

Master Thesis

Effect of particle contact point treatment on the heat transfer in packed beds

carried out for the purpose of obtaining the academic degree

Diplom-Ingenieur (Dipl.-Ing.)

submitted at TU Wien

Faculty of Mechanical and Industrial Engineering

by

Mario Pichler

Matr.Nr.: 01226397

under the supervision of

Ass.Prof. Dipl.-Ing. Dr.techn. Michael Harasek,

MSc. Bahram Haddadi Sisakht

and

Dipl.-Ing. Christian Jordan

Institute of Chemical, Environmental & Bioscience Engineering
E166

Affidavit

I confirm, that going to press of this thesis needs the confirmation of the examination committee.

I, Mario Pichler, declare in lieu of oath

1. that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such.
2. that this work is original and has not been submitted elsewhere for any examination nor is it currently under consideration for a thesis elsewhere.

Vienna, 26.01.2018

Mario Pichler

Danksagung

Auch wenn es in den letzten Monaten immer wieder Rückschläge gegeben hat und die Probleme oft überwindbar schienen, war doch immer jemand da der mich unterstützt und motiviert hat. Durch euch war das Erstellen dieser Arbeit (meistens) eine große Freude.

Deshalb danke an euch alle.

Acknowledgement

Financial support was provided by the Austrian research funding association (FFG) under the scope of the COMET program within the research project Industrial Methods for Process Analytical Chemistry From Measurement Technologies to Information Systems (imPACts, www.k-pac.at) (contract # 843546).

The author gratefully acknowledges the funding support of K1-MET GmbH, metallurgical competence center. The research program of the competence center K1-MET is supported by COMET (Competence Center for Excellent Technologies), the Austrian program for competence centers. COMET is funded by the Federal Ministry for Transport, Innovation and Technology, the Federal Ministry for Science, Research and Economy, the province of Upper Austria, Tyrol, and Styria, the Styrian Business Promotion Agency.

The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC).

Abstract

In recent years computational fluid dynamic (CFD) simulation of fully resolved fixed beds has become a popular tool for getting deeper insight in local phenomena in packings. To get correct simulation results, one has to be especially careful on how to treat particle contact points. Different methods of particle point treatment have significant influence on the heat transfer and pressure drop in packed beds.

In this work different contact point treatment methods are tested and the heat transfer in packed beds is compared to experimental measurements and well established correlations. Using 'Design of Experiments' (DoE), a set of experiments is carefully selected, to study the influence of sphere diameter, contact point treatment (including the size and effective thermal conductivity of cylinders, so called bridges, which are inserted between particles), and the bed material on the heat transfer. Here only natural convection is considered, so that conduction is the main heat transfer mechanism. Moreover a wide variety of thermal conductivities (glass $k_{glass} = 1.129 \text{ W/(m K)}$, steel $k_{steel} = 54 \text{ W/(m K)}$, aluminium $k_{alu} = 236 \text{ W/(m K)}$) is studied.

Packed beds were created using the Discrete Element Method (DEM), were modified in the pre-/post-processing software SALOME if necessary and the heat transfer was then simulated using the open-source software OpenFOAM®. Simulation results were compared to experimental measurements, and a model was fitted to reduce the simulation error. Using this model the effective thermal conductivity for bridges between spherical particles can be calculated. Good results are achieved in the CFD-simulation of packed beds, consisting of monosized spheres, without forced convection and for a wide range of thermal conductivities when using the bridges method and the presented model. To validate the model, in addition to the DoE simulations, simulations for steel spheres and different sphere sizes were carried out.

Kurzfassung

In den letzten Jahren hat sich die CFD-Simulation von voll aufgelösten Festbettreaktoren als ein populäres Hilfsmittel zur genaueren Untersuchung von lokal Transportphänomenen in solchen Packungen erwiesen. Auf das Problem der Behandlung von Kontaktpunkten zwischen zwei Partikeln und bei Partikel-Wand-Punkten, dass bei der Gittererzeugung auftritt, ist besonders Acht zu geben, um korrekte Simulationsergebnisse zu erhalten. Die verschiedenen Möglichkeiten Kontaktpunkte zu behandeln haben Auswirkung auf den berechneten Druckverlust über die Füllkörperschüttung und die Temperaturverteilung im Bett.

In der vorliegenden Arbeit werden verschiedene Möglichkeiten der Kontaktpunktbehandlung bei der Gittererstellung von Festbetten, bestehend aus Kugeln, getestet. Der Wärmetransport im Festbett wird mit experimentellen Messungen und etablierten Korrelationen verglichen. Die mithilfe von Design of Experiments (DoE) sorgfältig ausgewählte Experimente und Simulationen wurden genutzt, um den Einfluss von wichtigen Bettparametern auf den Wärmetransport in der Packung zu untersuchen. Diese sind Kugelgrößen, Kontaktpunktbehandlung (im speziellen der Größe und effektiven Wärmeleitfähigkeit von Zylindern zwischen Kugeln, den Brücken oder 'bridges') und das Bettmaterial. Dabei wurde ein Augenmerk darauf gelegt, dass ausschließlich natürliche Konvektion auftritt, um den Wärmetransport nicht mit anderen Transportphänomenen (konvektivem Wärmetransport) zu überlagern. Weiters wurde eine große Spanne von Wärmeleitfähigkeiten, von Glas ($k_{glass} = 1.129 \text{ W/(m K)}$) bis hin zu Aluminium ($k_{alu} = 236 \text{ W/(m K)}$), untersucht.

Kugelpackungen wurden mit Hilfe der diskreten Elemente Methode (DEM) erzeugt, gegebenenfalls in der Pre-/Post-Processing software SALOME modifiziert und der Wärmetransport anschließend mit der open-source Software OpenFOAM® simuliert. Die Simulationsergebnisse wurden mit den Messungen verglichen und ein Modell zur Minimierung der Abweichung zwischen Simulation und Experiment erstellt. Gute Ergebnisse bei der CFD-Simulation des Wärmetransports in Kugelpackungen, bestehend aus gleich großen Kugeln, ohne erzwungene Konvektion, für eine große Spanne von Wärmeleitfähigkeiten und der Verwendung der Brücken-Methode ('bridges method') in Verbindung mit dem präsentierten Model, wurden damit erzielt. Zur Validierung des Modells wurden zusätzlich zu den DoE-Simulationen für Stahlkugeln und verschiedenen Kugelgrößen durchgeführt.

Contents

1	Introduction	1
1.1	Motivation and Problem Statement	1
2	Literature review	2
2.1	Packed Beds	2
2.1.1	Heat Transfer effects	2
2.1.2	Types of Heat Transfer Models	2
2.1.3	The Zehner/Bauer/Schlünder Model	2
2.1.4	Porosity	5
2.1.5	Transport Equation	7
2.1.6	Pressure Drop	10
2.2	Contact Point Treatment	12
3	Design of Experiments	16
4	Experiments	19
4.1	Calibration Experiments	19
4.1.1	Platinum Resistance Sensors	19
4.1.2	Infrared Camera	20
4.2	Heater Power	22
4.3	Chemical Composition of Steel Spheres	23
4.4	Glass Properties	24
4.4.1	Glass Density	24
4.4.2	Glass Heat Capacity	25
4.5	Sphere Diameter	28
4.6	Heat Transfer Experiments	29
5	Computational Fluid Dynamics	31
6	Methods, Results and Discussion	33
6.1	Creation of a Packed Bed	33
6.2	Meshing Procedure of snappyHexMesh	33
6.3	Discretization of the α_w -Model	34
6.4	The necks method	35
6.4.1	Single Region Approach	36
6.4.2	Multi Region Approach	38
6.4.3	Testing the Volumetric Heat Source	43
6.4.4	CFD-Simulation	47
6.4.5	The Necks Method: Conclusion	54
6.5	The bridges method	55
6.5.1	Influence of the Bridge Size on the Bed Voidage	57
6.5.2	Fractional Factorial Screening Design	59
6.5.3	Full Factorial Screening Design	63
6.5.4	Validation	70

7 Conclusion and Outlook	75
List of Symbols	79
List of Figures	85
List of Tables	86
References	87
Appendices	90
A Experimental Data	A-1
B Mesh Quality Settings	B-1
C Important CFD-solver Settings	C-1
D Simulation Data, DoE Results	D-1
E Validation Results	E-1
F SALOME-code for bridged sphere packing	F-1
G R-code for DoE	G-1
H α_w -Model implementation (python-code)	H-1

1 Introduction

1.1 Motivation and Problem Statement

In chemical process industry fixed-bed reactors are used for important chemical unit operations. They are used for different applications, mostly in separation processes, e. g. rectification, absorption or adsorption, but also in catalytic reactions. Mathematical models, consisting of coupled partial equations that describe momentum, energy and material balances together with equilibrium equations and transport rates, are required for designing and optimizing fixed-bed columns. Due to the complexity of solving such coupled stiff partial differential equation systems, simplifications are made, to reduce the effort of solving such systems. As a consequence lots of different models exist, that are only valid for certain operation conditions and in particular situations (*Shafeeyan, Daud, and Shamiri (2014)*). In most cases the simplified models do describe the overall process with sufficient accuracy.

However these conventional plug flow based or pseudo-continuum models do not consider actual structure of the fixed bed and therefore phenomena as channeling, local backflow or stagnation, radial heat transfer or local kinetics, caused by local concentrations and temperatures, are not considered. For catalytic reactions and adsorption processes, local phenomena, especially local temperatures, is one of the most important factors for correctly describe the process.

