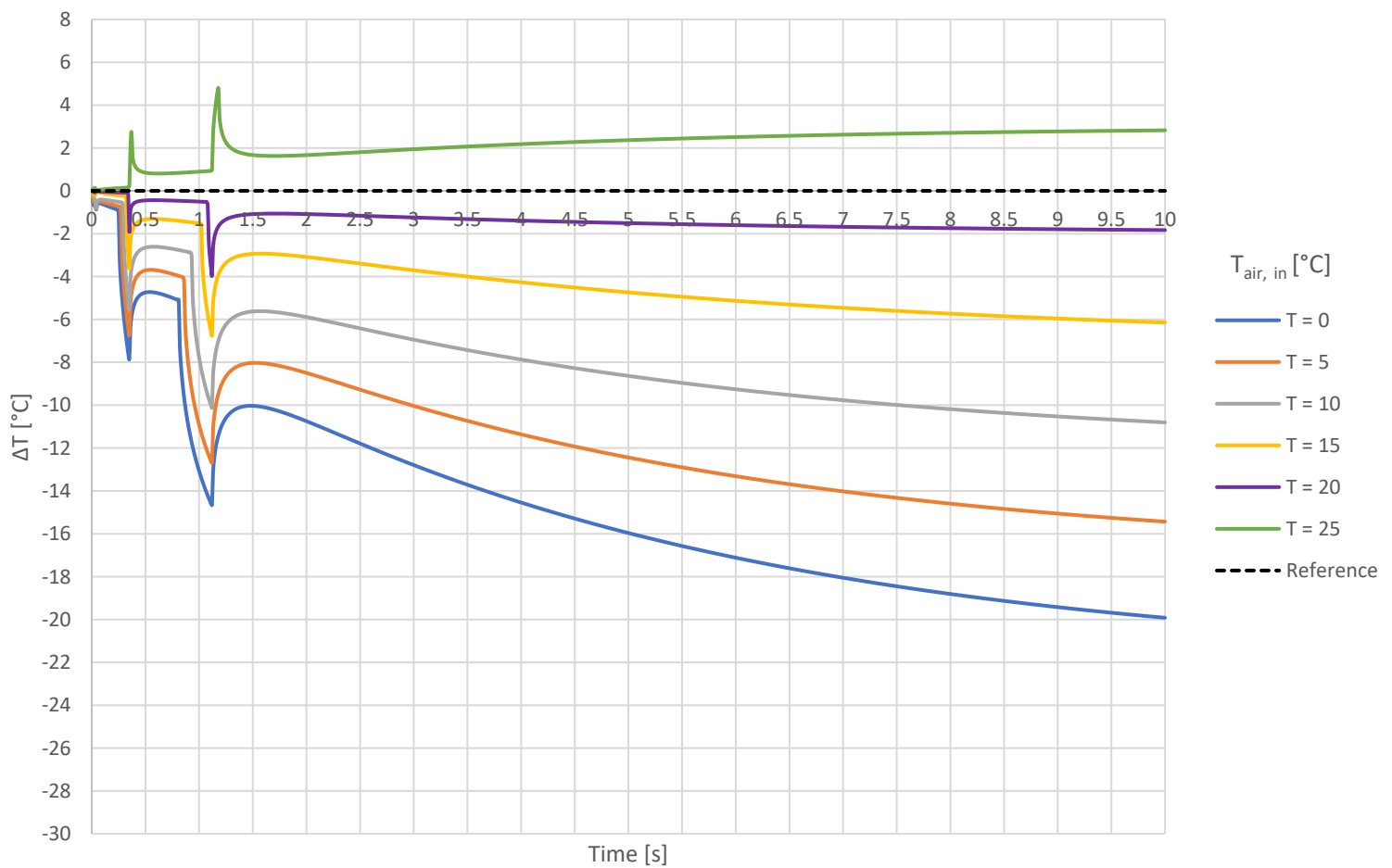
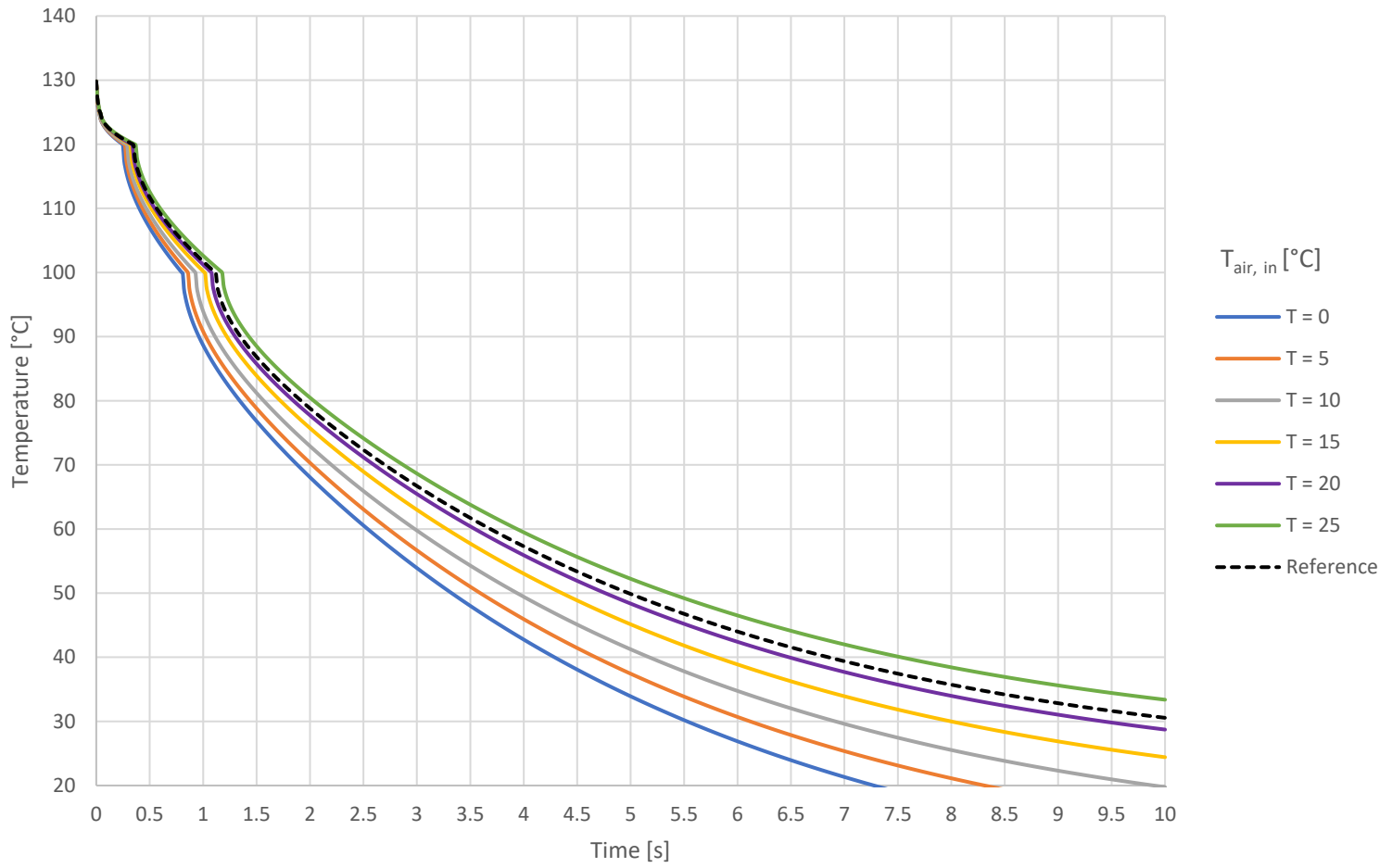
Core temperature developments (top) and temperature differences (bottom):



Critical temperature developments (top) and temperature differences (bottom):



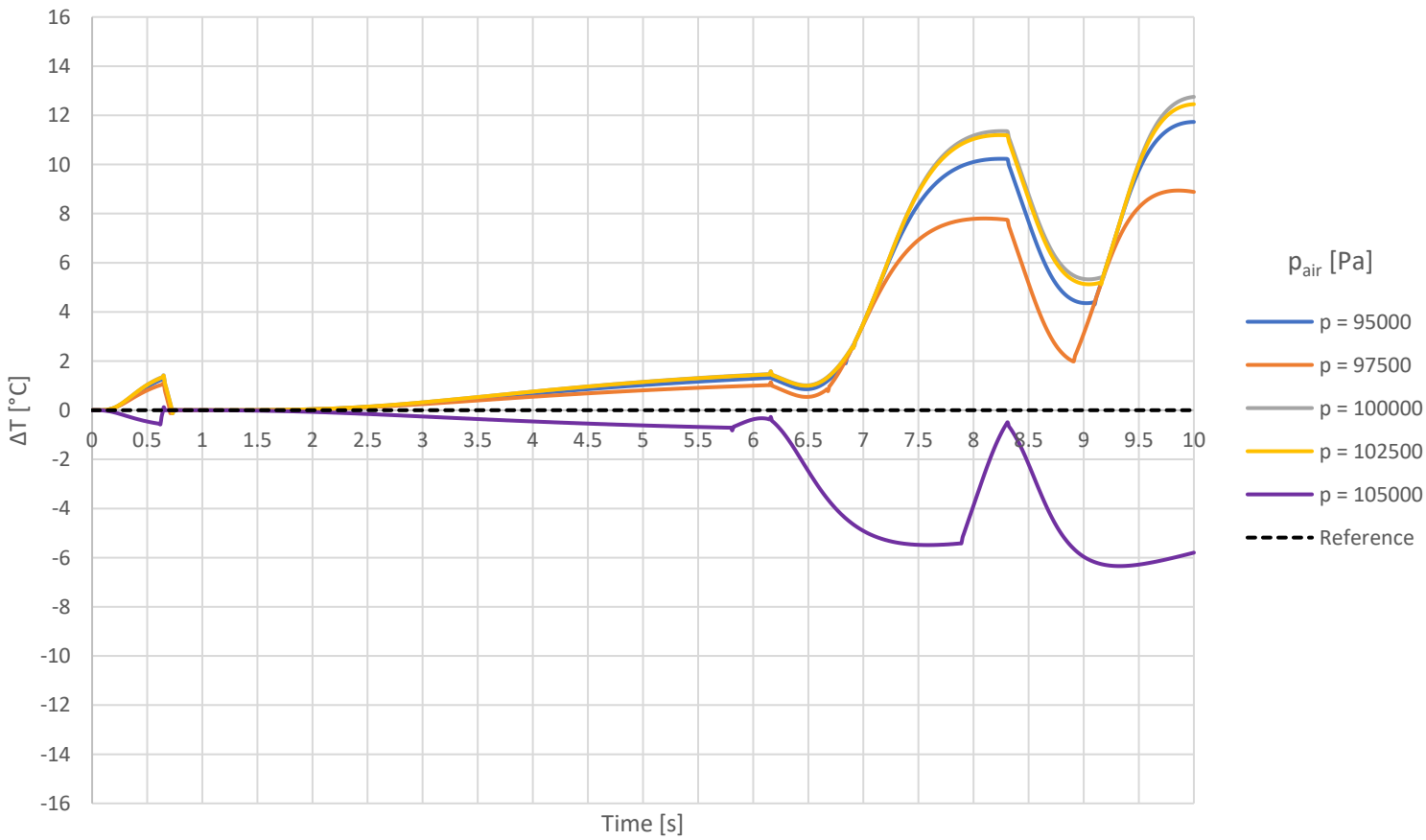
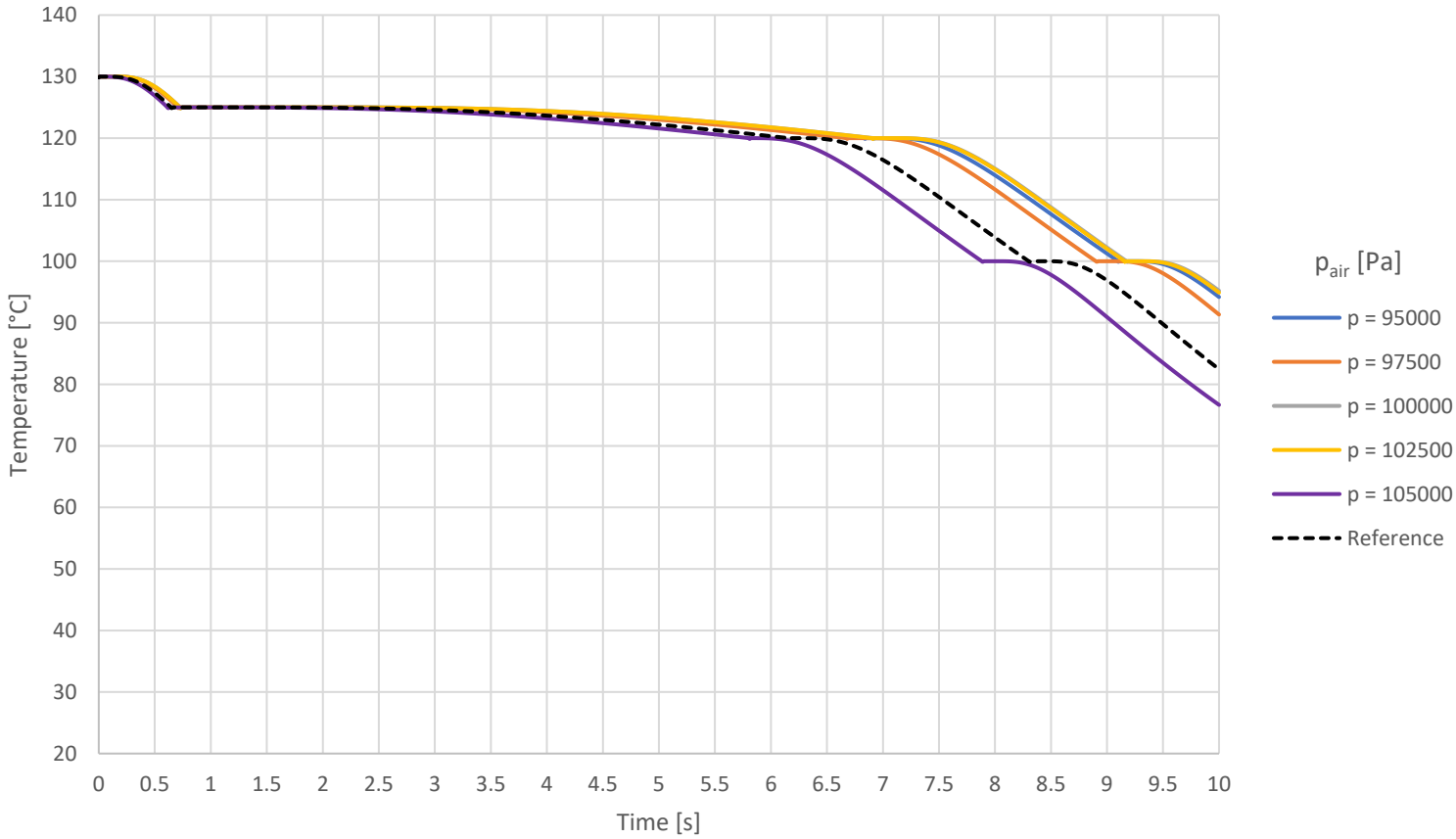
Surface temperature developments (top) and temperature differences (bottom):