In recent years computational fluid dynamics (CFD) has become the method of choice to obtain more information of local phenomena in fixed beds. Most of the early works on the CFD-simulation of packed beds focused on a unit-cell approach or used representative parts of the packing. But as local temperatures and heat transfer in the bed are necessary to simulate adsorption processes or catalysed reactions, a mesh of the entire bed is required. There are some difficulties in generating a realistic bed, due to complexity of the packing, and particle-particle and particle-wall contacts. Automated meshing can lead to extremely skewed cells around contact points, which can result unstable simulations. Many authors studied different contact-point modifications, in order to avoid this problem and correctly predict pressure drop, some also included heat transfer in packed beds (e.g. *Dixon and Nijemeisland (2001)*, *Ookawara et al. (2007)*, *Dixon, Nijemeisland, and Stitt (2013)*, *Bu et al. (2014)*). The focus of all of this works have been columns with forced fluid flow, mostly radial heat transfer and particles with low thermal conductivities. In these cases the main effect of heat transfer is convection, the share of conduction in the packed bed is small.

The aim of this thesis is to study the heat transfer in packed beds for different contact point treatment methods and a wide range of thermal conductivities (glass with a thermal conductivity of $k_{glass} = 1.129 \text{ W/(m K)}$, aluminium with $k_{aluminium} = 236 \text{ W/(m K)}$) to give further insight and understandings of the influence of the tested contact point modifications. Simulations are validated by comparison to experimental measurements.

The resulting correlation, containing the influence of the identified parameter, is then validated using a different packing material and a range of particle sizes.

2 Literature review

2.1 Heat Transfer and Preassure Drop in Packed Beds

2.1.1 Heat Transfer effects

Heat transfer in a packed bed can be split in three contributing parts according to *Antwerpen, Toit, and Rousseau (2010)*. The first part is that of the conductivity of a packed bed with stagnant fluid (λ_{eff}). In this part we can distinguish between four different heat transfer mechanisms.

1. conduction through the solids,
2. conduction through contact area in between adjacent particles due to surface roughness,
3. conduction through the fluid and
4. radiation between solid surfaces.

The second part describes the additional thermal conductivity in the fluid due to turbulent mixing in the interstices of the packing ($\lambda_{f,eff}$).

The third part describes the additional heat transfer when fluid as well as the solid phase is in motion ($\lambda_{s,eff}$). The effective thermal conductivity of the bed is then described as:

$$\lambda_{bed} = \lambda_{eff} + \lambda_{f,eff} + \lambda_{s,eff} \quad (2.1)$$

Here, only the first contributing part is discussed.

2.1.2 Types of Heat Transfer Models

According to *Tsotsas and Martin (1987)* the numerous models for heat transfer in packed beds can be classified in three categories.

Type I. The Laplace equation for heat conduction is solved analytically or numerically

Type II. Thermal resistances are introduced for solid and fluid phase and combined in order to get a thermal conductivity of the bed

Type III. The thermal conductivity of a unit cell is calculated and then set equal to the thermal conductivity of the bed

After an evaluation of many different models *Tsotsas and Martin (1987)* concluded that **Type III** models, and especially the unit cell model developed by *Zehner, Bauer and Schlünder*, can be recommended for practical use.

2.1.3 The Zehner/Bauer/Schlünder Model

The unit cell in this model, for calculating λ_{eff} , consists of a cylindrical core, where two half spheres are placed, and of an fluid-filled section around the core ([Figure 1](#)).

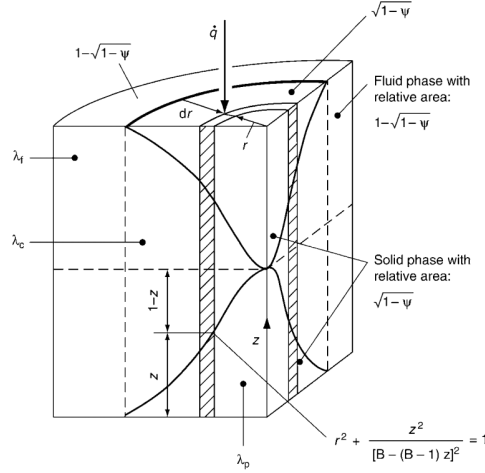


Figure 1: Unit cell of the Zehner/Bauer/Schlünder Model (*VDI-Wärmeatlas* (2013))

The reduced thermal conductivity $k_{bed} = \lambda_{eff}/\lambda_f$ can be calculated to

$$k_{bed} = (1 - \sqrt{1 - \Psi}) \Psi \left[(\Psi - 1 + k_G^{-1})^{-1} + k_{rad} \right] + \sqrt{1 - \Psi} [\varphi k_p + (1 - \varphi) k_c] \quad (2.2a)$$

with

$$k_c = \frac{2}{N} \left\{ \frac{B (k_p + k_{rad} - 1)}{N^2 k_G k_p} \ln \frac{k_p + k_{rad}}{B [1 + (1 - k_G) (k_p + k_{rad})]} \right\} + \frac{B + 1}{2B} \left[\frac{k_{rad}}{k_G} - B \left(1 + \frac{1 - k_G}{k_G} k_{rad} \right) \right] - \frac{B - 1}{N k_G} \quad (2.2b)$$

$$N = \frac{1}{k_G} \left(1 + \frac{k_{rad} - B k_G}{k_p} \right) - B \left(\frac{1}{k_G} - 1 \right) \left(1 + \frac{k_{rad}}{k_p} \right) \quad (2.2c)$$

$$B = C_f \left[\frac{(1 - \Psi)}{\Psi} \right]^{10/9} f(\xi) \quad (2.2d)$$

$$k_{rad} = \frac{\lambda_{rad}}{\lambda_f} = \frac{4 \sigma}{(2/\Psi) - 1} T^3 \frac{d}{\lambda_f} \quad (2.2e)$$

$$k_G = \frac{\lambda_G}{\lambda_f} = \left[1 + \left(\frac{l}{d} \right) \right]^{-1} \quad (2.2f)$$

$$l = 2 \frac{2 - \gamma}{\gamma} \sqrt{\frac{2 \pi R T}{M}} \frac{\lambda_f}{p (2 c_{p,g} - R/M)} \quad (2.3a)$$

$$\lg \left(\frac{1}{\gamma} - 1 \right) = 0,6 - \frac{1000 K/T + 1}{C} \quad (2.3b)$$

Equation 2.2b-f take into account all the primary and secondary parameters. The quantities λ_p , λ_f and Ψ , which are the thermal conductivity of the particles, the thermal conductivity of the fluid and the average bed voidage respectively, are called primary parameters. Sec-

ondary parameters are the heat transfer by radiation λ_{rad} , the pressure dependence k_G , heat transfer through flattened interparticle contacts (considered using a flattening coefficient ϕ), the particle shape C_f and the particle size d .

The influence of thermal radiation is taken into account by means of the quantity k_{rad} . *Chen and Chruchill (1963)* came to the conclusion that radiation is unimportant relative to conduction at temperatures below $1600^\circ F$ ($870^\circ C$).

The so called Smoluchowski effect (pressure dependence) is expressed by k_G . If the continuous phase is not gas, one may set $k_G = 1$. The modified mean free path of gas molecules l can be calculated according to [Equation 2.3a](#) using the accommodation coefficient γ , where T is the absolute temperature and C is dependent on the molar mass of the gas (He: $C=50$, H_2O : $C=3.6$, air: $C=2.8$).

The flattening coefficient φ can not be calculated with satisfying accuracy and has to be determined empirically.

The shape factor C_f describes the influence of the particle shape onto the thermal conductivity of the packed bed. This factor also has to be determined empirically. For orthorhombic beds made of equally large spheres this factor has a value of $C_f = 1.25$.

To capture the influence of different particle sizes in the bed a distribution function $f(\zeta)$ is foreseen in [Equation 2.2d](#). However, by setting the distribution function simply to $f(\zeta) = 1$ (polydispersed beds are treated as monodispersed beds), the thermal conductivity can be calculated quite accurately.

When using the model for polydispersed beds the particle diameter should be calculated from

$$d = \left(\sum \frac{Q_i}{d_i} \right)^{-1} \quad (2.4)$$

where Q_i is the volume fraction of the particles with diameter d_i .

By setting $k_{rad} = 0$, $k_G = 1$, $\varphi = 0$, $C_f = 1.25$ and $f(\zeta) = 1$, a simpler model, which is only considering primary effects, is obtained.

The reduced model (ZB-model) has been compared to a large amount of experimental results *Zehner and Schlünder (1970)* and *Tsotsas and Martin (1987)*. It was found that the model predicts the data with an accuracy of better than $\pm 30\%$ in almost all cases. The model represented in [Equation 2.2b-f](#) was also tested by *Bauer (1993)* and a good agreement between data and model was found. *Antwerpen, Toit, and Rousseau (2010)* concluded that the Zehner/Bauer/Schlünder-model (ZBS-model) gives reasonable results for the effective thermal conductivity in the bulk region of a randomly packed bed. The authors also point out that this unit cell model should be applied with caution in near-wall regions.

In the ZBS-model it is assumed that the heat transfer in a packed bed can be described with an effective thermal conductivity (homogeneous body). While this approach is successful practice and is reasonable for steady state problems, the assumption can not be applied when significant temperature differences occur between particles and fluid.

Also the flattening coefficient φ is a function of many undetermined factors, such as the elasticity of the material, external mechanical stress and surface roughness. Although some values of φ are suggested in *VDI-Wärmeatlas (2013)* (see [Table 1](#)), more detailed approaches should be used to take particle roughness into account, when the particle-particle heat transfer dominates. Although several other contact point models, taking into account

the surface roughness, external forces or the particle's Young modules, have been developed, the Zehner/Bauer/Schlünder-model gives the best results in the bulk region of a packed bed.

Table 1: Flattening coefficients for different particle shapes and materials (*VDI-Wärmeatlas* (2013))

Particle shape	φ	Material
Spherical	0.0013	Steel
	0.0077	Ceramic
	0.0253	Copper
Irregular	0.001	Sand

2.1.4 Porosity

One of the most important parameter contributing to the effective thermal conductivity of a packed bed is the porosity (also voidage, void fraction). Regular packings of monodispersed spheres have porosities between $\Psi_\infty = 0.259$ (cubic face centered or densest hexagonal arrangement) and $\Psi_\infty = 0.476$ (simple cubic packing). Infinitely extended random packed beds of monodispersed spheres are considered to have porosities between $\Psi_\infty = 0.36$ and 0.44. Four random packing modes for spheres are distinguished (*deKlerk* (2003)):

- Very loose random packing ($\Psi \approx 0.44$): obtained by gradual defluidization of a fluidized bed or by sedimentation.)
- Loose random packing ($\Psi \approx 0.40 - 0.41$): obtained by letting spheres roll individually in place, or by dropping the spheres into the container as a loose mass.
- Poured random packing ($\Psi \approx 0.375 - 0.391$): obtained by pouring spheres into a container.
- Dense random packing ($\Psi \approx 0.359 - 0.375$): obtained by vibrating or shaking down the packed bed.

According to *deKlerk* (2003), the average bed porosity is a function of the ratio of column to particle diameter $N = D/d$. The average void fraction increases with decreasing N . As *Zou and Yu* (1994) showed, not only $N = D/d$, but also the bed height to particle diameter ratio H/d , has an significant impact on the average voidage.

Random packed beds, consisting of equally sized spheres in tubes, show a steeply decrease of porosity with increasing distance from the wall, as shown in [Figure 2](#). A minimum is reached at approximately half of a particle diameter. At longer distances periodic oscillations are observed. These oscillations are damped and vanish after some particle diameters ($\approx 4 - 6 d$).

For calculating the void fraction in circular tubes filled with spheres of equal size, the equations proposed by *Bey and Eigenberger* (1997) ([Equation 2.5](#)) can be used.

$$r^* < 0 \Rightarrow \Psi(r) = \Psi_{min} + (1 - \Psi_{min}) r^{*2} \quad (2.5a)$$

$$r^* \geq 0 \Rightarrow \Psi(r) = \Psi_\infty + (\Psi_{min} - \Psi_\infty) \exp\left(\frac{-r^*}{a_1}\right) \cos\left(\frac{\pi r^*}{b_1}\right) \quad (2.5b)$$

with

$$r^* = \left[\frac{R-r}{x_{min}} \right] - 1 \quad (2.5c)$$

$$x_{min} = 0.5 d \left(\frac{D}{d} - \sqrt{\left(\frac{D}{d} - 1 \right)^2 - 1} \right) \quad (2.5d)$$

and

$$\Psi_{min} = 0.24, a_1 = 4, b_1 = 0.876$$

Here the constant a_1 is set to 4 (as proposed in *VDI-Wärmeatlas* (2013)), while *Bey* and *Eigenberger* (1997) used values of $a_1 = 10$. Ψ_∞ can be obtained by adjusting it, so that the integrated profile of Equation 2.5 fits to the value Ψ_k obtained by an empirical correlation for the voidage of the whole cross section. *Bey* and *Eigenberger* (1997) suggests the correlation of Jeshar to calculate the avarage voidage (Equation 2.6; *Brauer* (1971)). A different model, optimized for small bed to particle diameter values N ($D/d = 2-20$), is presented by *deKlerk* (2003) (Equation 2.7).

$$\Psi_k = 0.375 + 0.34 \frac{d}{D} \quad (2.6)$$

$$\Psi_k = \Psi_\infty + 0.35 \exp \left(-0.39 \frac{D}{d} \right) \quad (2.7)$$

Furthermore *deKlerk* (2003) developed a new radial bed porosity model (Equation 2.8a-b) for equally sized spheres and small bed to particle diameter ratios ($D/d = 2-20$). The aim of his work was to decrease the relative estimation error compared to other correlations. This model is able to reduce the relative estimated error from 7-12% down to 4%, compared to other established correlations.

$$z \leq 0.637 \Rightarrow \Psi(r) = 2.14z^2 - 2.53z + 1 \quad (2.8a)$$

$$z \geq 0.637 \Rightarrow \Psi(r) = \Psi_\infty + 0.29 \exp(-0.6z) [\cos(2.3\pi(z - 0.16))] + 0.15 \exp(-0.9z) \quad (2.8b)$$

Where

$$z = \frac{R-r}{d} \quad (2.9)$$

is the nondimensional distance from the wall. When omitting the oscillations the empirical correlation of *Giese* (1998) can be used.

$$\Psi(r) = \Psi_\infty \left(1 + a_2 \exp \left[-b_2 \frac{R-r}{d} \right] \right) \quad (2.10a)$$

with

$$\Psi_\infty = 0.40, a_2 = 1.36, b_2 = 5$$

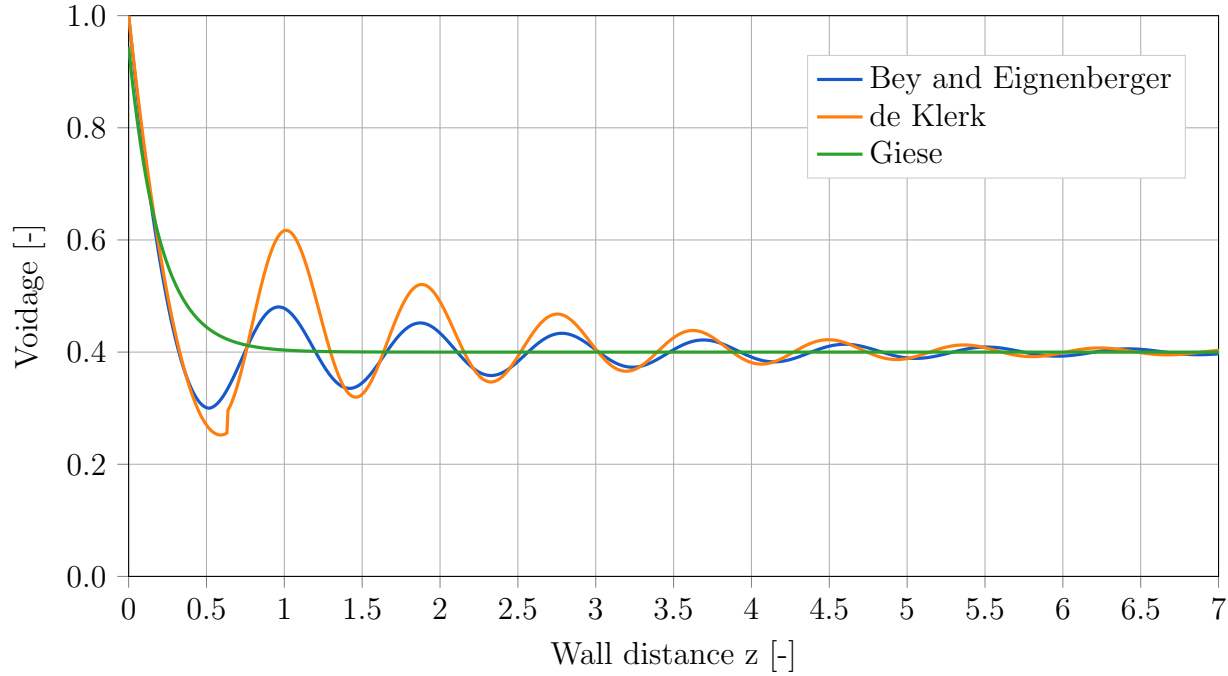


Figure 2: Bed porosities using Equation 2.5, Equation 2.8 and Equation 2.10 with $\Psi_\infty = 0.40$. z is the nondimensional distance from the wall, $z = (R-r)/d$. The unsteadiness of the model according to *deKlerk* (2003) at $z=0.637$ nearly disappears for $\Psi_\infty = 0.362$.

2.1.5 Transport Equation: The $\Lambda_r(r)$ -Model and the α_w -Model

A widely used way to describe the heat or mass transfer in packed beds with or without fluid flow is to use quasi-homogeneous models. In this kind of models the distinction between fluid and solid phase is abandoned by regarding state variables as local average between phases (*VDI-Wärmeatlas* (2013)). The heat transport in a packed circular tube can be described by

$$\{[1 - \Psi(r)] \rho_p c_p + \Psi(r) \rho_f c_f\} \frac{\partial \vartheta}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left[\Lambda_r(r) r \frac{\partial \vartheta}{\partial r} \right] + \Lambda_{ax}(r) \frac{\partial^2 \vartheta}{\partial z^2} - u_0(r) \rho_f c_f \frac{\partial \vartheta}{\partial z} \quad (2.11)$$

according to the $\Lambda_r(r)$ -model. Here ϑ is the temperature, t is the time, u_0 is the fluid velocity, ρ_f , ρ_p , c_f and c_p are the fluids and solids material parameter, $\Psi(r)$ is the radial porosity profile, $\Lambda_{ax}(r)$ and $\Lambda_r(r)$ are the axial and radial effective thermal conductivities, r is the radial coordinate and z is the axial coordinate.