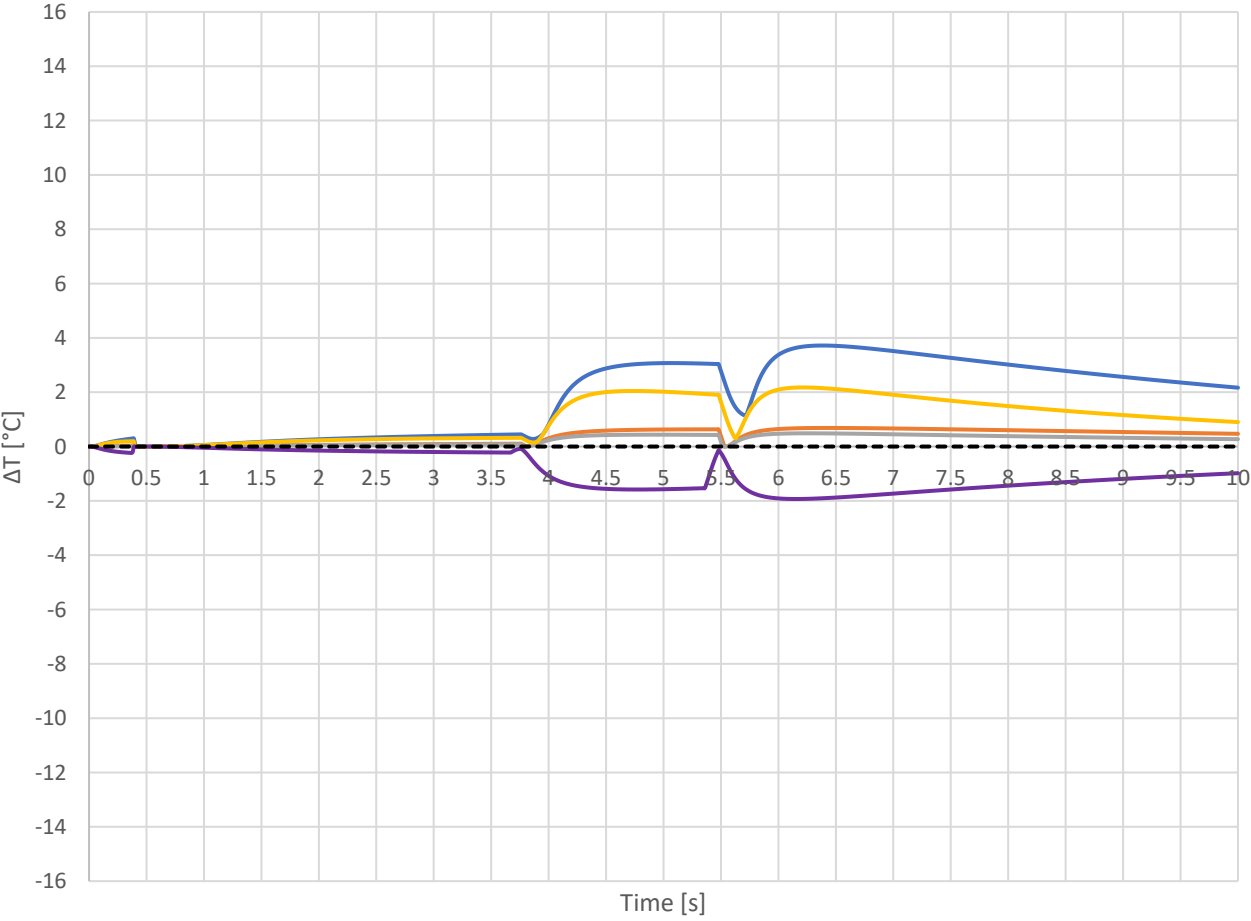
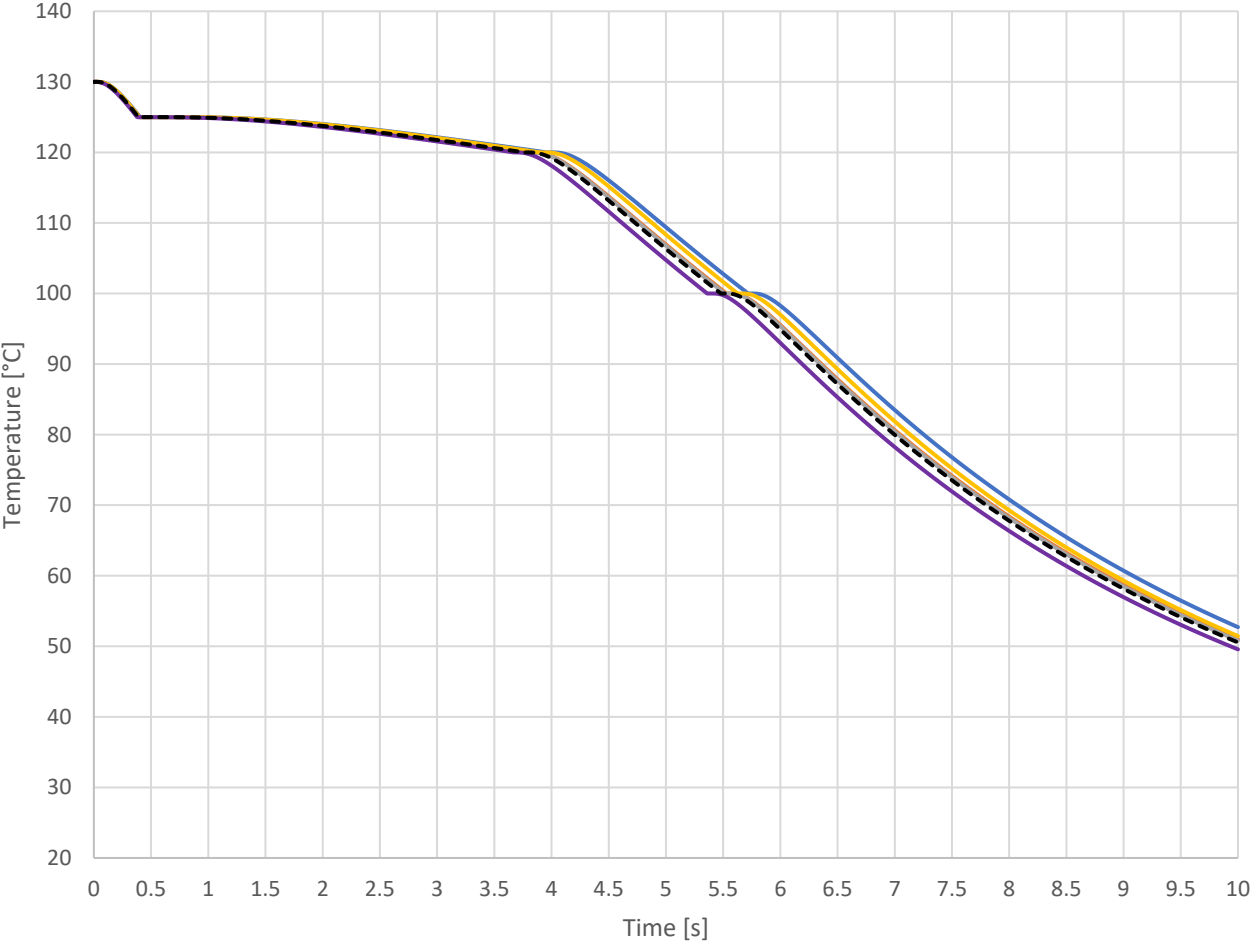
# Ambient air pressure

$p_{air, ref} = 101325 \text{ Pa}$

Core temperature developments (top) and temperature differences (bottom):

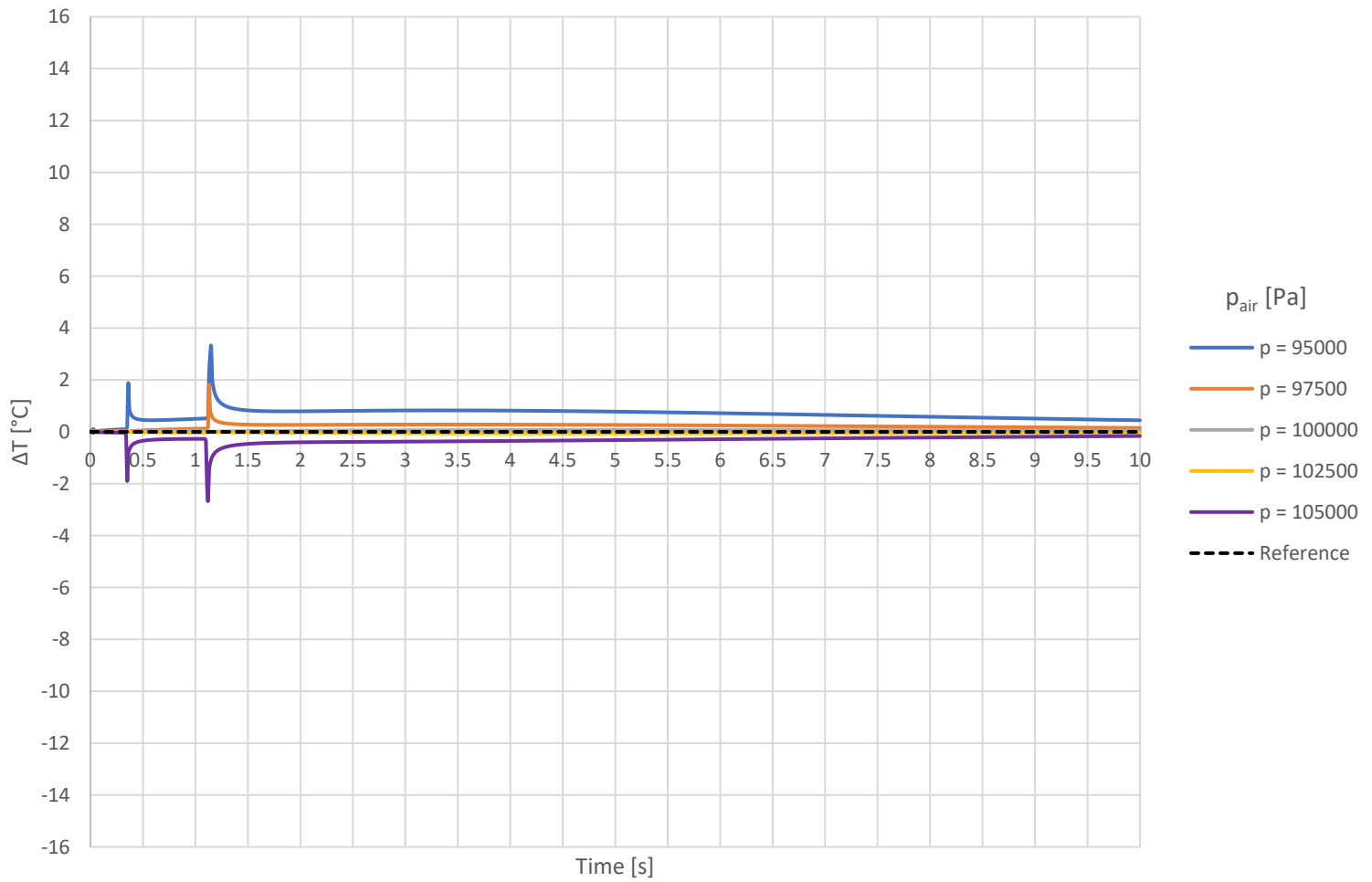
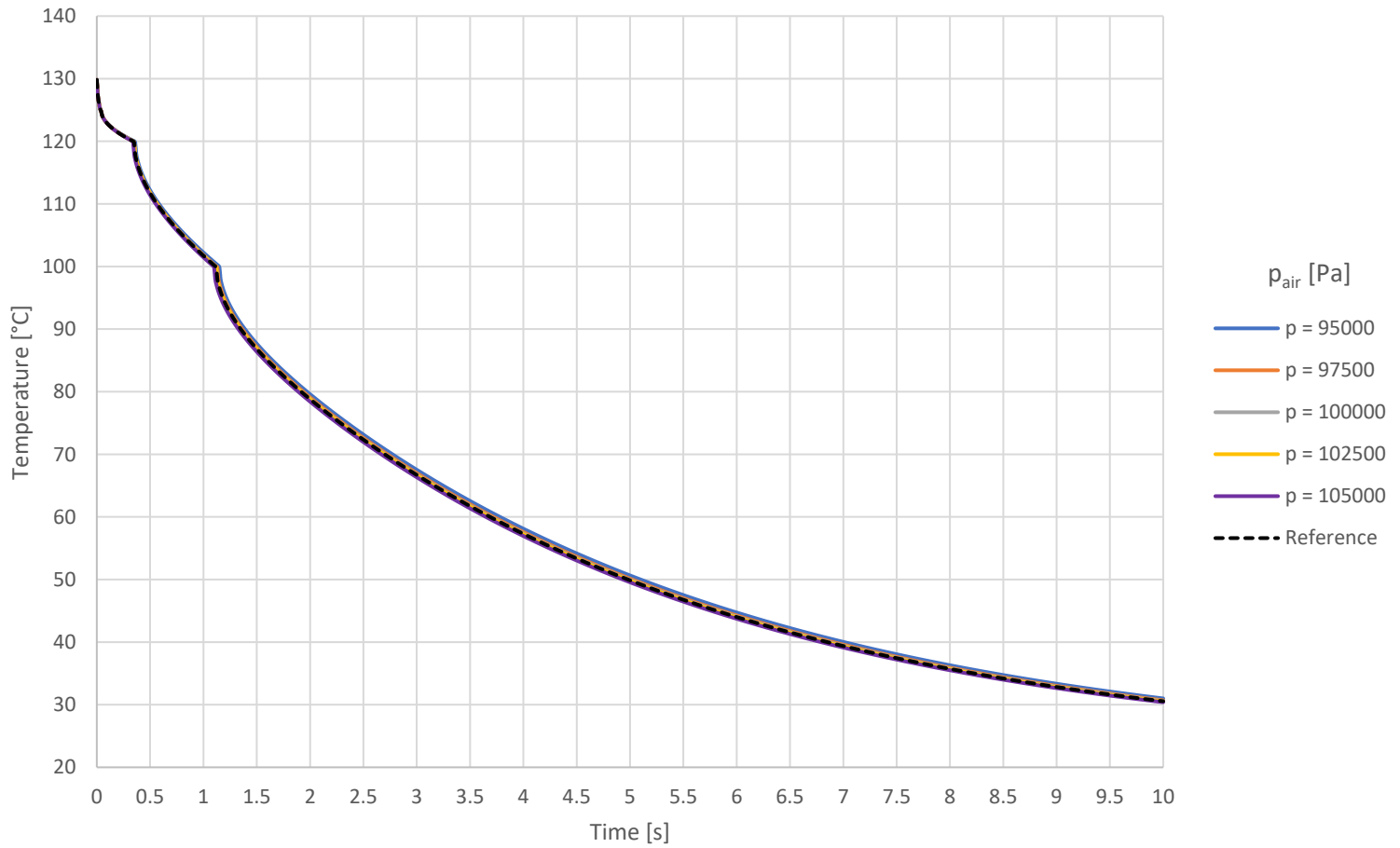


Critical temperature developments (top) and temperature differences (bottom):





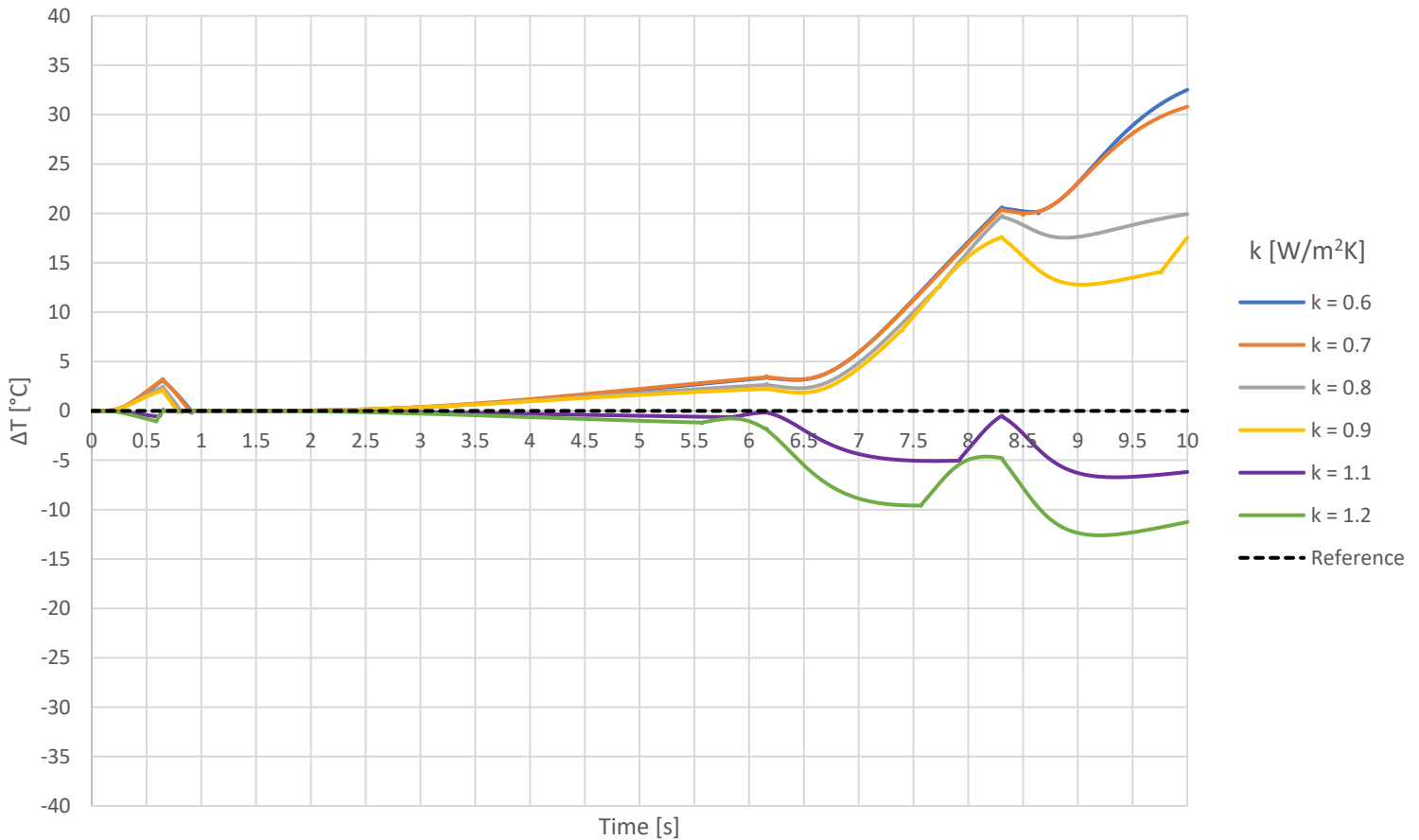
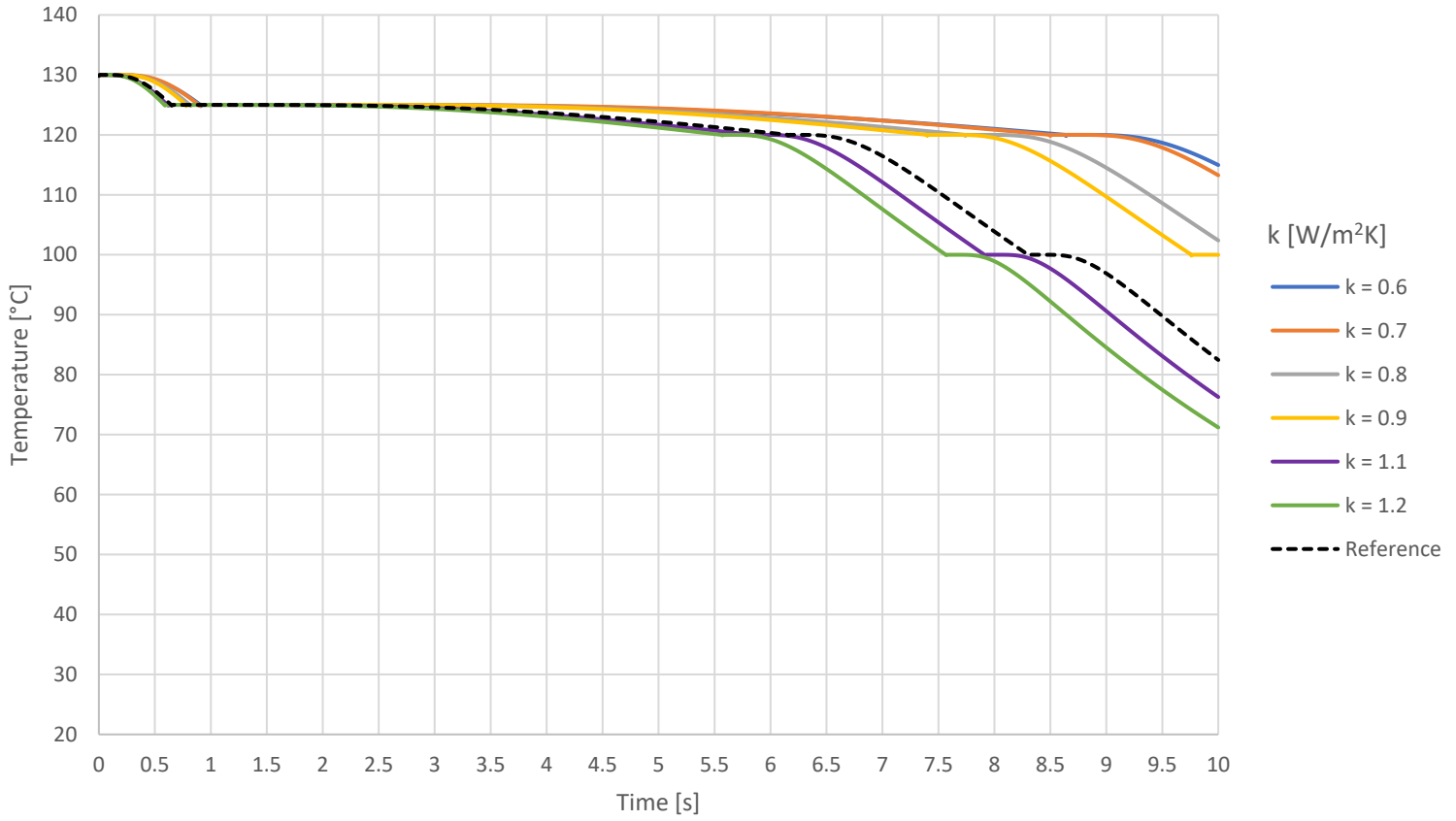
Surface temperature developments (top) and temperature differences (bottom):



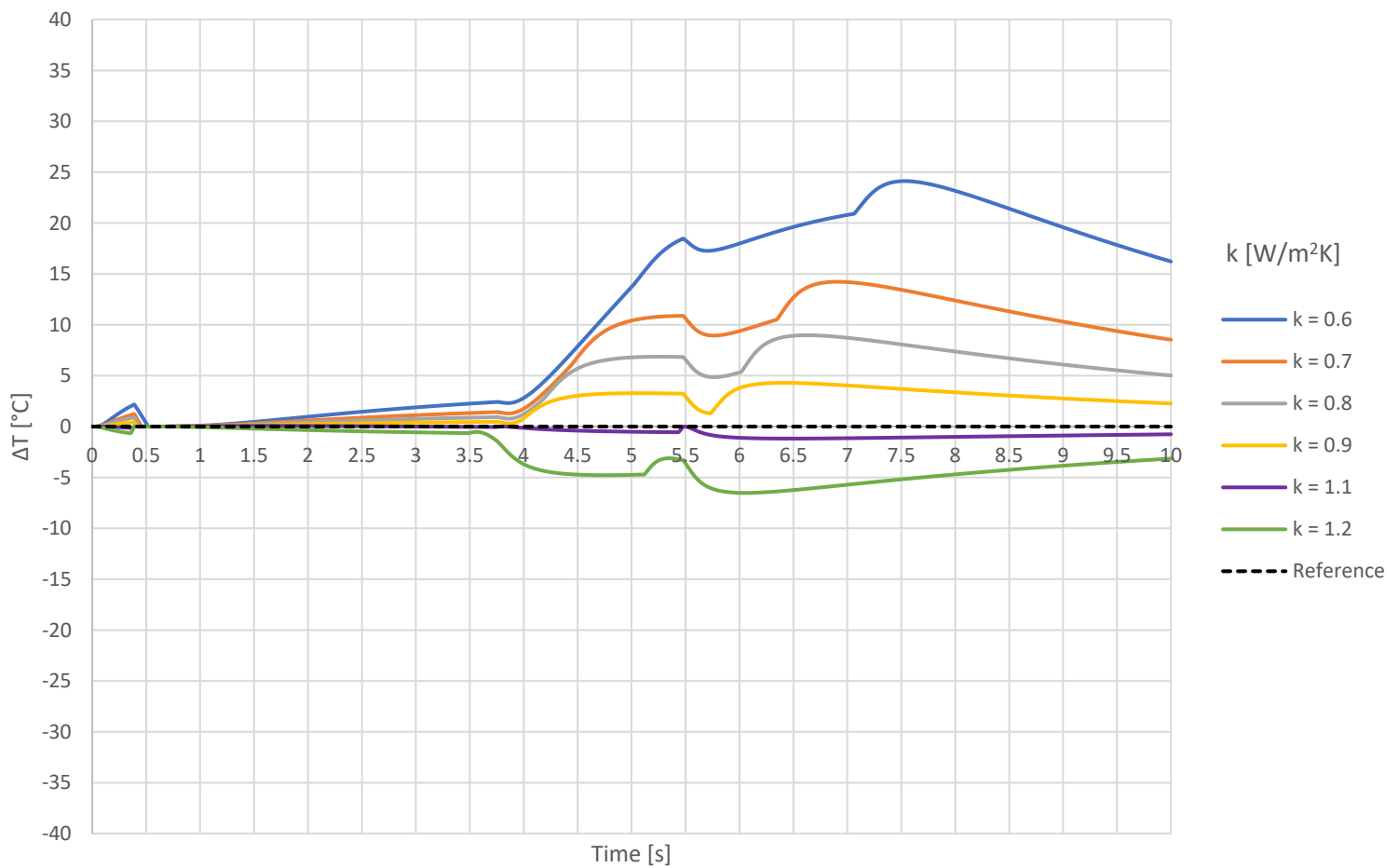
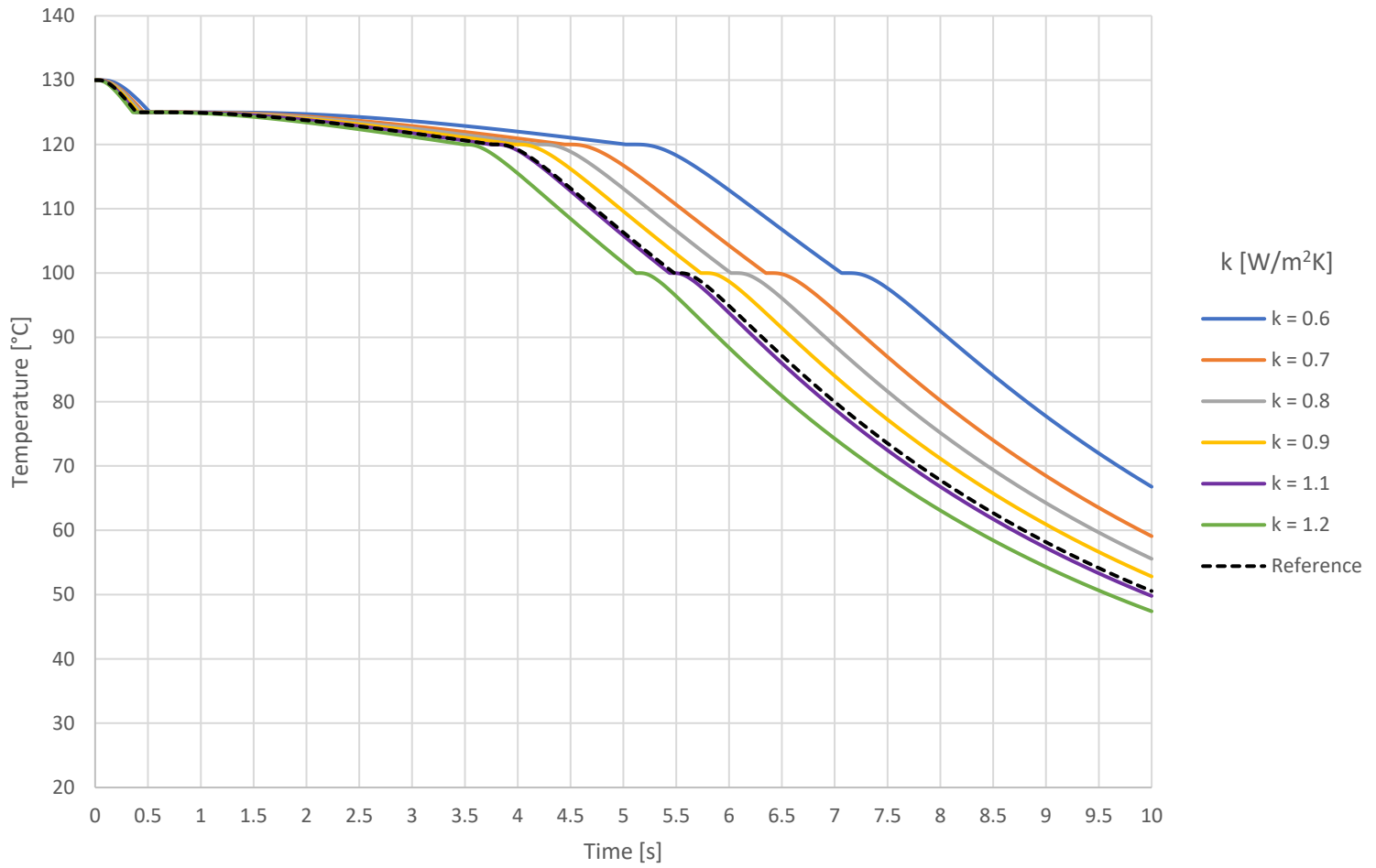
# Thermal conductivity

$K_{ref} = 1.0 \text{ W/m}^2\text{K}$

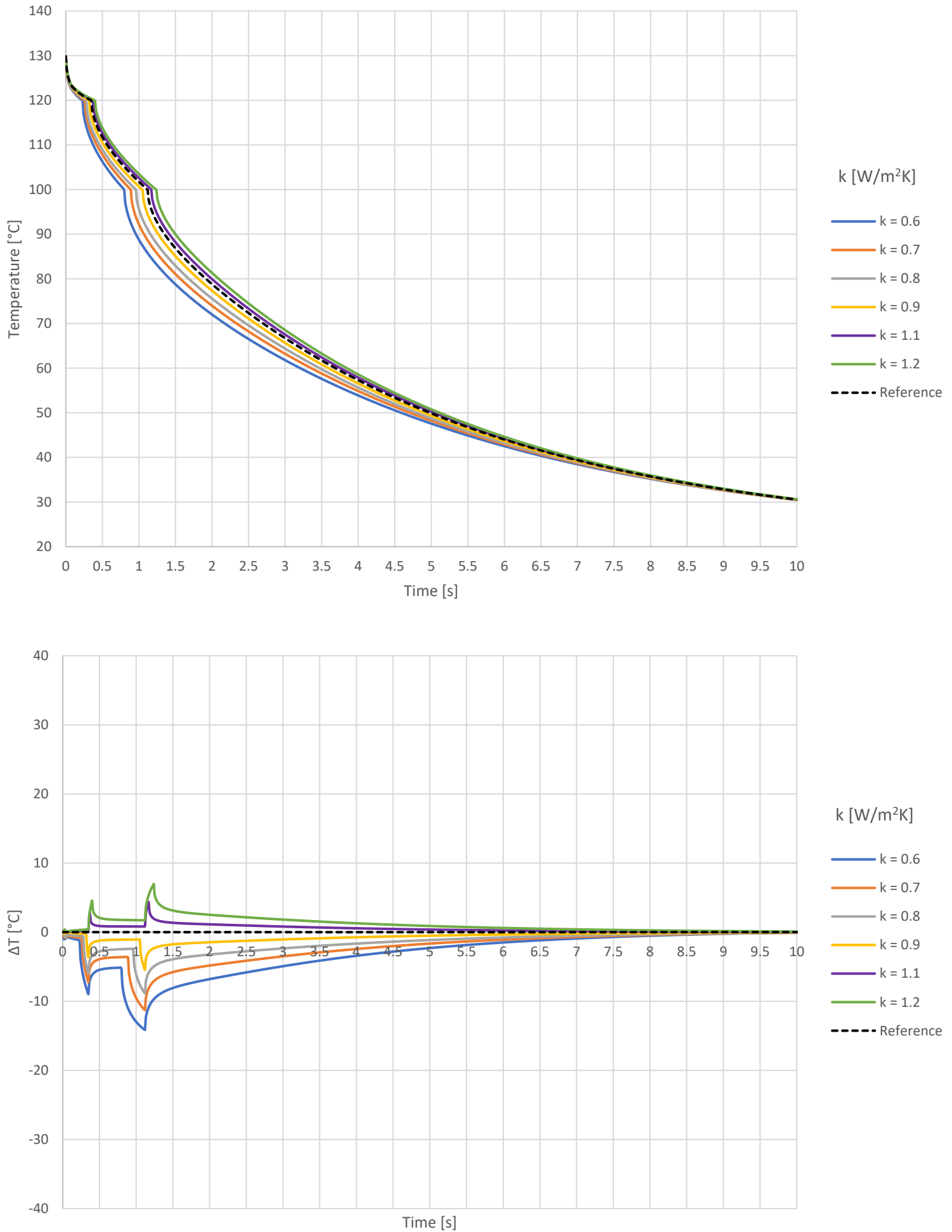
Core temperature developments (top) and temperature differences (bottom):



Critical temperature developments (top) and temperature differences (bottom):