When neglecting the radial variance of the porosity, the axial velocity and the effective thermal conductivity, Equation 2.11 transforms to the equation of the α_w -model.

$$\{[1 - \Psi] \rho_p c_p + \Psi \rho_f c_f\} \frac{\partial \vartheta}{\partial t} = \Lambda_r \left[\frac{\partial^2 \vartheta}{\partial r^2} + \frac{1}{r} \frac{\partial \vartheta}{\partial r} \right] + \Lambda_{ax} \frac{\partial^2 \vartheta}{\partial z^2} - u_0 \rho_f c_f \frac{\partial \vartheta}{\partial z} \quad (2.12)$$

The voidage Ψ can be calculated using one of the equations given in chapter 2.1.4. The axial

effective thermal conductivity Λ_{ax} can be calculated according to [Equation 2.13](#)

$$\frac{\Lambda_{ax}}{\lambda_f} = \frac{\lambda_{bed}}{\lambda_f} + \frac{Pe_0}{K_{ax}} \quad (2.13)$$

with

$$K_{ax} = 2 \quad (2.14)$$

and

$$Pe_0 = \frac{u_0 \rho_f c_f d}{\lambda_f} \quad (2.15)$$

In this work, for comparison to CFD-simulations, only axial heat transfer is considered in the α_w -Model. So [Equation 2.12](#) simplifies to

$$\{[1 - \Psi] \rho_p c_p + \Psi \rho_f c_f\} \frac{\partial \vartheta}{\partial t} = \Lambda_{ax} \frac{\partial^2 \vartheta}{\partial z^2} \quad (2.16)$$

with $\Lambda_{ax} = \lambda_{bed}$.

Boundary conditions:

$$z = 0 : T = T(t) \quad (2.17a)$$

$$z = x \cdot z_{max} : \frac{\partial T}{\partial z} = 0 \quad (2.17b)$$

When using boundary conditions shown in [Equation 2.17](#), namely a time dependent temperature boundary condition at $z = 0$ and a zero gradient boundary condition at $z = x z_{max}$, [Equation 2.16](#) can easily be solved numerically. The latter does not reflect reality, in a sense that there is heat transfer from the top of the packing to the surrounding. To take this into account a height-factor $x > 0$ is introduced. By changing the numeric value of x , the temperature evolution at the top of the packing at z_{max} can be adjusted to fit experimental data.

2.1.5.1 Drawbacks of Transport Equations

As stated in [Section 2.1.1](#), the effective thermal conductivity λ_{eff} is not the only contributing heat transfer mechanism in a packed bed. According to *VDI-Wärmeatlas* (2013), the relative importance of axial dispersion of heat λ_{eff} , the fluid-to-particle heat transfer $\lambda_{f,eff}$ and the intra-particle heat conduction $\lambda_{s,eff}$ changes with the molecular Péclet number Pe_0 . Often a linear combination of the three parts provides a good approximation of the axial dispersion of heat ([Equation 2.18a-d](#)). When plotting the three contributing parts over the Péclet number Pe_0 , as shown in [Figure 3](#), three regions can be identified. In each region one heat transfer mechanism is dominant. At the intersections points of the three curves two critical Péclet numbers $Pe_{0,cr,1}$ and $Pe_{0,cr,2}$ can be identified. Only for small Péclet numbers, $Pe_0 < Pe_{0,cr,1}$, the axial dispersion of heat is due to the effective thermal conductivity Λ_{ax} alone and thus a quasi homogeneous model is sufficient. For higher Péclet numbers, $Pe_0 > Pe_{0,cr,1}$, heterogeneous models should be used.

In heterogeneous models separate mass and heat balances for solid and fluid phase are used. This leads to the problem of additional transport parameters (effective thermal con-

ductivity for each phase), which are not accurately known. Also the interconnection between heat and momentum balance is an issue.

Since in this study only the quasi homogeneous Zehner/Bauer/Schlünder model is used for comparison with experimental and (CFD-)simulation data, an underprediction of Λ_{ax} is expected. This is because natural convection of the gas is not considered, but it is occurring in the experiments and is also considered in the simulations.

$$A = A_1 + A_2 + A_3 \quad (2.18a)$$

with

$$A_1 = \frac{(1 + K^*)^2}{Pe_0} \frac{\Lambda_{ax}}{\lambda_f} \quad (2.18b)$$

$$A_2 = \frac{K^{*2} Pe_0}{Nu A_v d} \quad (2.18c)$$

$$A_3 = \frac{K^{*2} Pe_0}{60(1 - \Psi) \lambda_p / \lambda_f} \quad (2.18d)$$

In Equation 2.18a-d

$$K^* = \frac{[(1 - \Psi) \rho_p c_p]}{[\Psi \rho_f c_f]} \quad (2.19)$$

is the thermal capacity ratio and

$$Nu = \frac{\alpha d}{\lambda_f} \quad (2.20)$$

is the Nusselt number for heat transfer between particle and fluid. The surface area of the packing per unit volume is denoted by A_v (*VDI-Wärmeatlas* (2013)).

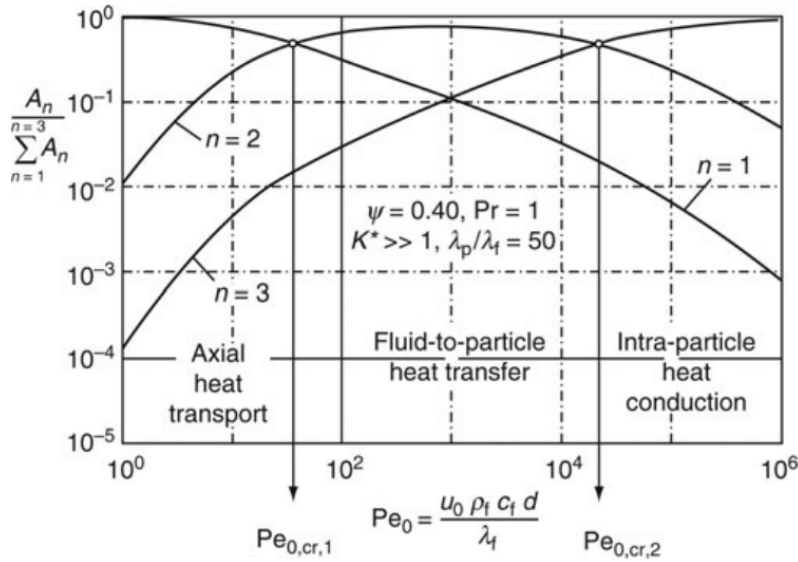


Figure 3: Relative influence of axial heat transport ($n=1$), fluid-to-particle heat transfer ($n=2$), and intraparticle heat conduction ($n=3$) on the experimentally observable axial dispersion. (*VDI-Wärmeatlas* (2013))

2.1.6 Pressure Drop

Over the last century many different correlations to calculate the pressure drop in packed beds have been developed and refined. Each new equation is claimed to be especially good in predicting the pressure drop under certain conditions or in general superior to previously published equations. Nevertheless the correlations published by *Carman* (1937) and *Ergun* (1952) are two of the most popular ones. According to *Erdim, Akgiray, and Demir* (2015) the Ergun equation can be written as

$$\frac{-\Delta P}{L} = 150\mu \frac{(1-\Psi)^2}{\Psi^3} \frac{u}{d^2} + 1.75\rho_f \frac{(1-\Psi)}{\Psi^3} \frac{u^2}{d} \quad (2.21)$$

for incompressible flow through a bed of spherical particles of identical size. Where $-\Delta P$ is the piezometric pressure drop in the bed, L is the depth of the bed, u is the superficial velocity, Ψ is the voidage of the bed, d is the particle diameter, ρ_f is the fluid density and μ is the dynamic fluid viscosity.

By introducing the 'dimensionless pressure drop' (also 'modified particle friction factor') f_P

$$f_P = \frac{\Delta P d}{\rho_f u^2 L} \quad (2.22)$$

Equation 2.21 can be expressed as:

$$f_P = \left(150 + 1.75 \left(\frac{Re}{1-\Psi} \right) \right) \frac{(1-\Psi)^2}{Re \Psi^3} \quad (2.23)$$

Using the dimensionless pressure drop the Carman-equation can be written as follows:

$$f_P = \left(180 + 2.871 \left(\frac{Re}{1-\Psi} \right)^{0.9} \right) \frac{(1-\Psi)^2}{Re \Psi^3} \quad (2.24)$$

Erdim, Akgiray, and Demir (2015) carried out a total number of 39 pressure drop experiments using a range of particle diameter d , porosities Ψ , Reynolds numbers Re and column to particle diameter ratios D/d (see Table 2) and compared the experimental result to 38 different models for the pressure drop caused by fluid flow through packed beds.

Table 2: Properties of particles and bed and Reynolds numbers used in *Erdim, Akgiray, and Demir (2015)*

d, mm	$\rho_P, g/cm^3$	Ψ	Re_m range	D/d
1.18	2.479	0.377, 0.386, 0.406	2-130	34.1
1.99	2.537	0.377, 0.381, 0.394, 0.408, 0.413	10-382	20.1
3.18	2.494	0.385, 0.389, 0.396, 0.408, 0.413	21-900	12.6
4.03	2.499	0.389, 0.391, 0.417, 0.25	31-1020	9.9
4.98	2.529	0.401, 0.436, 0.438, 0.440	66-1454	8.1
6.03	2.556	0.408, 0.443, 0.445	139-1834	6.7
7.15	2.521	0.422, 0.448, 0.456	158-2304	5.6
8.03	2.571	0.424, 0.462	184-2777	5.0
9.99	2.568	0.444, 0.460, 0.470	350-3582	4.0

The Fahien-Schriver (Equation 2.25) correlation has been found to give the best overall accuracy with 4.42% mean error. Furthermore this equation always predicted the pressure drop with an accuracy of $\pm 10\%$.

$$f_P = \left(q \frac{f_{1L}}{Re_m} + (1 - q) \left(f_2 + \frac{f_{1T}}{Re_m} \right) \right) \frac{1 - \Psi}{\Psi^3} \quad (2.25a)$$

with

$$q = \exp \left(- \frac{\Psi^2(1 - \Psi)}{12.6} Re_m \right) \quad (2.25b)$$

$$f_{1L} = \frac{136}{(1 - \Psi)^{0.38}} \quad (2.25c)$$

$$f_{1T} = \frac{29}{(1 - \Psi)^{1.45} \Psi^2} \quad (2.25d)$$

and

$$f_2 = \frac{1.87 \Psi^{0.75}}{(1 - \Psi)^{0.26}} \quad (2.25e)$$

The Carman-equation shows the smallest mean error for Reynolds numbers $Re_m > 300$, with a mean error of 3.08%. For Reynolds numbers of $Re_m < 300$, the Carman-equation shows a mean error of 8.71%, compared to the slightly lower mean error of 5.13% in the prediction of the pressure drop of the Ergun-equation. It should be said that the Ergun-equation should not be used for $Re_m > 500$.

Finally the authors presented a simple new equation derived by using their experimental data (Equation 2.26), which, unsurprisingly, showed the smallest mean error.

$$f_P = \left(160 + 2.81 \left(\frac{Re}{1 - \Psi} \right)^{0.904} \right) \frac{(1 - \Psi)^2}{Re \Psi^3} \quad (2.26)$$

2.2 Contact Point Treatment

One of the big challenges in simulating packed beds is to find an appropriate way to handle the particle-particle and particle-wall contact points in meshing for CFD-simulation. At these points the number of cells is the limiting factor. When reducing the cell size in the close proximity of the contact points, one would get a better approximation of reality, but the number of cells would increase dramatically.

There are different ways to deal with contact points in packed beds. Two global modifications and three local modifications are shown in Figure 4. All this methods and their influence on the heat transfer and pressure drop have been studied by several authors.

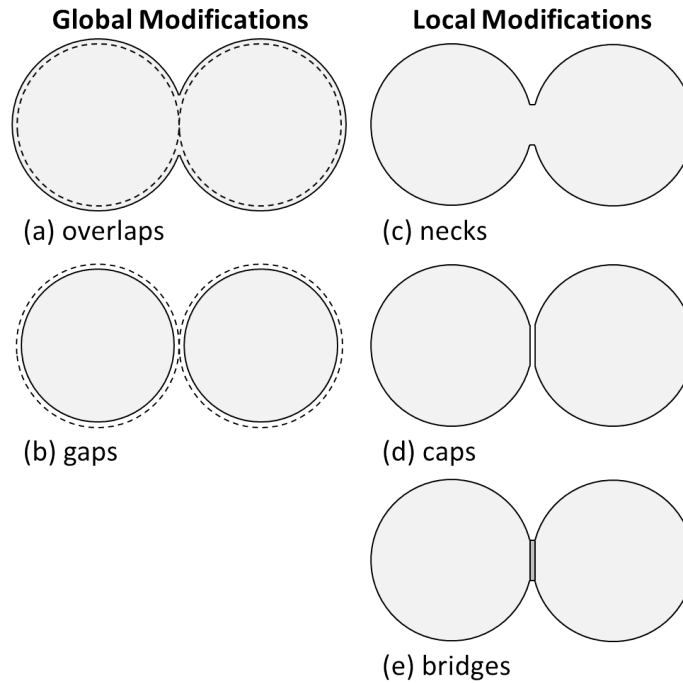


Figure 4: Different global and local contact point modification methods

To avoid contact points, many authors used global methods to treat contact points. Two possible global modifications are the so called gaps method and the overlaps method (wording was introduced by *Dixon et al. (2011)*). Here the particle diameter is decreased or increased to get gaps between particles or overlapping particles respectively. When using one of this methods bad cells at contact points can be avoided.

Dixon and Nijemeisland (2001) used CFD simulation to compute the velocity field in tubular fixed-bed reactors with small N ($N=2$, 44-sphere and $N=4$ with axial periodic boundary). Here, because of this small particle diameter to reactor diameter ratio, a structured packed bed with non touching spheres was used.

Nijemeisland and Dixon (2001) used the "near miss model" (later referred as gaps model) with different scaling factors to simulate not only the velocity field, but also heat transfer for 44-spheres and $N=2$. By comparison with experiments they concluded that the gaps model does not correctly predict the radial temperature profile. By taking into account different

sources of errors of the experiments and the CFD-simulation, the results can be corrected to get a much better agreement.

Tabib, Johansen, and Amini (2013) also used this method to calculate the pressure drop, velocity and temperature profiles of cut segments of packed beds consisting of cylindrical particles and fluted ring pellets. The goal was to find representative cut segments of packed beds that can serve as good representative for CFD simulation. The authors used the simulation results to calculate parameters of different empirical pressure drop and heat transfer correlations and compared them to the original parameters. They state that the values obtained in their work differ substantially from the original parameters. Thus the validity of the global modification is questionable.

Behnam et al. (2013) also omitted particle-particle heat transfer by shrinking spherical particle by 1%. By averaging the obtained radial and axial velocity profiles, the authors calculated the effective thermal conductivity at each radial position by pointwise application of the Zehner-Bauer equation. They claim that this is a good approach to predict radial heat transfer in packed beds. But as a global method of treating contact points, with this approach the pressure drop could not be predicted with reasonable accuracy.

The opposite approach is used by *Guardo et al. (2004)*, who expanded the spheres so that they are overlapping by 1% of their diameter. Here the simulated pressure drop is slightly higher compared to Ergun equation, but shows good agreement for particle Reynolds numbers of up to 1000. The effective radial conductivity also is overpredicted in the CFD-simulation compared to accepted correlations (*Yagi and Wakao (1959)*, *Yagi and Kunii (1960)*). It is worth to mention that in this study a four-layer arrangement of 44 sphere was used. For such small packings, the contact point treatment might not have a big influence on the pressure drop.

Other authors as well applied the overlaps method to avoid the contact point problem. However this global modifications have big impact on the overall packing structure, more precisely the voidage. An change in diameter of 1% lead to approx. 10-15% error of the drag coefficient (*Dixon, Nijemeisland, and Stitt (2013)*).

A different approach was introduced by *Ookawara et al. (2007)* and *Kuroki et al. (2007)*. They used the bridges method to calculate the pressure drop and heat transfer in packed beds. They linked particles by placing cylinders in between adjacent spheres.

Eppinger, Seidler, and Kraume (2011) moved the vertices at close proximity of contact points towards particle centres to get small caps between particles caps method. The bridges method and the caps method are both local modifications.

Dixon, Nijemeisland, and Stitt (2013) tested the different local and global contact point modification methods for spherical particles. They concluded that global methods lead to unacceptable high errors in pressure drop and heat transfer. In order to get a reasonable pressure drop, caps or bridges with $d/D \leq 0.2$ should be used. For heat transfer bridges with $d/D \leq 0.2$, for both particle-particle and particle-wall contacts, with an effective thermal conductivity can be used for $Re_p \leq 2000$. For $Re_p \geq 2000$ the bridge size should be restricted to $d/D \leq 0.1$.

Bu et al. (2014) studied the influence of gaps, overlaps, bridges and caps as contact point treatment on the pressure drop and heat transfer in structured beds and compared the simulated results to experiments carried out by *Yang et al. (2012)*. They found out that gaps

and overlaps lead to remarkable changes in porosity and therefore in pressure drop and are not suitable for pressure drop simulations. The bridges method gave the most reasonable pressure drops. Thus simulations using bridge sizes of $d/D = 0.1$, $d/D = 0.16$, $d/D = 0.2$ and $d/D = 0.24$ were carried out to study the influence of different bridge sizes on the pressure drop. The average deviations compared to experimental results were found to be 67.5%, 32.5%, 16.7% and 17.1% respectively. The authors concluded that bridges with $d/D = 0.16$ up to $d/D = 0.2$ should be used in structured packed beds.

Wehinger (2016) showed that the difference in porosity for the caps and the bridges method used for cylindrical particles is less than 1% (absolute). The pressure drop differs by approximately 15% from the lowest value (caps) to the highest value (bridges). Different contact point treatments have been used to simulate the heat transfer for cylindrical particles. Bridges at heater-particle contacts, with the same conductivity as the particles, gave the highest temperatures. Bridges with different conductivities and the local caps method gave close results. However these results have not been compared to experimental data. The author concluded that for high Reynolds numbers the influence of the contact-point treatment is getting less important for radial heat transfer, as convective heat transfer is dominant. Furthermore he pointed out that the thermal conductivity of the bridges can be used as a tuning parameter to get correct heat transfer.

Summing up, it is obligatory to insert cylindrical bridges with an effective thermal conductivity according to Dixon, Nijemeisland, and Stitt (2013) at contact points, to get reasonable results for axial heat transfer and pressure drop in packed beds. The thermal conductivity of the cylindrical bridges can be calculated to

$$k_{eff} = \frac{2h_b}{r_b^2} \int_0^{r_b} \frac{k_p k_{fr}}{h_i k_{fr} + h k_p} r dr \quad (2.27a)$$

$$h_b = R - \sqrt{R^2 - r_b^2} \quad (2.27b)$$

$$h = R - \sqrt{R^2 - r^2} \quad (2.27c)$$

$$h_b = h + h_i \quad (2.27d)$$

$$k_{fr} = \frac{k_f}{1 + (2\Lambda/h)(2 - \gamma)/\gamma} \quad (2.27e)$$

where h is the height of the bridge, k_p is the thermal conductivity of the particles, k_f is the thermal conductivity of air, γ is the accommodation coefficient and Λ is the mean free path of the gas molecules. Geometrical variables are defined in Figure 5. Equation 2.27 takes into account the Smoluchowski effect (Equation 2.27 (e), see Section 2.1.3). γ and Λ are calculated according to Equation 2.3 (b) and Equation 2.3 (a) respectively. Dixon, Nijemeisland, and Stitt (2013) used fixed values for γ ($= 0.1$) and Λ ($= 0.07 \cdot 10^{-6} \text{ m}$). Especially the accommodation coefficient of $\gamma = 0.1$ does not agree with the accommodation coefficient calculated according to Equation 2.3 (b) ($\gamma = 0.899$ for $T = 300 \text{ K}$).