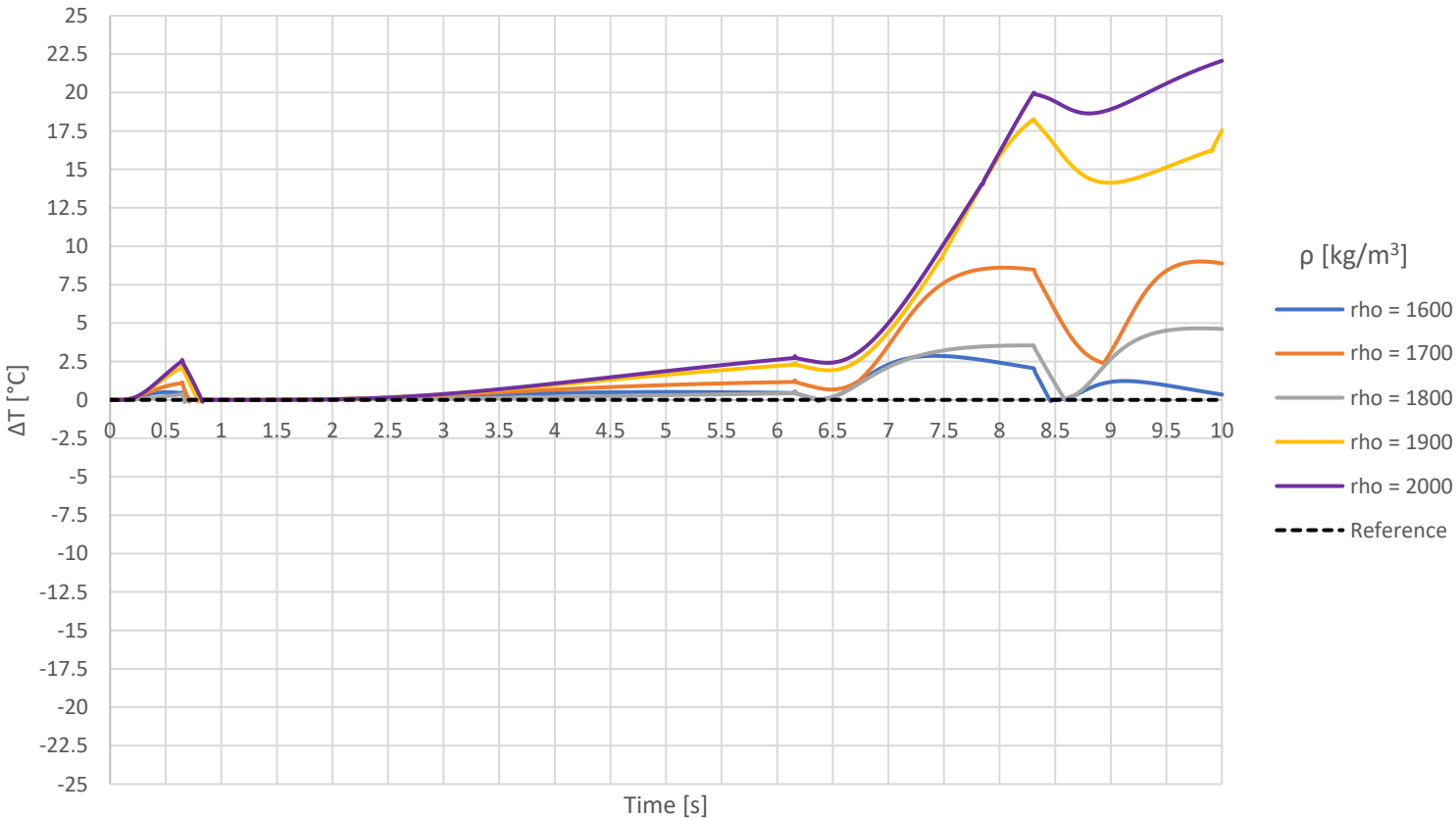
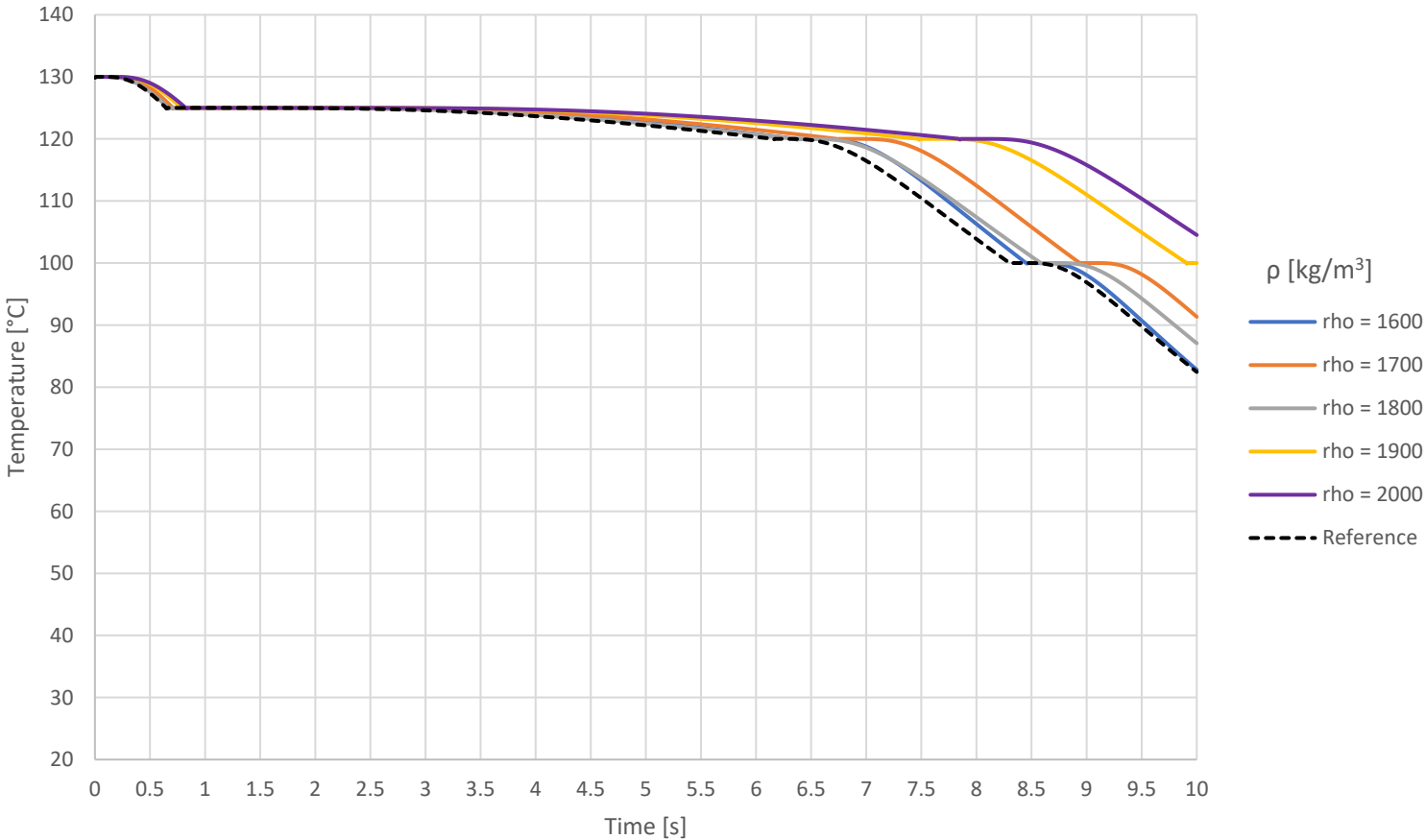
Surface temperature developments (top) and temperature differences (bottom):



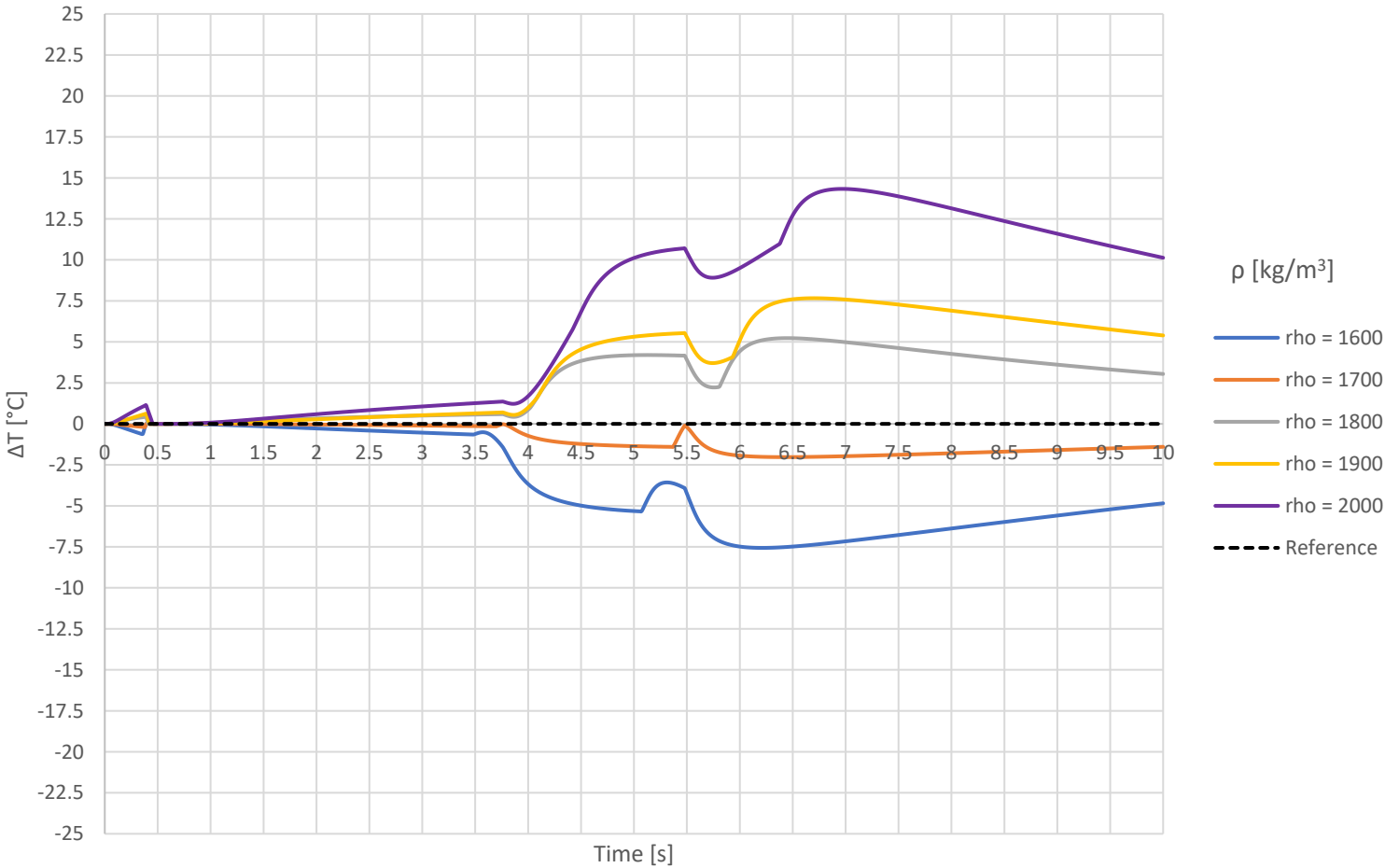
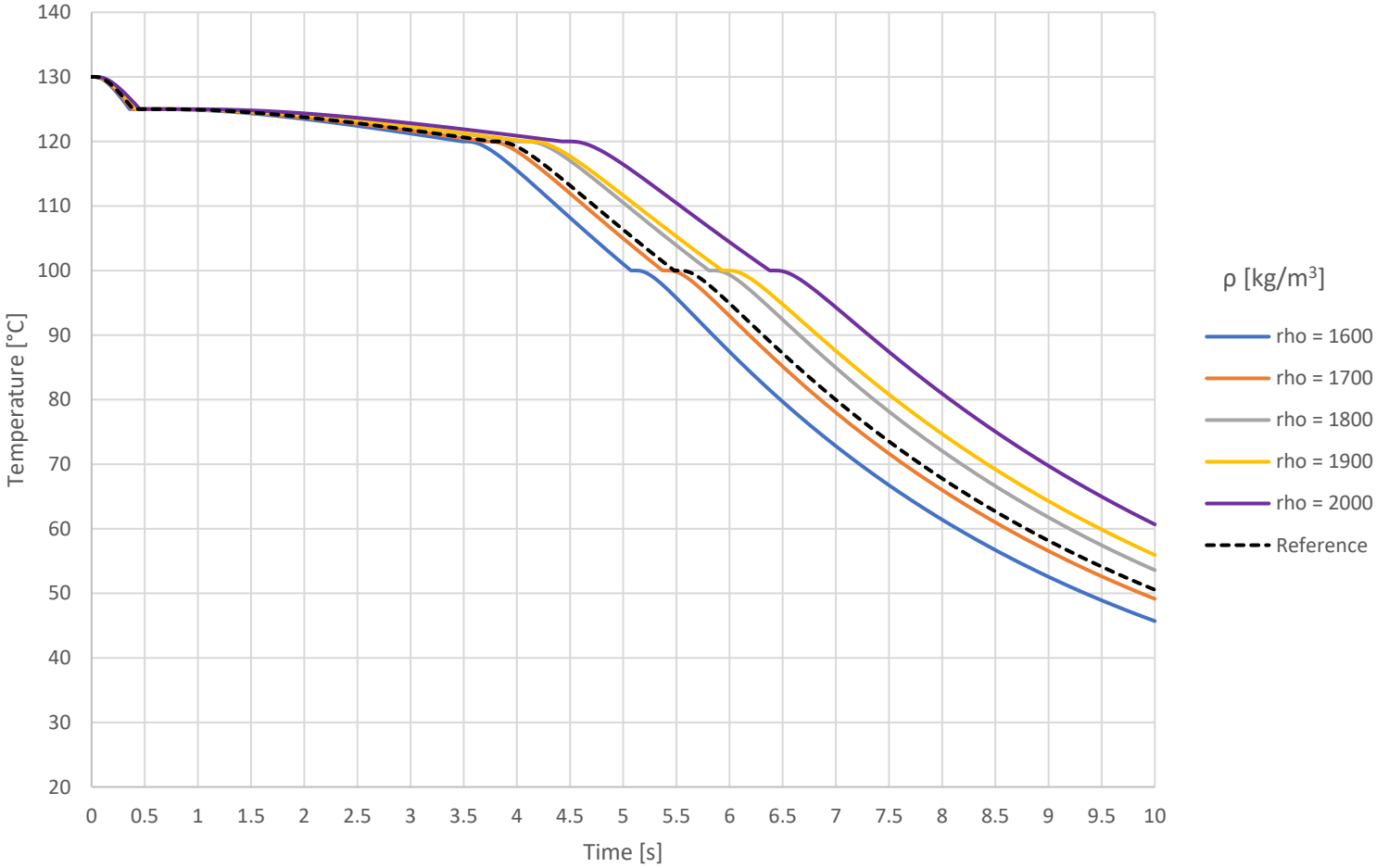
# Particle density

$\rho_{p, ref} = 1747 \text{ kg/m}^3$

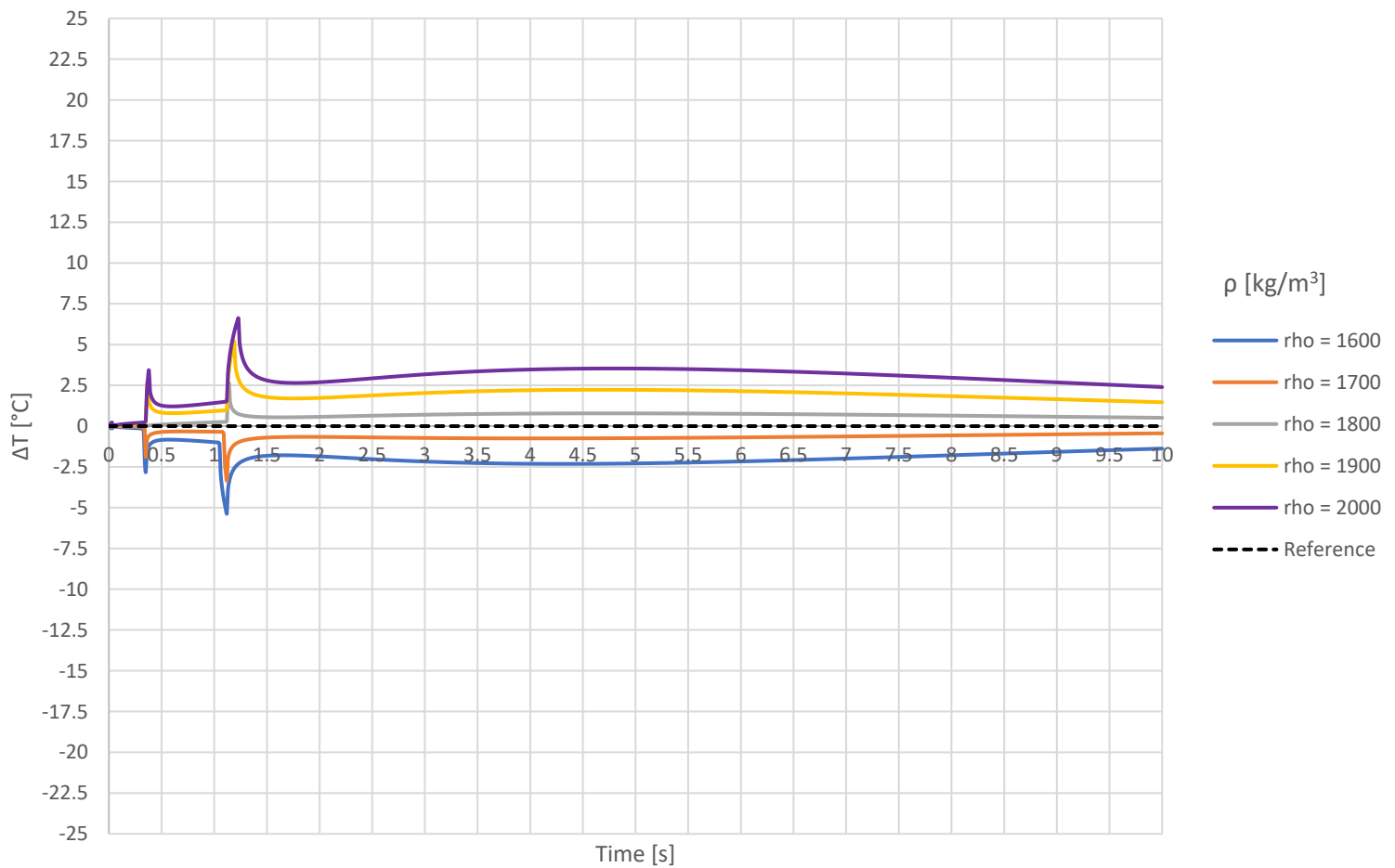
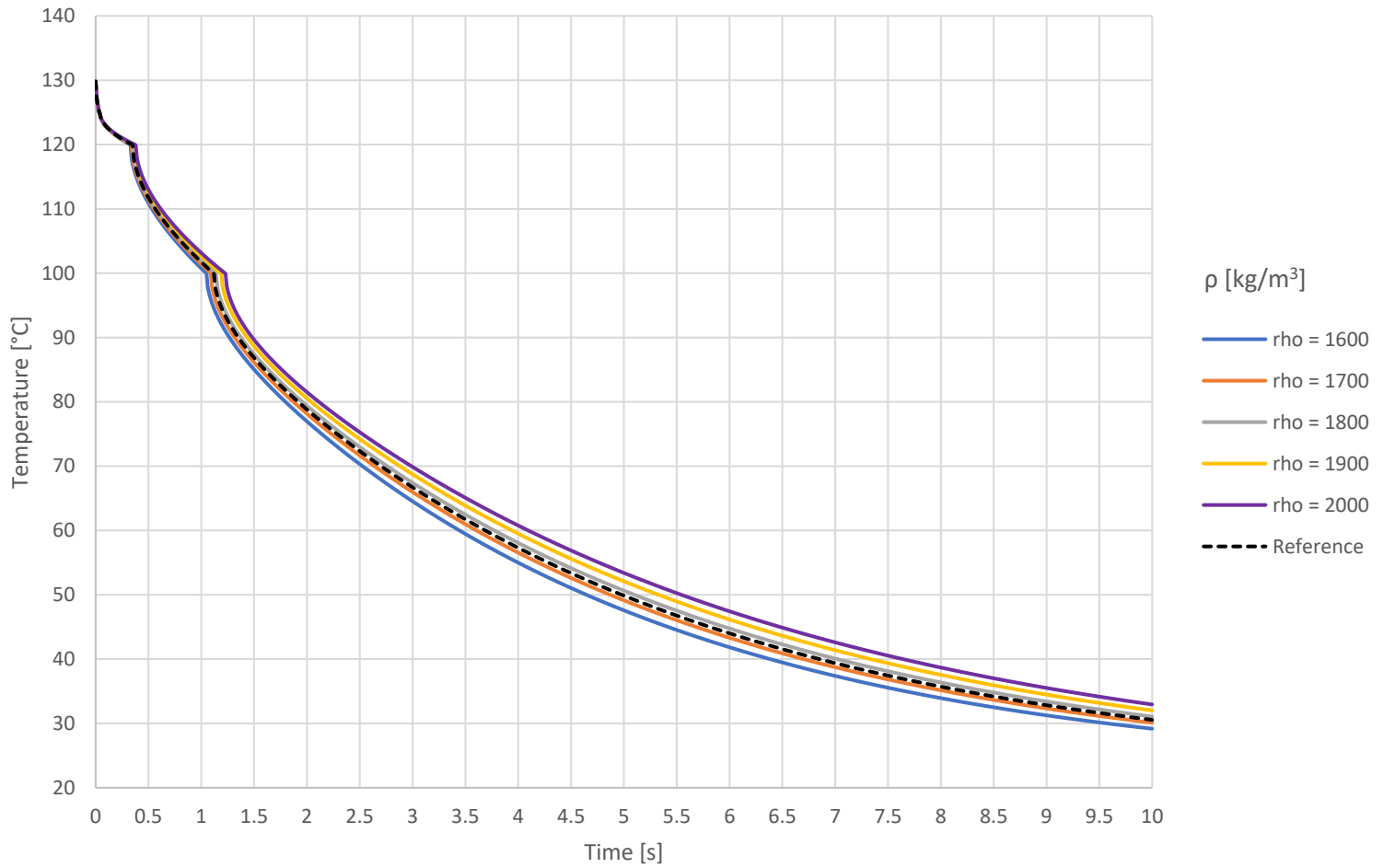
Core temperature developments (top) and temperature differences (bottom):



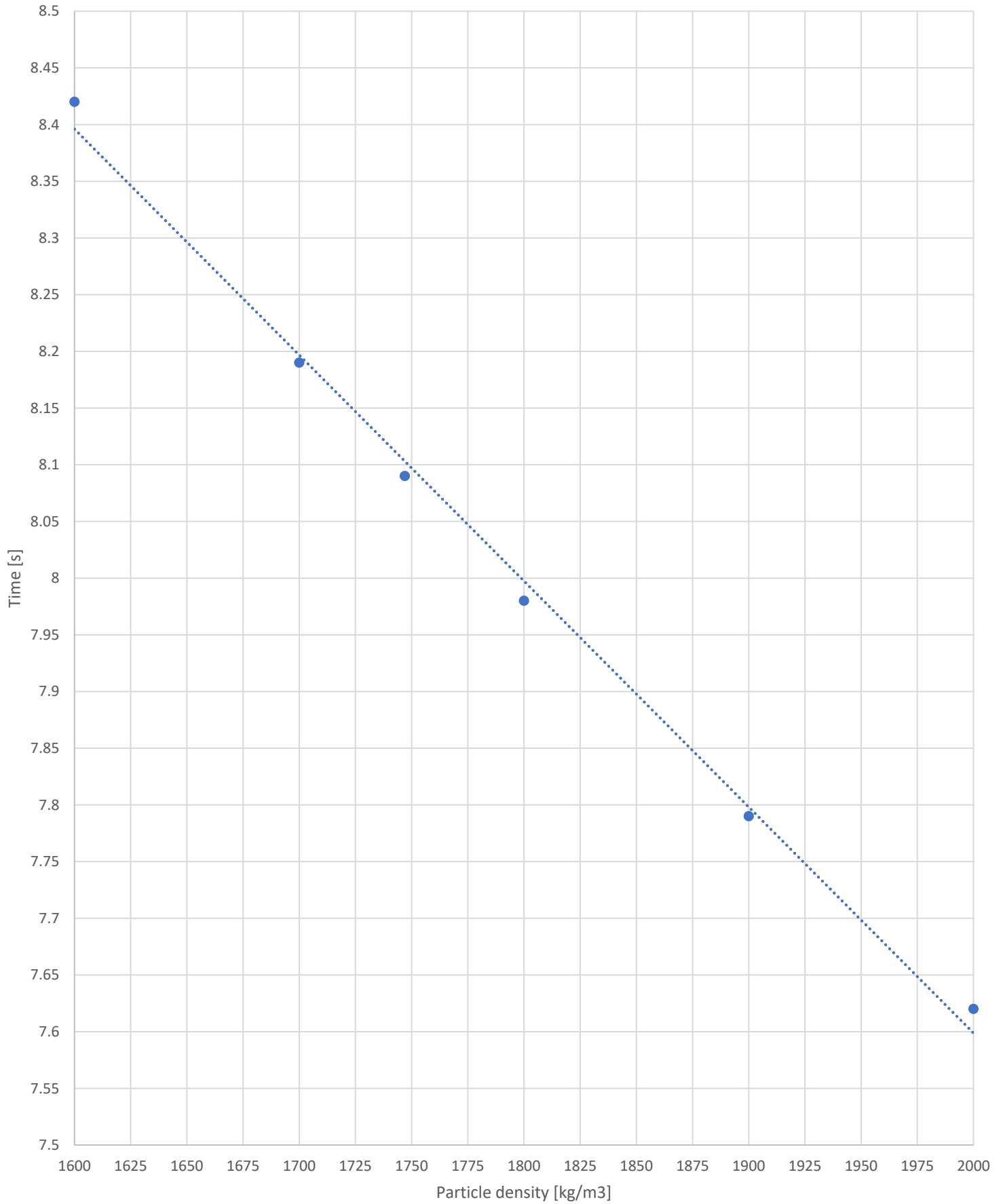
Critical temperature developments (top) and temperature differences (bottom):



Surface temperature developments (top) and temperature differences (bottom):



*Time-of-flights:*



$$T.O.F = -0.002 \times \rho_p + 11.585$$

$$R^2 = 0.9961$$



# Appendix C

## AN-MAP-Water phase diagram

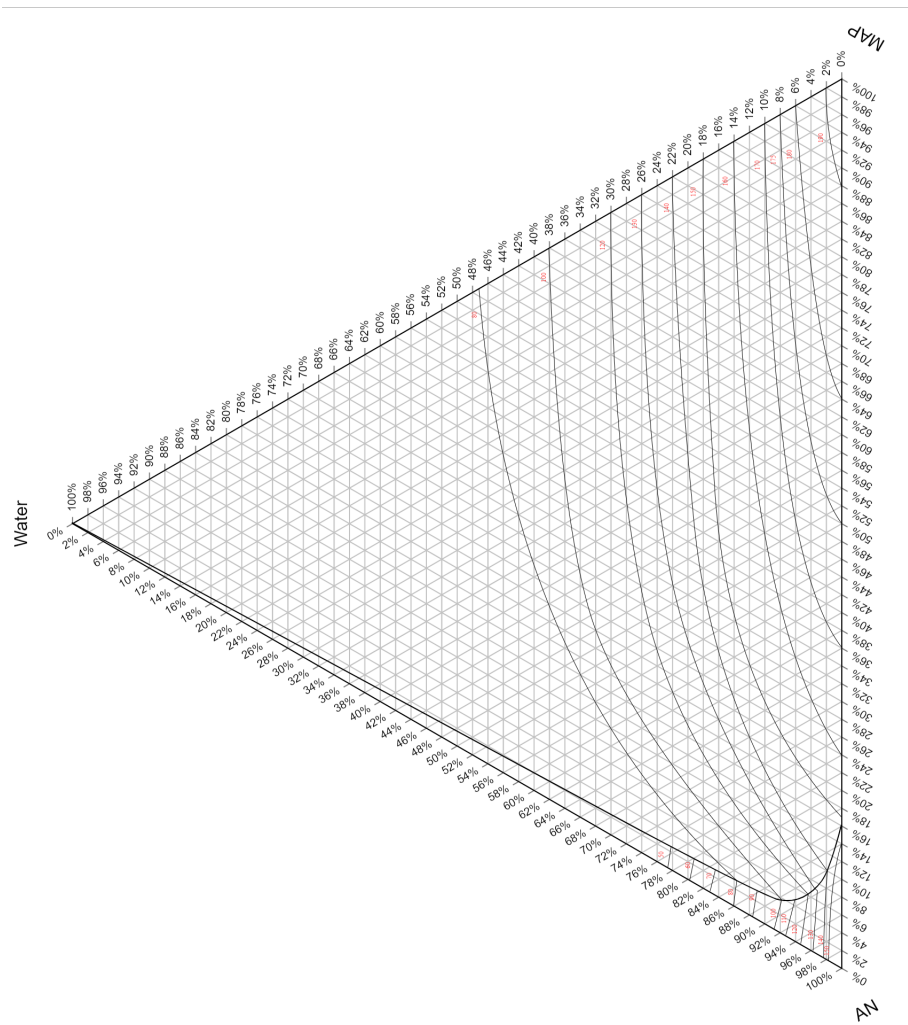


Figure C.1: AN-MAP-Water phase diagram. Phase diagram was made digital by the author, but all information was gathered from Stokka, P. (1985) [20].



# Appendix D

## NPK product data

The following appendix gives insight into where the product data from section 6.4 was gathered. The information is shown in the form of phase diagrams, liquid-/ solid-fractions and composition at different points. The phase diagram that is being used is identical to the one in Appendix C. The phase diagrams were constructed using product recipes from Yara International, which will not be disclosed in this thesis. Due to the setup of the phase diagrams, only portions of them will be shown in the figures. This may lead to some confusion when reading off values from the sides, however, a general layout is shown in figure D.1 below to help the reader.

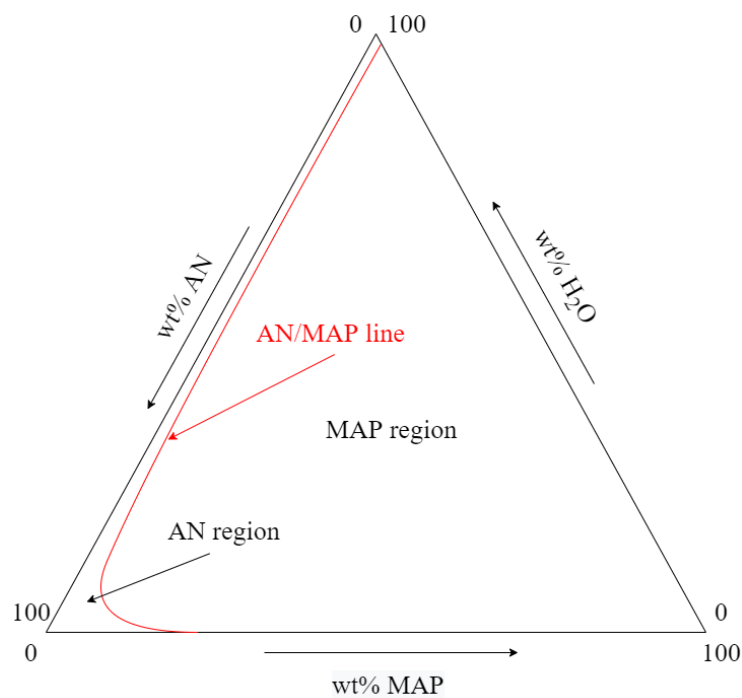


Figure D.1: Simplified sketch of the AN-MAP-Water phase diagram, can be used for reference when examining figures in Appendix B.

## D.1 NPK 25-2-6

The AN-MAP-Water phase diagram for NPK 25-2-6 is shown in figure D.2 below:

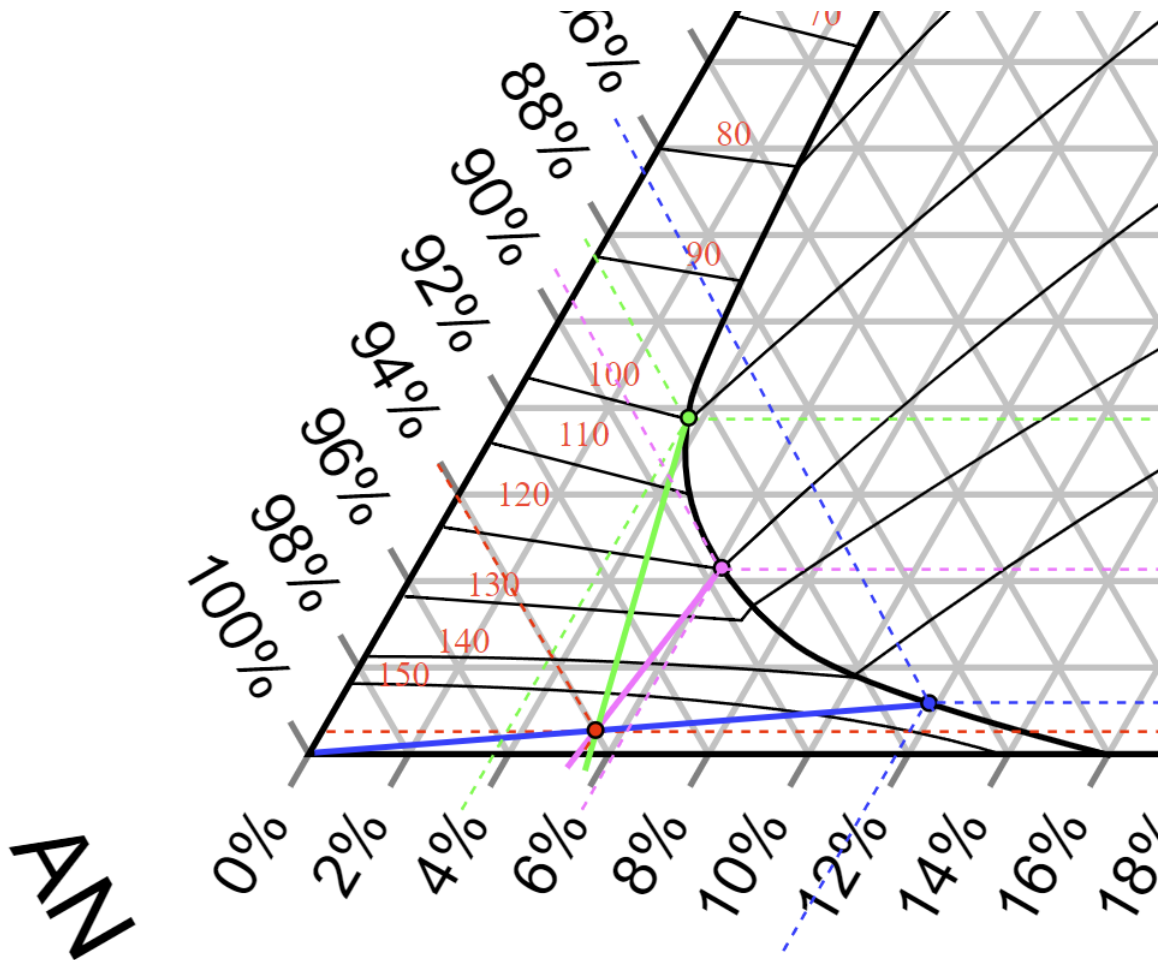


Figure D.2: AN-MAP-Water phase diagram for NPK 25-2-6

The following data was gathered from the phase diagram:

- Crystallization start (red dot):
  - $T = 156^{\circ}\text{C}$
  - Crystallizing compound: AN
- Initial MAP + AN crystallization (blue dot):
  - $T = 145^{\circ}\text{C}$
  - Solid fraction,  $X_s = 0.54$ , only AN

- Liquid fraction,  $X_l = 0.46$ . 86.7% AN, 11.8% MAP, 1.5% H<sub>2</sub>O
- End of main crystallization (pink dot):
  - T = 120°C
  - Solid fraction,  $X_s = 0.87$ . 94.5% AN, 5.5% MAP
  - Liquid fraction,  $X_l = 0.13$ . 89.6% AN, 6.1% MAP, 4.3% H<sub>2</sub>O
- End of rest crystallization (green dot):
  - T = 100°C
  - Solid fraction,  $X_s = 0.92$ . 94.3% AN, 5.7% MAP
  - Liquid fraction,  $X_l = 0.08$ . 88.4% AN, 3.8% MAP, 7.8% H<sub>2</sub>O

## D.2 NPK 21-4-10

The AN-MAP-Water phase diagram for NPK 21-4-10 is shown in figure D.3 below:

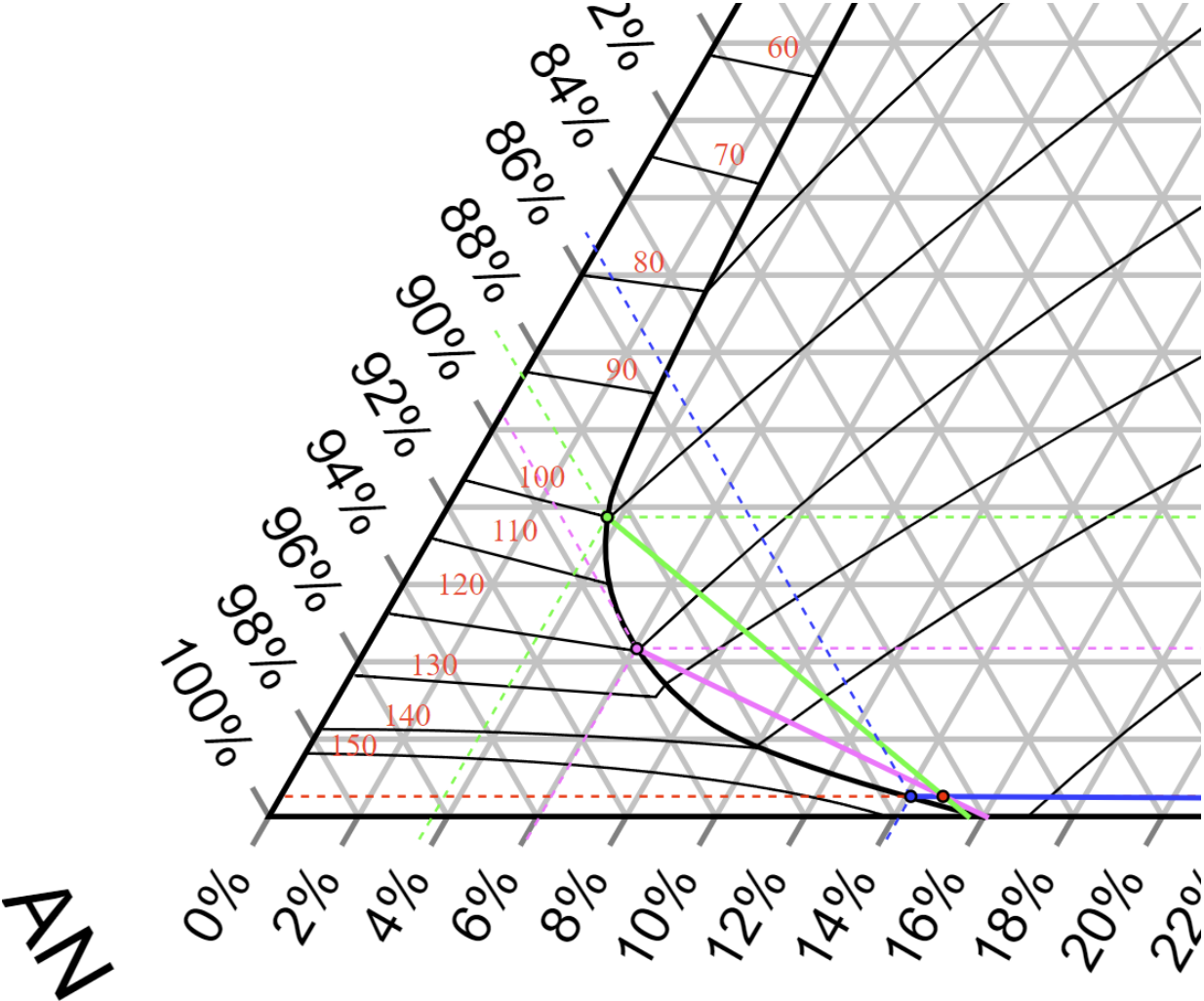


Figure D.3: AN-MAP-Water phase diagram for NPK 21-4-10

The following data was gathered from the phase diagram:

- Crystallization start (red dot):
  - T = 148°C
  - Crystallizing compound: MAP
- Initial MAP + AN crystallization (blue dot):

- $T = 146^{\circ}\text{C}$
- Solid fraction,  $X_s = 0.0075$ , only MAP
- Liquid fraction,  $X_l = 0.9925$ . 85.4% AN, 14.1% MAP, 0.5%  $\text{H}_2\text{O}$
- End of main crystallization (pink dot):
  - $T = 120^{\circ}\text{C}$
  - Solid fraction,  $X_s = 0.86$ . 84.0% AN, 16.0% MAP
  - Liquid fraction,  $X_l = 0.14$ . 89.6% AN, 6.1% MAP, 4.3%  $\text{H}_2\text{O}$
- End of rest crystallization (green dot):
  - $T = 100^{\circ}\text{C}$
  - Solid fraction,  $X_s = 0.94$ . 84.5% AN, 15.5% MAP
  - Liquid fraction,  $X_l = 0.06$ . 88.4% AN, 3.8% MAP, 7.8%  $\text{H}_2\text{O}$

### D.3 NPK 21-06-13

The AN-MAP-Water phase diagram for NPK 21-06-13 is shown in figure D.4 below:

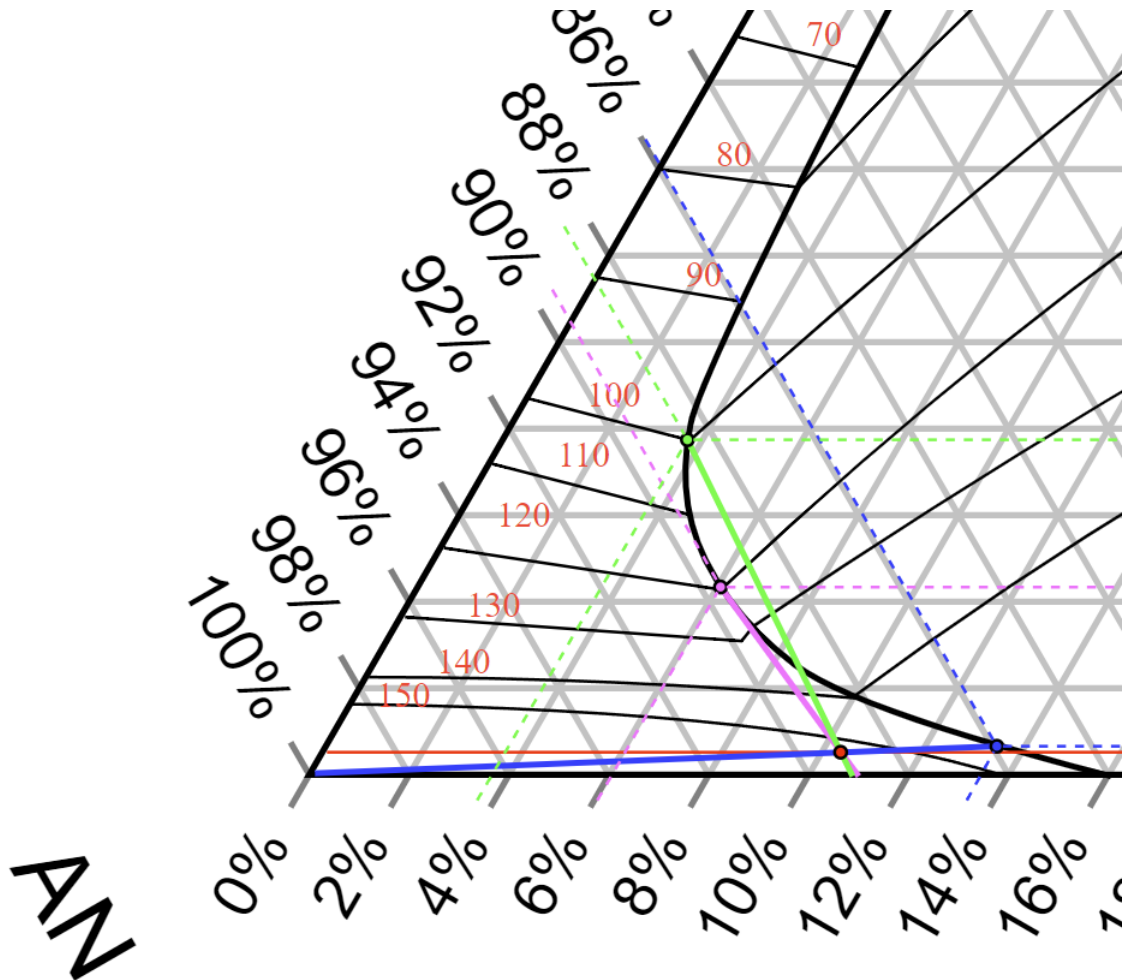


Figure D.4: AN-MAP-Water phase diagram for NPK 21-06-13

The following data was gathered from the phase diagram:

- Crystallization start (red dot):
  - $T = 152^{\circ}\text{C}$
  - Crystallizing compound: AN
- Initial MAP + AN crystallization (blue dot):
  - $T = 145^{\circ}\text{C}$
  - Solid fraction,  $X_s = 0.228$ , only AN



- Liquid fraction,  $X_l = 0.772$ . 85.9% AN, 13.5% MAP, 0.6% H<sub>2</sub>O
- End of main crystallization (pink dot):
  - T = 120°C
  - Solid fraction,  $X_s = 0.87$ . 89.0% AN, 11.0% MAP
  - Liquid fraction,  $X_l = 0.13$ . 89.6% AN, 6.1% MAP, 4.3% H<sub>2</sub>O
- End of rest crystallization (green dot):
  - T = 100°C
  - Solid fraction,  $X_s = 0.93$ . 89.2% AN, 10.8% MAP
  - Liquid fraction,  $X_l = 0.07$ . 88.4% AN, 3.8% MAP, 7.8% H<sub>2</sub>O

## D.4 NPK 25-07-07

The AN-MAP-Water phase diagram for NPK 25-07-07 is shown in figure D.5 below:

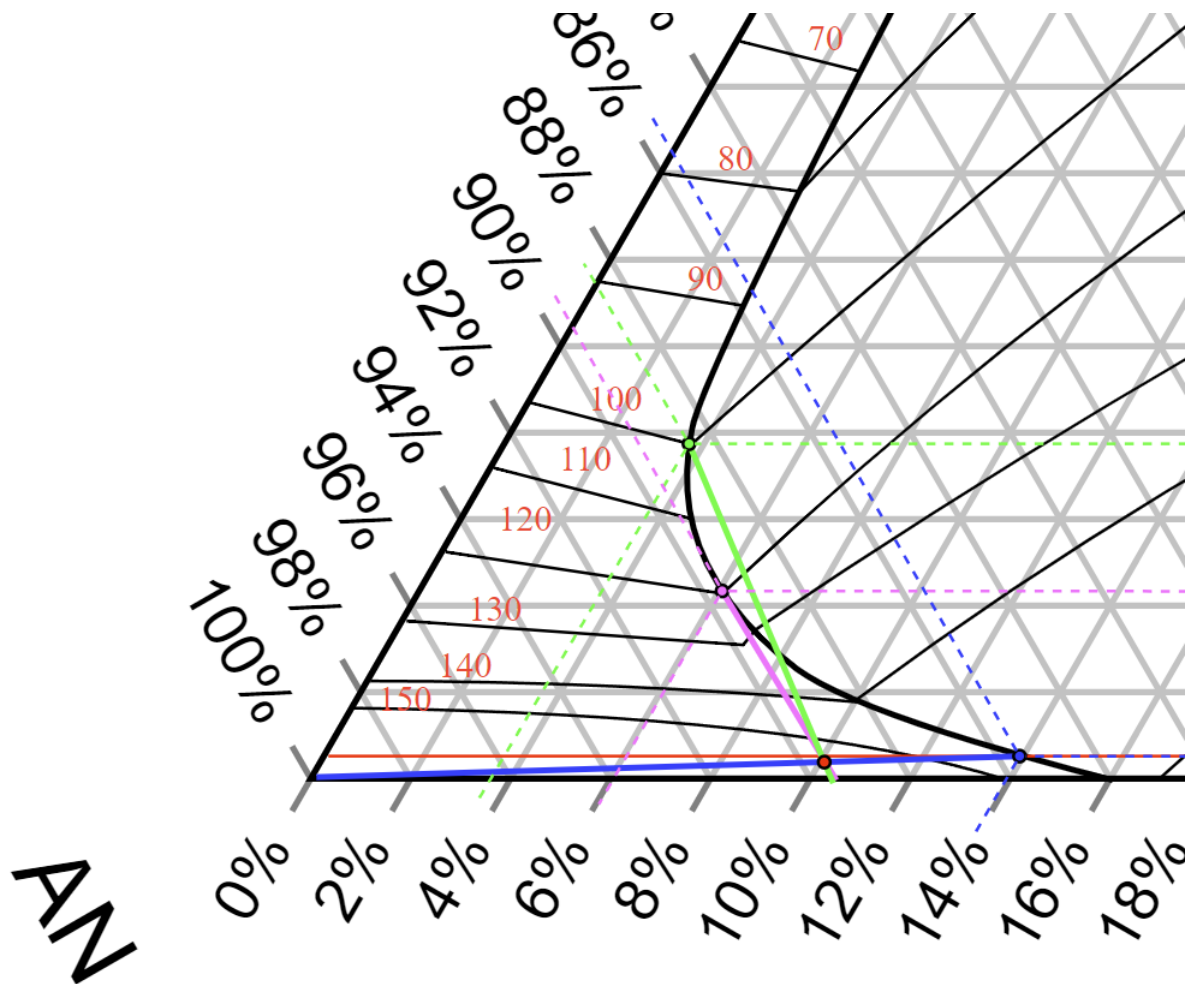


Figure D.5: AN-MAP-Water phase diagram for NPK 25-07-07

The following data was gathered from the phase diagram:

- Crystallization start (red dot):
  - $T = 154^{\circ}\text{C}$
  - Crystallizing compound: AN
- Initial MAP + AN crystallization (blue dot):
  - $T = 147^{\circ}\text{C}$
  - Solid fraction,  $X_s = 0.275$ , only AN

- Liquid fraction,  $X_l = 0.725$ . 85.6% AN, 13.9% MAP, 0.5% H<sub>2</sub>O
- End of main crystallization (pink dot):
  - T = 120°C
  - Solid fraction,  $X_s = 0.925$ . 89.7% AN, 10.3% MAP
  - Liquid fraction,  $X_l = 0.075$ . 89.6% AN, 6.1% MAP, 4.3% H<sub>2</sub>O
- End of rest crystallization (green dot):
  - T = 100°C
  - Solid fraction,  $X_s = 0.955$ . 89.8% AN, 10.2% MAP
  - Liquid fraction,  $X_l = 0.045$ . 88.4% AN, 3.8% MAP, 7.8% H<sub>2</sub>O



# Appendix E

## Python model source code

### E.1 Composite Ytterbø model

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Sat Feb 19 18:39:26 2022
4
5 @author: Anders
6 """
7 import numpy as np
8 import matplotlib.pyplot as plt
9 import pandas as pd
10
11
12 # Particle parameters
13 d_p = 2.85
14     # Particle diameter [mm]
15 A_p = 4*np.pi*(d_p*1e-3/2)**2
16     # Particle surface area[m2]
17 V_p = 4/3*np.pi*((d_p*1e-3)/2)**3
18     # Particle volume [m3]
19 k = 1
20     # Thermal conductivity [W/m*K]
21 r_o = (d_p*1e-3)/2
22     # Outer radius of particle [m]
23
24 # Tower parameters
25 r_tower = 12
26     # Tower radius [m]
27 A_tower = np.pi*r_tower**2
28     # Tower cross-sectional area [m2]
29 tower_height = 43.4
30     # Tower height [m]
31
32 # Operating parameters
33 m_dot_slurry = 1.06e5
34     # Mass flow of slurry exiting the centrifuge [kg/h]
```