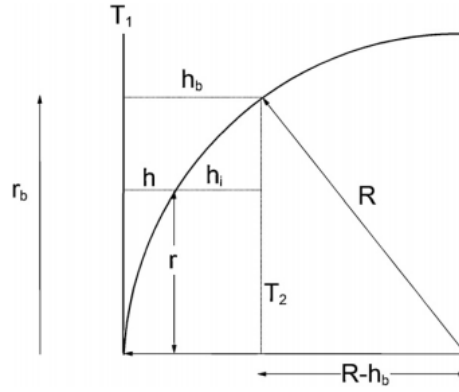


Figure 5: Geometry of the bridge and definition of the variables used in Equation 2.27 (*Dixon, Nijemeisland, and Stitt (2013).*)

However all these studies, which did investigations on the CFD-simulation of heat transfer in packed beds, used particles with low thermal conductivity:

- *Dixon, Nijemeisland, and Stitt (2013)*
 - (Alumina): $k_p = 1 \text{ W/(mK)}$, $\rho_p = 1947 \text{ kg/m}^3$, $c_p = 1000 \text{ J/(kgK)}$
- *Dixon et al. (2012)*
 - Alumina: $k_p = 0.25 \text{ W/(mK)}$, $\rho_p = 2000 \text{ kg/m}^3$, $c_p = 1000 \text{ J/(kgK)}$
 - Nylon: $k_p = 0.4 \text{ W/(mK)}$, $\rho_p = 1140 \text{ kg/m}^3$, $c_p = 1700 \text{ J/(kgK)}$
- *Wehinger (2016)*
 - $k_p = 5 \text{ W/(mK)}$, $\rho_p = 1300 \text{ kg/m}^3$, $c_p = 1000 \text{ J/(kgK)}$

Moreover only setups with fluid flow have been investigated. Here convection will dominate and conduction only plays a secondary role. To be able to give more general recommendations, regarding contact-point modifications, for a wider range of materials and other flow regimes, additional research is required. In this study Design of Experiments (DoE) is used to find the optimal contact point treatment for packed beds consisting of monodisperse spheres, without forced convection. This will be discussed briefly in Section 3.

3 Introduction to Design of Experiments (DoE)

Design of Experiments provides a plan for collecting data from a set of carefully selected experiments to discover unknown effects, to test or establish a hypothesis, or to illustrate known effects. In this set of experiments a number of factors is varied and their effects on the response(s) are studied. By using DoE the number of experiments that are necessary to achieve a certain goal (screening, optimization), and therefore invested time and design costs, can be reduced compared to the "intuitive" COST (Change only One Separate factor at a Time) approach.

Before one can carry out the experiments some basic questions have to be answered.

1. What are the experimental objectives?
2. Which factors might influence the output (response) of the experiments?
3. How big is the experimental area that has to be covered? In other words: In what range should the factors be varied?
4. What are the responses of the system/process?

The experimental objectives

Two of the main objectives in Design of Experiments are screening and optimization.

Screening is used at the beginning of the experimental procedure. The objective is to explore many factors in order to reveal whether they have an significant influence on the response and to identify their appropriate ranges. Prior screening the examined factors are ascribed the same chance of influencing the response, whereas after the design has been performed and evaluated, only a few important factors remain.

Optimization is used after Screening. The objectives are to predict the response values for all possible combinations of factors within the experimental region, and to identify an optimal experimental point. However, when several responses are treated at the same time, it is usually difficult to identify a single experimental point at which the goals for all responses are fulfilled. Therefore the final result often reflects a compromise between partially conflicting goals (*Eriksson et al. (2008)*). In summary, with an optimization design one can extract detailed information regarding how different combinations of factors influence the response.

The process of Design of Experiments is illustrated in [Figure 6](#).

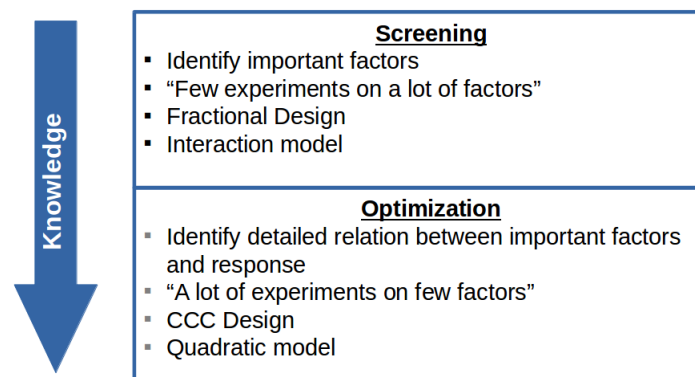


Figure 6: Summary of screening and optimization

The factors

The factors are the variables which, due to changes in their level, will exert an influence on the response of the system or the process. Factors can be divided in controllable and uncontrollable and/or quantitative and qualitative factors. After defining the factors, their range needs to be specified. The range might be limited by the experimental settings or objectives. For example the temperature of liquid water at ambient pressure may be varied between 0 °C and 100 °C, but in some cases it might only be relevant to consider the temperature range 30-40 °C.

The responses

It is important to select responses that are relevant according to the problem formulation. Many responses are often necessary to map the performance characteristics of a process well. It is not a problem to handle many responses at the same time.

The design

Depending on the experimental objectives, one may choose different designs, which represent the levels and combinations of the factors used in all the experiments. For screening usually fractional factorial designs are sufficient, while full factorial designs or central composite designs are needed for optimization. The graphical representation of different commonly used designs are shown in Figure 7.

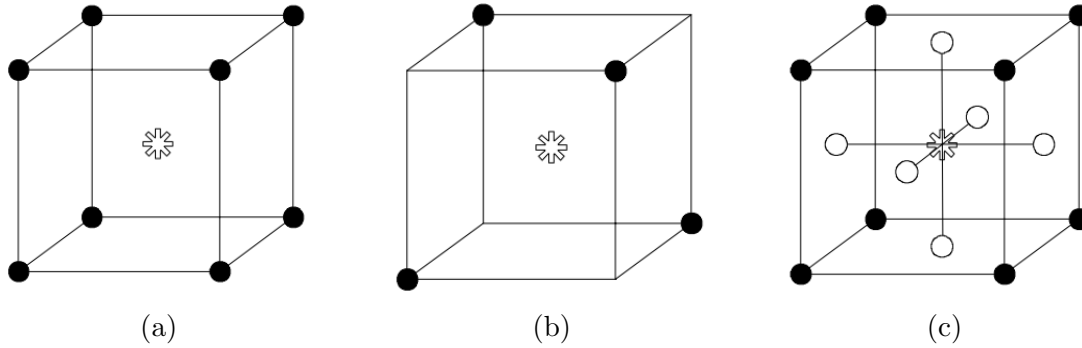


Figure 7: Example of common three factor designs used for screening or optimization. Each of the circles represents the levels of the factors for a single experiment. (a) Full Factorial Design, (b) Fractional Factorial Design, (c) Central Composite Face-Centered Design (CCF). (Eriksson et al. (2008))

The model

Three main types of polynomial models, which are frequently used, are distinguished in DoE to be fitted to the experimental data. These are *linear*, *interaction*, and *quadratic* models. Here y , β_i , β_{ij} , x_i and ϵ are the response, main effects coefficients, interaction coefficients, factors and the residual, respectively.

Linear

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \dots + \epsilon \quad (3.28)$$

Interaction

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + x_1x_2\beta_{12} + \dots + \epsilon \quad (3.29)$$

Quadratic

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + x_1^2\beta_{11} + x_2^2\beta_{22} + x_1x_2\beta_{12} + \dots + \epsilon \quad (3.30)$$

Being the most complex model, a quadratic polynomial requires more experiments than the others. The choice of which model to use is not completely free. If screening was selected as objective, either a linear or an interaction model is pertinent. The interaction model is highly recommended for screening. If optimization is the objective of your DoE, only a quadratic model will do.

Moreover, one may occasionally end in a situation where a cubic model is necessary. This is especially the case when modelling the performance of living systems, e.g. growth rate of algae. However, it should be made clear that cubic models are rarely relevant in industrial practice, because they are too complex and demand too many experiments.

In Design of Experiments coded factor levels are used instead of the actual levels (e.g. '-' or '-1' and '+' or '+1' instead of 30 °C and 40 °C, respectively). The coded factor levels are used as the independent variables in order to calculate the regression coefficients. In consequence to this, the factors $x_1 \dots x_n$ have to be set to the coded factor levels (-1, +1, 0, -1.414, ...) when using the model to predict a response.

The constant term x_0 always relates to the estimated response value at the design center point, that is, when the factors $x_1 \dots x_n$ have the value zero in the *coded units*. (*Eriksson et al. (2008)*)

4 Experiments

In this study different contact point modifications are simulated and compared to experimental results. In a simple experimental setup, the surface temperature of a packed bed was measured with a platinum resistance sensor (Pt1000) and an infrared camera, while heating the packing from below using two PTC (positive thermal coefficient of resistance) heaters, with a nominal power of 20 W each. In contrast to validation experiments of other authors, in this study only natural convection is considered, to avoid convective heat transfer as dominant factor. The experimental setup is shown in [Figure 8](#).

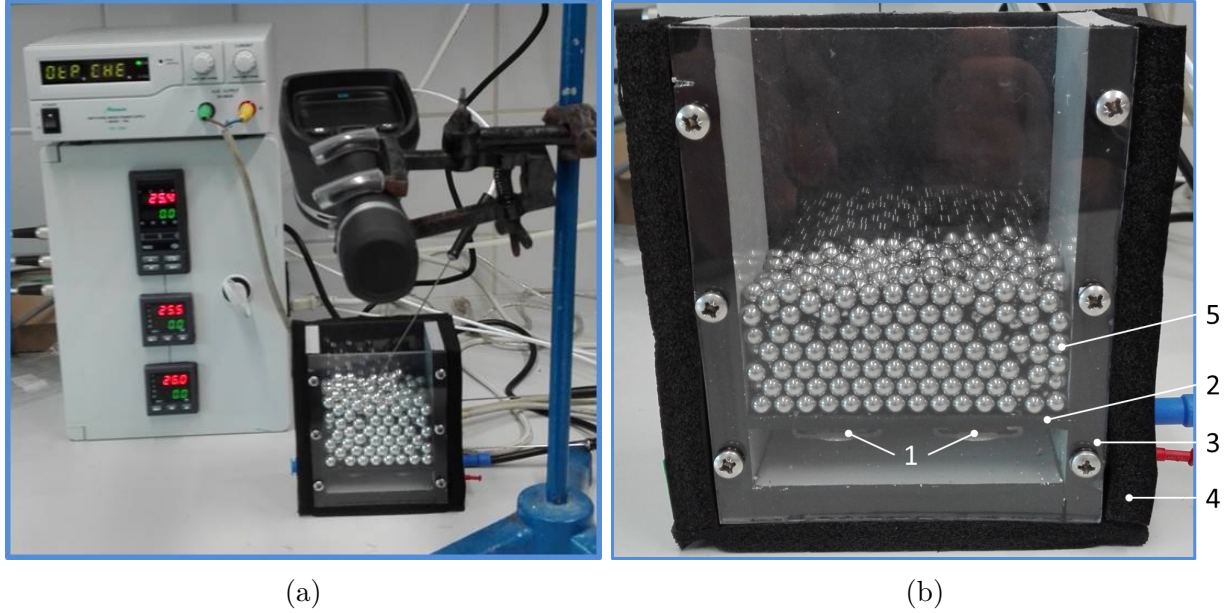


Figure 8: Experimental setup. (a) Setup including platinum resistance sensor and infrared camera (b) 1...heating elements, 2...aluminium bottom plate, 3...plastic container, 4...isolation, 5...sphere packing

4.1 Calibration Experiments

4.1.1 Platinum Resistance Sensors

To get accurate results for the surface temperature of the packing, platinum resistance sensors (Pt1000) have been used. For calibration a simple two point calibration was used. The temperature of boiling water and ice water was measured with all available Pt1000. Calibration experiments were carried out in a beaker under continuous stirring on a magnetic stirrer. To avoid errors it was ensured that the sensors did not touch any wall. To be able to correct all subsequent measurements, a linear regression line was inserted between the two measurement points. Results can be seen in [Table 3](#) and [Equation 4.31](#).

Table 3: Comparison of theoretical and measured temperature of ice water and boiling water.

	theoretical T, °C	measured T, °C
ice water	0	1
boiling water	100	99.5

$$T_{corr} = T_{measured} * 1.0152 - 1.0152 \quad (4.31)$$

4.1.2 Infrared Camera

In addition to the platinum resistance sensor (Pt1000) a FLIR® E6 infrared camera was used to get not only the surface temperature of the packing at a single point, but also the temperature distribution. Since an emission coefficient ϵ needs to be set when using a infrared camera, it had to be determined. To avoid different emission coefficients for all the different particles, it was decided to colour the top layer of spheres black. Therefore matt black high temperature paint was used. The emission coefficient ϵ was determined using a simple experimental procedure. The setup is shown in [Figure 9](#).

Painted and unpainted spheres were heated up to approximately 90 °C. When the desired temperature was reached, the heater was turned off and the temperature of the spheres was measured using the platinum resistance sensor and the infrared camera using a fixed ϵ . Every minute a picture of the surface temperature of the spheres was saved and the temperature was compared to the temperature measured by the Pt1000-sensor. The measured temperatures and temperature differences for three different sphere sizes can be seen in [Figure 10](#) -[Figure 12](#). Using $\epsilon = 0.98$ led to high errors for higher temperatures for $d = 10 \text{ mm}$ spheres ([Figure 10](#)), but gave reasonable results for $d = 6 \text{ mm}$ spheres ([Figure 12](#)). After 20 minutes ϵ was set to 0.99, thus a steep decrease of the error can be seen in [Figure 12](#). An emission coefficient of $\epsilon = 0.99$ gave satisfying results for a wide range of temperatures for $d = 8 \text{ mm}$ spheres ([Figure 11](#)). In all subsequent experiments an emission coefficient of $\epsilon = 0.99$ was used.

It is worth to note that the temperature measured with the infrared camera was fluctuating by $\pm 2^\circ\text{C}$, which is exactly the accuracy of the IR-camera according to the manufacturer.

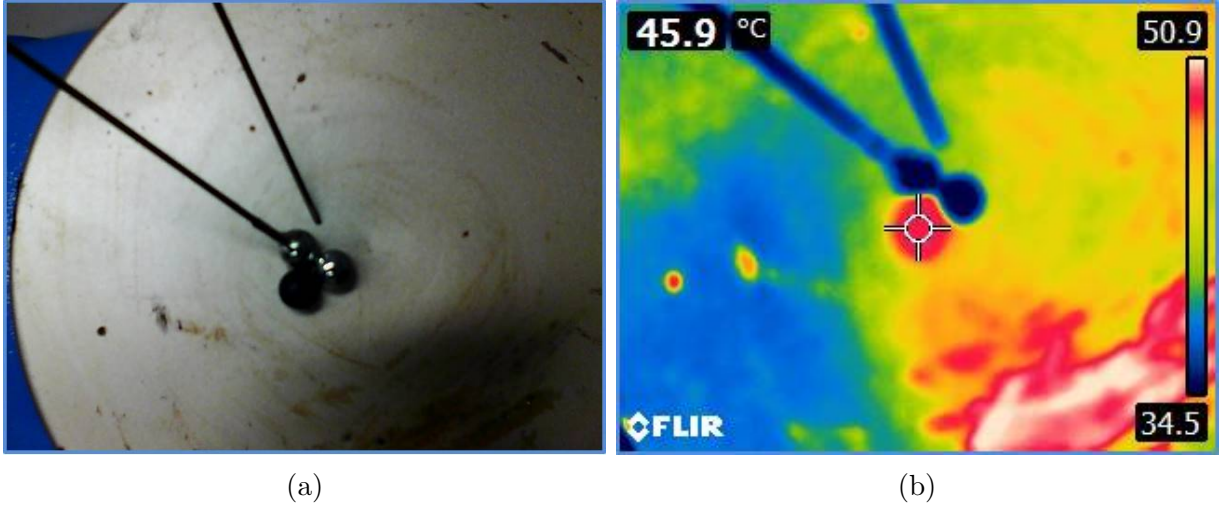


Figure 9: Experimental setup to identify the emission coefficient ϵ . (a) Heater plate, Pt1000 element, soldered to a unpainted sphere, and black painted sphere for IR measurement; (b) infrared picture of the setup

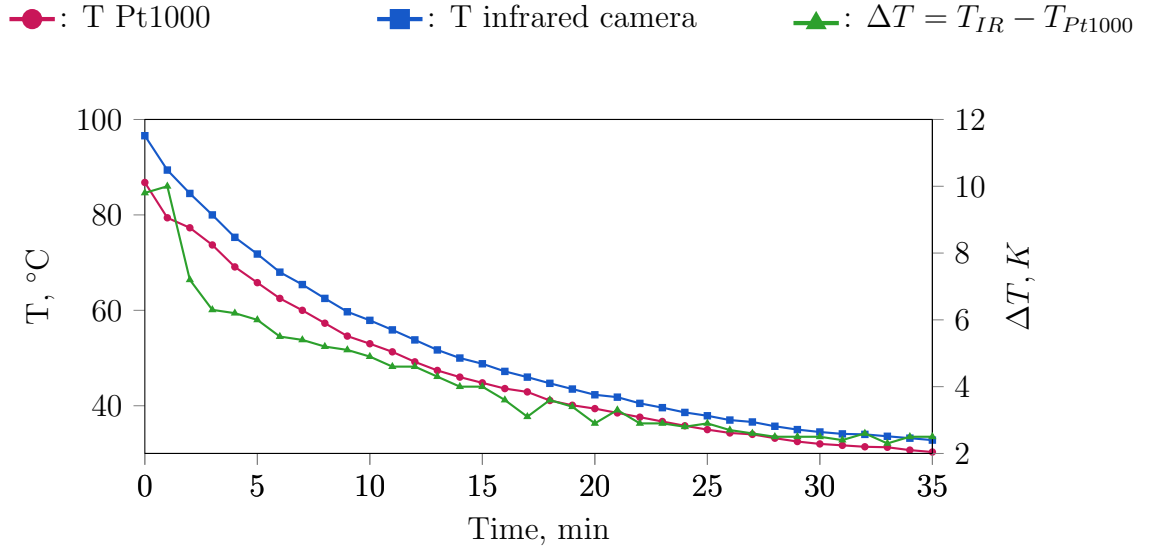


Figure 10: Infrared ($\epsilon = 0.98$) and Pt1000 temperature data for d=6 mm steel spheres