```

26 m_dot_air = 1.16e6
    # Mass flow of air into the tower [kg/h]
27 T_inf = 22
    # Air temperature at inlet [degC]
28 p_air = 101325
    # Air pressure [Pa]
29 T_i = 130
    # Initial temperature of slurry [degC]
30 RPM = 390
    # Centrifuge rotations-per-minute [1/min]
31
32
33 # Crystallization parameters
34 T_1 = 125
    # Temperature where main crystallization starts [degC]
35 T_2 = 120
    # Temperature where rest crystallization starts [degC]
36 T_3 = 100
    # Temperature where rest crystallization ends [degC]
37 T_4 = T_inf
    # Temperature where crystallization has stopped [degC]
38 Q_cr = 108
    # Heat of crystallization [kJ/(kg*K)]. Taken from Stokka (1985)
39
40 # Simulation paramaters
41 time = 10
    # Simulation time [s]
42 dt = 0.01
    # Time step [s]
43 N_t = int(time/dt)
    # Number of iterations
44 t = np.linspace(0, N_t*dt, N_t+1)
    # Time interval, evenly distributed
45
46 ''' Fluid parameters '''
47 # List of air properties at different temperatures
48 # All values are taken from Fundamentals of Heat and Mass Transfer,
    appendix A.4.
49 T_val = [250, 300, 350]
    # Temperature [K]
50 cp_val = [1006, 1007, 1009]
    # Heat capacity [J/(kg*K)]
51 mu_val = [159.6e-7 , 184.6e-7 , 208.2e-7]
    # Dynamic viscosity [(N*s)/m2]
52 k_val = [22.3e-3, 26.3e-3, 30.0e-3]
    # Thermal conductivity [W/(m*K)]
53 a_val = [15.9e-6 , 22.5e-6 , 29.9e-6]
    # Thermal diffusivity [m2/s]
54
55 # This function performs linear interpolation to estimate the properties
56 # of air at a temperature between certain intervals. Inputs to the function

```

```

    are:
57 # x = the property one wishes to evaluate
58 # (ex. heat conductivity, diffusivity etc... See list above)
59 # i = list index of the lower value range.
60 # E.g. if the temperature one wishes to assess is 330K,
61 # i would be 1, since the lower value range of the interpolation is 300k,
62 # which is located at index 1.
63 def lin(x, i):
64     dT= (T_inf+273.15)-T_val[i]
65     # Temperature difference [K]
66     int_val = x[i] + ((x[i+1]-x[i])/(T_val[i+1]-T_val[i]))*dT
67     # Linear interpolation formula
68     return int_val
69     # Function returns the linearly interpolated value of x (variable)
70
71 # This function returns all necessary air parameters at a given temperature
72 # (NO INPUT REQUIRED). This formula automatically adjusts based on air
73 # temperature,
74 # which has already been declared in operating parameters.
75 def air_val():
76     if T_inf+273 < 300:
77         cp_air = lin(cp_val, 0)
78         # Heat capacity of air [kJ/kg*K]
79         mu_air = lin(mu_val, 0)
80         # Dynamic viscosity of air [N*s/m2]
81         k_air = lin(k_val, 0)
82         # Thermal conductivity of air [W/m*K]
83         a_air = lin(a_val, 0)
84         # Thermal diffusivity of air [m2/s]
85
86     if 300 < T_inf+273 < 350:
87         cp_air = lin(cp_val, 1)
88         # Heat capacity of air [kJ/kg*K]
89         mu_air = lin(mu_val, 1)
90         # Heat capacity of air [kJ/kg*K]
91         k_air = lin(k_val, 1)
92         # Heat capacity of air [kJ/kg*K]
93         a_air = lin(a_val, 1)
94         # Heat capacity of air [kJ/kg*K]
95
96     return cp_air, mu_air, k_air, a_air
97     # When function is called, return heat capacity, dynamic viscosity,
98     # thermal conductivity and thermal diffusivity
99
100 # This function calculates the density of air.
101 # The formula uses the mass-modified ideal gas law (PV=nRT) to calculate
102 # density.
103 # calculations are based on two variables, air temperature and pressure.
104 # These are not defined within the function (NO INPUT REQUIRED)
105 # Temperature and pressure can be found in the operating parameters section

```

```

91 def density():
92     Mm_O = 16.00/1000
93     # Molar mass of elemental oxygen, O [kg/mol]
94     Mm_N = 14.01/1000
95     # Molar mass of elemental oxygen, N [kg/mol]
96     Mm_O2 = 2 * Mm_O
97     # Molar mass of oxygen, O2 [kg/mol]
98     Mm_N2 = 2 * Mm_N
99     # Molar mass of nitrogen, N2 [kg/mol]
100
101     R = 8.314
102     # Universal gas constant [(m3*Pa)/(K*mol)]
103     Mm_air = 0.21 * Mm_O2 + 0.79 * Mm_N2
104     # Molar mass of air [kg/mol]
105     rho_air = (p_air*Mm_air)/(R*(T_inf+273.15))
106     # Ideal gas law for calculating air density [kg/m3] as a function
107     # of air temperature and pressure
108
109     return rho_air
110     # When called, function returns the density of air
111
112 ''' Air calculations '''
113 air_data = air_val()
114     # Function air_val is called with 0 arguments. cp_air, mu_air,
115     # k_air, a_air are returned.
116 rho_air = density()
117     # Function density is called with 0 arguments. rho_air is returned.
118
119 # All variables from the air_val function are stored as separate variables:
120 cp_air = air_data[0]
121 mu_air = air_data[1]
122 k_air = air_data[2]
123 a_air = air_data[3]
124
125 ''' Mass balance and heat capacities model '''
126 # This function calculates the heat capacities of selected intervals, based
127 # on the product composition and heat of crystallization. This method is
128 # used by Stokka in his 1985 report. This formula is divided into two main
129 # sections.
130 # The first section calculates the density of a solid particle,
131 # given a certain composition. The second section (total_heat_cap),
132 # calculates the specific combined heat capacities that arise from
133 # combining
134 # the heat capacities of the crystallization process and the innate
135 # component
136 # heat capacities.
137 def cp_calc():
138     m_NP = 100
139     # Mass basis used for further calculations. All other components
140     # are presented as a fraction of this base. [kg NP melt]

```



```

124 X_KCl = 0.27
      # Fraction of KCl [kg potassium chloride/kg NP liqour]
125 X_Kies = 0.09
      # Fraction of Kies [kg of kieserite/kg NP liqour]
126 m_KCl = X_KCl*m_NP
      # Total mass of KCl [kg]
127 m_Kies = X_Kies*m_NP
      # Total mass of Kieserite [kg]
128
129 # We input the amount of solids
130 NP_s = 9
131 KCl_s = 16
132 Kies_s = m_Kies
133 sum_s = NP_s + KCl_s + Kies_s
      # Total amount of solids, given a 100kg basis of NP liqour [kg]
134
135 # The given basis and calculated solid masses allows us to calculate
136 # the mass of liquids in the melt:
137 NP_l = m_NP - NP_s
      # Mass of liquid NP melt [kg]
138 conversion_KCl = 0.4
      # Conversion from solid to liquid KCl [-]
139 KCl_l = m_KCl * conversion_KCl
      # Mass of liquid KCl [kg]
140
141 sum_l = NP_l + KCl_l
      #Total amount of liquid, given a 100kg basis of NP liqour [kg]
142
143 # The total mass of all compounds in all phases are calculated by
144 # adding the liquid and solid components
145 m_tot = sum_l + sum_s
146
147 # Densities for the solid mixture and total mixture are based on
calculations
148 # and measured values. Sources are given where applicable.
149 rho_mix_l = 1600
      # Density of mixed liquid [kg/m3] taken from source (1), shown in
Appendix 4 in Stokka (1985)
150 rho_NP_s = 2000
      # Density of solid NP melt [kg/m3]
151 rho_KCl_s = 2000
      # Density of solid KCl [kg/m3]
152 rho_Kies_s = 2700
      # Density of solid Kieserite[kg/m3]
153
154 # The density of the solid mixture is calculated by weighting
155 # each compounds density by its fraction of the solids
156 rho_mix_s = rho_NP_s*(NP_s/sum_s) + rho_KCl_s*(KCl_s/sum_s) +
rho_Kies_s*(Kies_s/sum_s) # Density of solid mixture [kg/m3]
157 # The same procedure is done for the total mixture density
158 rho_tot = rho_mix_l*(sum_l/(m_tot)) + rho_mix_s*(sum_s/(m_tot))

```

```

159     # Total slurry density for NPK 21:4:10 [kg/m3]
160     '''Chapter 2: Appendix 4, heat capacity and temperature intervals'''
161     # We define our heat capacity function
162     # and initialize our variables:
163     def total_heat_cap(eps, m_tot, T_start, T_end):
164         m_cr = sum_l*0.99
165         # Initial mass available for crystallization. [kg] 1% of the liquid
166         # is set to stay a liquid. The available mass is therefore 99% of the sum
167         # of liquids.
168         X_s = sum_s/m_tot
169         # Mass fraction of solids [kg/kg]
170         X_l = sum_l/m_tot
171         # Mass fraction of liquids [kg/kg]
172         Cp_NP_s = 0
173         # Specific heat capacity of solid NP liquor (if applicable)
174         Cp_KCl_s = 1.5
175         # Specific heat capacity of solid potassium chloride
176         Cp_Kies_s = 1.0
177         # Specific heat capacity of solid Kieserite
178         Cp_NP_l = 2.0
179         # Specific heat capacity of liquid NP liquor
180         Cp_KCl_l = 2.0
181         # Specific heat capacity of liquid potassium chloride
182         Cp_Kies_l = 0
183         # Specific heat capacity of liquid Kieserite (if applicable)
184         Cp_s = Cp_NP_s*(NP_s/sum_s) + Cp_KCl_s*(KCl_s/sum_s) + Cp_Kies_s*(
185         Kies_s/sum_s) # Total heat capacity of the solids, weighted formula
186         Cp_l = Cp_NP_l*(NP_l/sum_l) + Cp_KCl_l*(KCl_l/sum_l)
187         # Total heat capacity of the liquids, weighted formula
188
189         # Core function
190         Cp_t = (m_cr*eps*Q_cr)/(m_tot*(T_start-T_end))+((Cp_s*X_s)+(Cp_l*
191         X_l)) # Total heat capacity of the mixture, weighted formula
192         return Cp_t
193         # function returns the total heat capacity, based on degree of
194         # crystallization, total mass and temperature interval
195
196     Cp_t_arr = [] # We initialize an array that the total heat capacity
197     # values can be stored in.
198     # In the following lines, we call the heat capacity function (
199     # total_heat_cap)
200     # a total of four times, one for each interval. These calculated in
201     # turn,
202     # and the sum of liquids and solids are updated (because of the
203     # solidification/crystallization)
204     # All values are appended to the empty array above.
205
206     # Interval 1: T_i to T_1
207     eps = 0
208     # Degree of crystallization in interval 1

```

```

188 Cp_t_arr.append(total_heat_cap(eps, m_tot, T_i, T_1))
      # Call function total_heat_cap for interval 1
189 sum_s = sum_s+(eps*sum_l)
      # Update the sum of solids
190 sum_l = sum_l-(eps*sum_l)
      # Update the sum of liquids
191
192 # Interval 2: T_1 to T_2
193 eps = 0.80
      # Degree of crystallization in interval 2
194 Cp_t_arr.append(total_heat_cap(eps, m_tot, T_1, T_2))
      # Call function total_heat_cap for interval 2
195 sum_s = sum_s+(eps*sum_l)
      # Update the sum of solids
196 sum_l = sum_l-(eps*sum_l)
      # Update the sum of liquids
197
198 # Interval 3: T_2 to T_3
199 eps = 0.95
      # Degree of crystallization in interval 3
200 Cp_t_arr.append(total_heat_cap(eps, m_tot, T_2, T_3))
      # Call function total_heat_cap for interval 3
201 sum_s = sum_s+(eps*sum_l)
      # Update the sum of solids
202 sum_l = sum_l-(eps*sum_l)
      # Update the sum of liquids
203
204 # Interval 4: T_3 to T_4
205 eps = 1
      # Degree of crystallization in interval 4
206 Cp_t_arr.append(total_heat_cap(eps, m_tot, T_3, T_4))
      # Call function total_heat_cap for interval 4
207 sum_s = sum_s+(eps*sum_l)
      # Update the sum of solids
208 sum_l = sum_l-(eps*sum_l)
      # Update the sum of liquids
209
210 return Cp_t_arr, rho_tot
      # Function returns the total heat capacity values (one for each
      # interval) and the density. Heat capacities and density are saved as
      # different lists.
211
212 ''' Mass balance and heat capacities calculations '''
213 cp_data = cp_calc()
      # Call function cp_calc, without any parameters
214 cp = []
      # Initialize an empty array to store each Cp value in
215 # Here, we define a for loop that iterates through the heat capacities,
216 # and converts them from kJ to J.
217 for i in cp_data[0]:
218     cp.append(i*1000)

```

```

219 cp_1 = cp[0]
      # Save the first index (0) in the list cp as cp_1
220 cp_2 = cp[1]
      # Save the second index (1) in the list cp as cp_2
221 cp_3 = cp[2]
      # Save the third index (2) in the list cp as cp_3
222 cp_4 = cp[3]
      # Save the fourth index (3) in the list cp as cp_4
223 rho_p = cp_data[1]
      # Save the second index (1) in the list cp_data as rho_p
224 m = rho_p*V_p
      # Mass of particle [kg]
225
226 ''' Velocity model'''
227 def vel_and_pos(RPM, rho_air, time, dt):
228     # Velocity specific parameters
229     theta = 20
230         # Centrifuge angle [deg]
231     C_d = 0.44
232         # Drag coefficient
233     r_eject = 0.1
234         # Discharge radius [m]
235     r_i = 0.265
236         # Inner (smallest) radius of centrifuge [m]
237     g = 9.81
238         # Gravitational constant [m/s2] POSITIVE DIRECTION DOWNWARDS
239     K = 1/2*rho_air*C_d*A_p
240         # Friction drag constant [kg/m]
241     F_b = V_p*rho_air*g
242         # Buoyancy force, using V_p because particle is fully submerged
243
244     # Initialization of variable storage
245     ux = np.zeros(N_t+1)
246         # Empty horizontal velocity array
247     uy = np.zeros(N_t+1)
248         # Empty vertical velocity array
249     u = np.zeros(N_t+1)
250         # Empty total velocity array
251     x = np.zeros(N_t+1)
252         # Empty horizontal coordinate array
253     y = np.zeros(N_t+1)
254         # Empty vertical coordinate array
255
256     # Initial conditions
257     ux[0] = (RPM/60)*2*np.pi*r_eject
258         # Initial horizontal velocity [m/s]. It is assumed here that the
259         # particles the bucket in a straight horizontal line.
260     uy[0] = 0
261         # Initial vertical velocity [m/s]
262     u[0] = ux[0]
263         # Initial total velocity [m/s]

```

```

248     x[0] = 0 #r_eject
        # Initial horizontal position [m]. This will be equal to the
        radius of the bucket where the prill is ejected.
249     y[0] = 0 #-np.tan(theta)*(r_eject-r_i)
        # Initial vertical position [m].
250
251     # Step equations forward in time
252     for n in range(N_t):
253         # Looping through the equations below N_t number of times.
        d_ux_dt = - (K*(u[n]*ux[n]))/m
        # Diff.eq for velocity change in horizontal direction. Only
        affected by drag force.
254         d_uy_dt = g - (F_b + K*(u[n]*uy[n]))/m
        # Diff.eq for velocity change in vertical direction. Affected by
        gravitational force, drag force and buoyancy force.
255         ux[n+1] = ux[n] + d_ux_dt*dt
        # Euler scheme for predicting horizontal velocity at next time step
        (n+1).
256         uy[n+1] = uy[n] + d_uy_dt*dt
        # Euler scheme for predicting vertical velocity at next time step (
        n+1).
257         u[n+1] = np.sqrt(ux[n+1]**2+uy[n+1]**2)
        # Calculating the total velocity at next time step (n+1)
258         x[n+1] = x[n]+ux[n]*dt
        # Calculating horizontal position at next time step (n+1)
259         y[n+1] = y[n]+uy[n]*dt
        # Calculating vertical position at next time step (n+1)
260     return [ux, uy, u, x, y, t]
        # When function is called, return one list with a set of sub lists
        containing ux, uy, u, x, y and time t.
261
262 ''' Velocity calculations'''
263
264 v_data = vel_and_pos(RPM, density(), time, dt)
        # Call function vel_and_pos, using RPM, density function, time and
        step time as input variables.
265 ux = v_data[0]
        # Horizontal velocity [m/s]. Fetched from list v_data, index 0.
266 uy = v_data[1]
        # Vertical velocity [m/s]. Fetched from list v_data, index 1.
267 u = v_data[2]
        # Total velocity [m/s]. Fetched from list v_data, index 2.
268 x = v_data[3]
        # Horizontal position [m]. Fetched from list v_data, index 3.
269 y = v_data[4]
        # Vertical position [m]. Fetched from list v_data, index 4.
270 t = v_data[5]
        # Time [s]. Fetched from list v_data, index 5.
271
272 # In this section, we want to locate the exact time (index) where
273 # the particles vertical position (y) is equal to the tower height.

```

```

274 # This is to say that we want to fetch the time where the particle hits the
      floor.
275 # We start by defining a function called Index (using lambda) and then call
      it.
276 # The index is stored in the variable t_index
277 Index = lambda a, val, tol: next(i for i, _ in enumerate(a) if np.isclose(_
      , val, tol))
278 t_index = Index(y, tower_height, 0.01)
279
280 ''' Convection model '''
281 # This formula calculates an average heat transfer coefficient based on the
282 # particle velocity and air density. The particle velocity is added to the
      air velocity
283 # which gives us a relative velocity, considering the case where the
      particle
284 # is standing still and the air is moving at a relative speed around it.
285 # The velocity contains N_t number of values, and is therefore averaged.
286 def h_conv(u, rho_air):
287     u_air = m_dot_air*1/3600*(1/rho_air)*(1/A_tower)
          # Air velocity in the tower [m/s] ASSUMED CONSTANT
288     u_air_relative = np.asarray(u + u_air)
          # Relative air velocity, considering a frame of reference where the
          particle is stationary
289     L_c = d_p*1e-3
          # Characteristic length EQUAL TO PARTICLE DIAMETER IN THIS CASE [m]
290
291     #Calculating Reynolds number, Prandtl number and Nusselts number
292     Re = (rho_air*u_air_relative*L_c)/mu_air
          # Reynolds number [-]
293     Pr = (cp_air*mu_air)/k_air
          # Prandtl number [-]
294     Nu = 2.0 + 0.6*Re**(1/2)*Pr**(1/3)
          # Nusselts number [-]
295     # Calculating convection heat coefficient
296     h_c = Nu*(k_air/(L_c))
          # Convection heat coefficient [W/(m2*K)]
297     # Constant radiation heat coefficient
298     h_r = 10
          # Radiation heat coefficient [W/(m2*K)] ASSUMED CONSTANT
299     # Total and average heat coefficient
300     h_tot = h_c + h_r
          # Total heat coefficient. This will be a list of values, one for
          each Nu.
301     h_ave = np.average(h_tot)
          # Average total heat coefficient.
302     return h_ave
          # When function is called, return h_ave
303
304 ''' Convection calculations'''
305 h_ave = h_conv(u, density())
          # Call function h_conv, with velocity and density function as

```

```

parameters
306 Bi = (h_ave*(d_p*1e-3))/k
      # Biot number [-]
307 print('The Biot number for this case is {:.2f}'.format(Bi))
      # Print the Biot number
308
309 ''' Conduction model'''
310 # The functions below are all part of the conduction model.
311 # The conduction model itself is contained within cond_model, which is a
312 # function that takes in radius, heat capacity and initial temperature as
      input.
313 # Inside this function, several sub-functions have been defined to make
314 # calculations easier.
315 # The accuracy of the conduction model is dependent of how many terms we
      sum up.
316 # One term is created for each root that have been pre-calculated. (See
      root_calc program)
317 # Once the conduction model calculations have been made, the results can be
      fetched
318 # based on what radius the user wants to look at. This is done by calling
      temp_calc.
319 def temp_calc(r):
320     df = pd.read_excel('roots.xlsx', index_col=(0))
      # Find the excel file called 'roots.xlsx' in the current working
      directory, read column 0, convert it to DataFrame
321     roots = np.asarray(df)
      # DataFrame is converted to an array, named 'roots'
322
323     def cond_model(r, C_p, T_i):
324         alpha = k/(rho_p*C_p)
      # Thermal diffusivity [m2/s]
325         r_dless = r/r_o
      # Dimensionless radius
326         theta_r_t = []
      # Dimensionless temperature
327         Q_over_Q_o_array = []
328
329         def Fo(t):
330             Fo = (alpha*t)/(r_o**2)
      # Dimensionless time (Fourier number)
331             return Fo
332
333         def core_func(x, Fo, r):
334             const = 4*(np.sin(x)-x*np.cos(x))/(2*x-np.sin(2*x))
      # Constant term in the analytical conduction equation
335             exp = np.exp(-x**2*Fo)
      # Exponential term in the analytical conduction equation
336             f = (1/(x*r))*(np.sin(x*r))
      # Root-radius term in the analytical conduction equation
337             return f*const*exp
      # Return constant term * exponential term * root-radius term

```

```

338
339     def Q_over_Q_o(x, Fo):
340         const = 4*(np.sin(x)-x*np.cos(x))/(2*x-np.sin(2*x))
341         exp = np.exp(-x**2*Fo)
342         new = np.sin(x)-x*np.cos(x)
343         return 1-((3*const*exp)/(x**3)*new)
344
345     Q = []
346     for i in t:
347         Q.append(Q_over_Q_o(roots[0], Fo(i)))
348
349     # At this point, the analytical equation has been established. Now the
350     # function
351     # has to be carried out at each time step. Actually, it has to run a
352     # number of times
353     # at each time step. The number of times being equal to the number of
354     # roots.
355     # This is done by creating a nested for loop. First, we iterate over
356     # the time-period
357     # Then, for each time period, we iterate over each root, run the
358     # function, and add
359     # all values together for an approximate solution.
360     for i in t:
361         # Initialize iteration over the time period
362         sum_x = 0
363         # Initialize the sum of all calculated values
364         for m in roots:
365             # Initialize iteration over the roots, within the same time
366             # iteration.
367             x = core_func(m, Fo(i), r_dless)
368             # This is the term inside the summation sign of the analytical
369             # conduction equation
370             sum_x = sum_x + x
371             # Solution is added to the variable called sum_x. This
372             # happens for every iteration.
373             theta_r_t.append(sum_x)
374             # For each timestep, the sum of the analytical conduction equation
375             # is added to the dimensionless temperature variable.
376             T = np.asarray(theta_r_t)*(T_i-T_inf)+T_inf
377             # Solve the equation of theta_r_t for T
378             return T, Q
379             # Return the temperature
380
381     # Below, we call the cond_model function 4 times. ONE FOR EACH INTERVAL
382     .
383     # Each interval has a specific heat capacity and starting temperature.
384     T_t_r_1 = cond_model(r, cp_1, T_i)[0]
385     T_t_r_2 = cond_model(r, cp_2, T_1)[0]
386     T_t_r_3 = cond_model(r, cp_3, T_2)[0]
387     T_t_r_4 = cond_model(r, cp_4, T_3)[0]
388     Q_over_Q_o_1 = cond_model(r, cp_1, T_i)[1]

```



```

371 Q_over_Q_o_2 = cond_model(r, cp_2, T_i)[1]
372 Q_over_Q_o_3 = cond_model(r, cp_3, T_i)[1]
373 Q_over_Q_o_4 = cond_model(r, cp_4, T_i)[1]
374
375 Q_o_1 = rho_p*cp_1*V_p*(T_i-T_inf)
376 Q_o_2 = rho_p*cp_2*V_p*(T_1-T_inf)
377 Q_o_3 = rho_p*cp_3*V_p*(T_2-T_inf)
378 Q_o_4 = rho_p*cp_4*V_p*(T_3-T_inf)
379
380 # At this point, we have calculated four different temperature values
381 # for each timestep.
382 # We know that at a certain temperature (given under crystallization
383 # parameters) the interval changes.
384 # This means that we need to start out plotting the temperatures of
385 # interval 1,
386 # using cp_1 and T_i, and then when the temperature reaches a specific
387 # point (T_1)
388 # We need to start plotting the temperatures for the second interval.
389 # However,
390 # when the intervals change, the new interval needs to start plotting
391 # from time = 0, not time [i].
392 # This is done using the code below.
393
394 T_t_r = []
395 # Initialization of empty dimensionless temperature array
396 Q_star_1 = []
397 Q_star_2 = []
398 Q_star_3 = []
399 Q_star_4 = []
400 # This for loop goes from t = 0 to t = N_t.
401 # If the calculated temperature of the first interval parameters is
402 # lower
403 # or equal to initial temp (T_i), and higher than T_1,
404 # then add the values calculated in the FIRST INTERVAL
405 # If the function is called with radius = outer radius, then add the
406 # heat
407 # released to the 'Newton' array
408 for i in range(0, N_t+1):
409     if (T_t_r_1[i]) <= T_i and (T_t_r_1[i]) > T_1:
410         T_t_r.append(T_t_r_1[i])
411         Q_star_1.append(Q_over_Q_o_1[i]*Q_o_1)
412
413 # This for loop goes from t = 0 to t = N_t MINUS the number of values
414 # that have already been accounted for in the previous loop(s).
415 # If the calculated temperature of the second interval parameters is
416 # lower
417 # or equal to initial interval 2 (T_1) temp, and higher than T_2,
418 # then add the values calculated in the SECOND INTERVAL
419 # If the function is called with radius = outer radius, then add the
420 # heat
421 # released to the 'Newton' array

```

```

411     for i in range(0, (N_t+1)-len(T_t_r)):
412         if (T_t_r_2[i]) <= T_1 and (T_t_r_2[i]) > T_2:
413             T_t_r.append(T_t_r_2[i])
414             Q_star_2.append(Q_over_Q_o_2[i]*Q_o_2)
415
416     # This for loop goes from t = 0 to t = N_t MINUS the number of values
417     # that have already been accounted for in the previous loop(s).
418     # If the calculated temperature of the third interval parameters is
419     lower
420     # or equal to initial interval three temp (T_2) and higher than T_3,
421     # then add the values calculated in the THIRD INTERVAL
422     # If the function is called with radius = outer radius, then add the
423     heat
424     # released to the 'Newton' array
425     for i in range(0, (N_t+1)-len(T_t_r)):
426         if (T_t_r_3[i]) <= T_2 and (T_t_r_3[i]) > T_3:
427             T_t_r.append(T_t_r_3[i])
428             Q_star_3.append(Q_over_Q_o_3[i]*Q_o_3)
429
430     # This for loop goes from t = 0 to t = N_t MINUS the number of values
431     # that have already been accounted for in the previous loop(s).
432     # If the calculated temperature of the fourth interval parameters is
433     lower
434     # or equal to initial interval four temp (T_3), and higher than T_inf,
435     # then add the values calculated in the FOURTH INTERVAL
436     # If the function is called with radius = outer radius, then add the
437     heat
438     # released to the 'Newton' array
439     for i in range(0, (N_t+1)-len(T_t_r)):
440         T_t_r.append(T_t_r_4[i])
441         Q_star_4.append(Q_over_Q_o_4[i]*Q_o_4)
442
443     Q = [Q_star_1[-1], Q_star_2[-1], Q_star_3[-1], Q_star_4[-1]]
444
445     return T_t_r, Q # Function
446     returns list with temperature as a function of radius and time.
447
448 r_crit = np.cbrt((0.2/np.pi)*(3/4)*V_p)
449     # Calculation of a 'critical radius'. This is the radius where
450     r_crit -> r_o accounts for 80% of the volume. (core -> r_crit = 20%
451     volume)
452
453 '''Temperature calculations'''
454 print('Time-of-flight for a particle with diameter = {} mm, given a tower
455 height of {} m is {:.2f} s.'.format(d_p, tower_height, t[t_index])) #
456 Writes TOF for a given dp and tower height to the console
457 print('The critical radius for a particle with diameter = {} is {:.2f} mm'.
458 format(d_p, r_crit*1000)) # Writes critical radius for a given dp to the
459 console
460 T_core = temp_calc(0.01*r_o)[0]

```

```

# Calls temp_calc func, using 0.01*outer radius as parameter.
450 T_surface = temp_calc(r_o)[0]
# Calls temp_calc func, using outer radius as parameter.
451 T_crit = temp_calc(r_crit)[0]
# Calls temp_calc func, using critical radius as parameter.
452
453
454 '''Heat release calculations'''
455 m_dot_prill = m_dot_slurry/m
456 Q = temp_calc(r_o)[1]
457 Q_released = sum(Q)*m_dot_prill*(1/1000)
458 print('Heat released by the prills: {:.2f} kJ/h'.format(float(Q_released)))
# Print the heat absorbed by the air in the console
459
460 #FOR TESTING AND SENSITIVITY PURPOSES, USE T_AIR_OUT as written below, ELSE
# USE T_AIR_OUT MEASUREMENTS
461 #T_air_out = Q_released/(m_dot_air*cp_air) + T_inf
# Calculated outgoing temperature [degC], used for model
# sensitivity
462 T_air_out_measured = 37.8
# Measured outgoing temperature [degC], used for model validation
463 T_air_out = T_air_out_measured
464 deltaT = T_air_out - T_inf
# Temperature difference of the air in the tower inlet and outlet
465 Q_absorbed = m_dot_air*cp_air/1000*deltaT
# Heat absorbed by the air, deltaT can be selected to be include
# measured or calculated outgoing air temperature
466 print('Heat absorbed by the air: {:.2f} kJ/h'.format(float(Q_absorbed)))
# Print the heat absorbed by the air in the console
467 err = ((Q_released-Q_absorbed)/Q_absorbed)*100
# Calculates the error between heat released by the prills and heat
# taken up by the air. THIS WILL BE 0% IF AIR TEMPERATURE OUT IS SET TO
# BE CALCULATED
468 print('This gives us an error of {:.2f}%'.format(float(err)))
# Print the error in the console
469 ''' Plotting and printing '''
470
471 fig = plt.figure(4)
472 plt.plot(t, T_core, label='Core', color='xkcd:orange red')
473 plt.plot(t, T_surface, label='Surface', color='xkcd:lightish blue')
474 plt.plot(t, T_crit, label='Critical radius', color='xkcd:barney')
475
476 plt.axhline(T_1, c='k', linestyle='dashed', label='$Cr_{main, start}$')
477 plt.axhline(T_2, c='k', linestyle='dotted', label='$Cr_{secondary, start}$'
)
478 plt.axhline(T_3, c='k', linestyle='dashdot', label='$Cr_{tertiary}$')
479
480 plt.xlim(0, t[t_index])
481 plt.xlabel('Time [s]')
482 plt.ylabel('Prill temperature [ $^{\circ}$ C]')
483 plt.legend()

```

```
484 plt.grid(True)
```

Listing E.1: Source code for composite ytterbø model. Model is shown with input parameters equal to those given in table 6.4

## E.2 Root program

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Feb 14 17:52:15 2022
4
5 @author: Anders
6 """
7
8 import math
9 import numpy as np
10 import pandas as pd
11 import os
12
13 #os.remove('roots.xlsx')
14
15 res = []
16
17 Bi = 0.5446836972029505
18 f=lambda x:(1-(x*(1/np.tan(x)))-Bi)
19
20 def rootsearch(f,a,b,dx):
21     x1 = a; f1 = f(a)
22     x2 = a + dx; f2 = f(x2)
23     while f1*f2 > 0.0:
24         if x1 >= b:
25             return None,None
26             x1 = x2; f1 = f2
27             x2 = x1 + dx; f2 = f(x2)
28     return x1,x2
29
30 def bisection(f,x1,x2,switch=0,epsilon=1.0e-9):
31     f1 = f(x1)
32     if f1 == 0.0:
33         return x1
34     f2 = f(x2)
35     if f2 == 0.0:
36         return x2
37     if f1*f2 > 0.0:
38         print('Root is not bracketed')
39         return None
40     n = int(math.ceil(math.log(abs(x2 - x1)/epsilon)/math.log(2.0)))
41     for i in range(n):
42         x3 = 0.5*(x1 + x2); f3 = f(x3)
43         if (switch == 1) and (abs(f3) > abs(f1)) and (abs(f3) > abs(f2)):
44             return None
```

```

45     if f3 == 0.0:
46         return x3
47     if f2*f3 < 0.0:
48         x1 = x3
49         f1 = f3
50     else:
51         x2 =x3
52         f2 = f3
53     return (x1 + x2)/2.0
54
55 def roots(f, a, b, eps=1e-6):
56     print ('The roots on the interval [%f, %f] are:' % (a,b))
57     while 1:
58         x1,x2 = rootsearch(f,a,b,eps)
59         if x1 != None:
60             a = x2
61             root = bisect(f,x1,x2,1)
62
63             if root != None:
64                 pass
65                 print (round(root,-int(math.log(eps, 10))))
66                 res.append(root)
67         else:
68             print ('\nDone')
69             break
70
71 roots(f, 0.5, 200)
72
73 df = pd.DataFrame(res)
74 df.to_excel('roots.xlsx')

```

Listing E.2: Program for finding multiple roots for an equation in a given interval