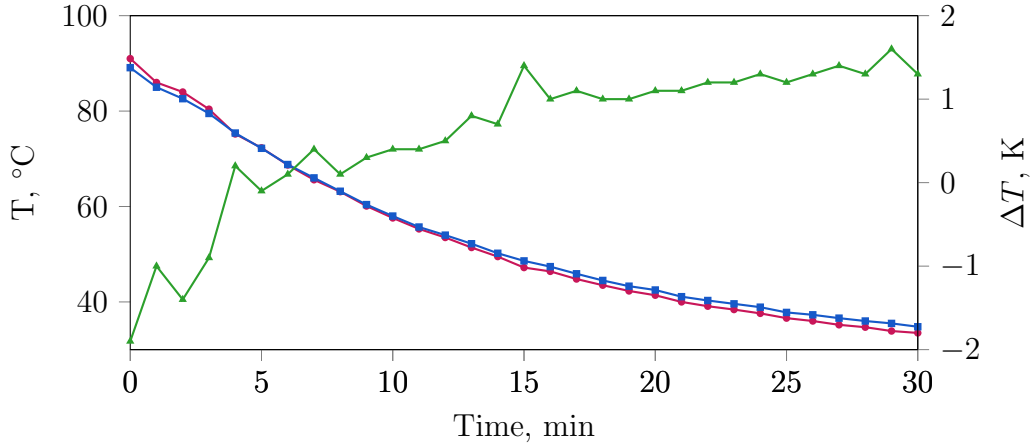


Figure 11: Infrared ($\epsilon = 0.99$) and Pt1000 temperature data for $d=8$ mm steel sphere

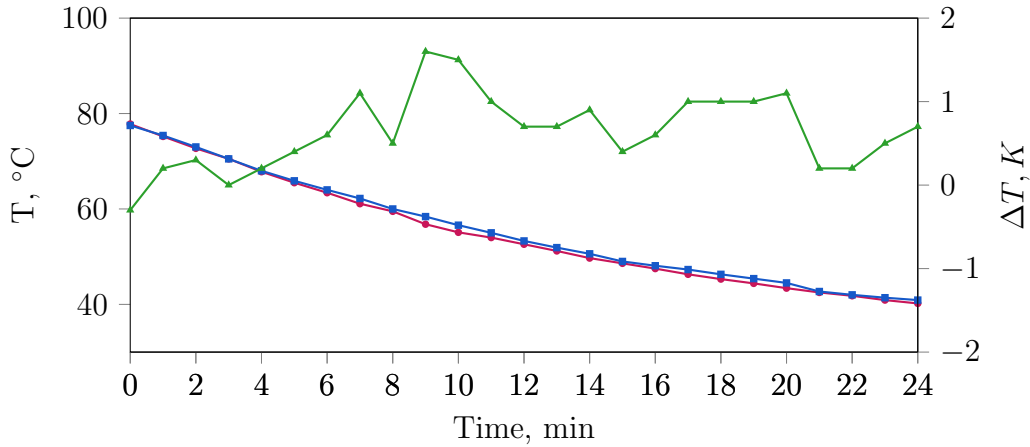


Figure 12: Infrared ($\epsilon = 0.98$) and Pt1000 temperature data for $d=10$ mm steel sphere. After 20 minutes ϵ was set to 0.99 .

4.2 Heater Power as Function of the Heater's Temperature

Two self regulating PTC (positive thermal coefficient of resistance) heaters, with a nominal power of 20 W each, have been used for heating up the packed beds. Such heater have high power when they are cold and rapidly heat them up to a constant temperature. To get accurate simulation results, a temperature dependent volumetric heat-source-term was implemented in OpenFOAM®. To obtain the needed temperature dependent power of a single heating element, the drawn current was measured for different regulated voltage values (15 V, 20 V, 25 V, 30 V) at several temperatures ($\approx 20\text{--}80$ °C). To get constant temperatures a thermostat was used and the resulting heater temperature was measured for each voltage value. As reported in [Section 6.4.4](#) this volumetric heat source did not give correct results for bigger time steps, and thus was not used in the simulations. Nevertheless the experimental data is presented here.

[Figure 13](#) shows the obtained power-data for different temperatures and voltage values. For calculating the needed volumetric power, a heater volume of $V = 2.068 \cdot 10^{-6} \text{ m}^3$ was

used. The regression coefficients for $U = 30\text{ V}$ are presented in Equation 4.32 (where T is the temperature in K). The polynomial regression lines for different voltage values got implemented in OpenFOAM®.

$$f_1 = 242898375.4 - 1195971.0 T + 1447.6 T^2 \quad (4.32)$$

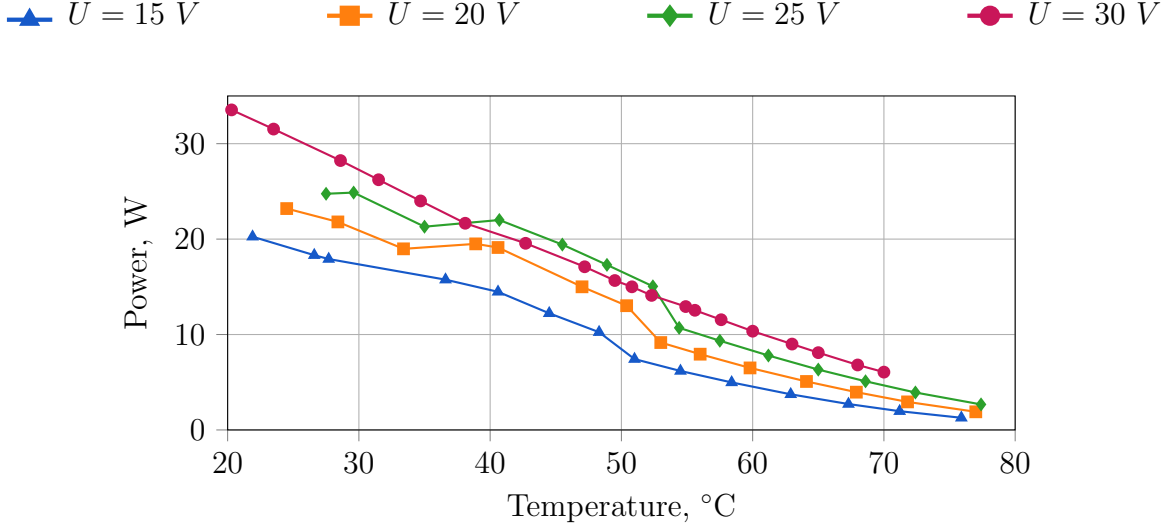


Figure 13: Temperature dependence of the heater power for different voltage values.

4.3 Chemical Composition of Steel Spheres

X-ray fluorescence analysis was used to verify the chemical composition of the used steel spheres. Results for 10 mm, 8 mm and 6 mm spheres are shown in Table 4. Based on the presented results, it was decided to use material properties of low alloy steel: specific heat capacity $c_p = 430\text{ J/(kg K)}$; thermal conductivity $\kappa = 54\text{ W/(m K)}$; density $\rho = 7850\text{ kg/m}^3$.

Table 4: Composition of steel spheres using X-ray fluorescence analysis

Element	10 mm spheres	6 and 8 mm spheres
Mo	0.01 %	n. n.
Nb	n. n.	n. n.
Cu	0.03 %	0.04 %
Ni	0.03 %	0.03 %
Fe	99.31 %	98.99 %
Mn	0.42 %	0.30 %
Cr	0.05 %	0.52 %
V	n. n.	n. n.
Ti	n. n.	n. n.
S	0.03 %	0.02 %
P	0.01 %	n. n.
Si	0.11 %	0.10 %
Total	100 %	100 %

4.4 Density and Heat Capacity of Glass Spheres

First simulation results for glass spheres underpredicted the surface temperature of the packed bed compared to the experiments (Figure 14, and Section 6.4.4.2). Because of the contact point modification used in this simulations, this result was unexpected. Since material properties of glass have been used for bridges, the temperature is expected to be much higher in this simulation. Material properties according to the manufacturer (*Hilgenberg (Technical data sheet 0133)*) are presented in Table 5.

The first obvious source of error are the material properties. Thus the density and heat capacity of the used glass spheres had to be verified. The methods for verifying the material properties are presented in following sections.

Table 5: Thermal properties of soda lime glass according to the manufacturer. *Hilgenberg (Technical data sheet 0133)*

description	value	unit
density ρ_{glass}	2500	$\frac{kg}{m^3}$
heat capacity c_{glass}	1335	$\frac{J}{kg K}$
thermal conductivity k_{glass}	1.129	$\frac{W}{m K}$

— Experiment IR-cam — Experiment Pt1000 — Simulation

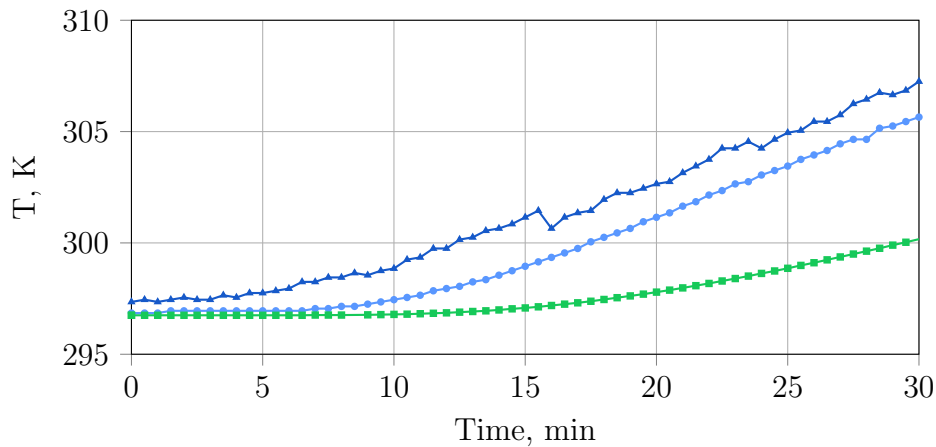


Figure 14: Surface temperature of $d = 6 \text{ mm}$ glass spheres, $h_{bed} = 40 \text{ mm}$. Solid bridges with $d/D = 0.14$ and glass properties have been used as contact point treatment in the simulation.

4.4.1 Glass Density

A 200 ml measuring cylinder was filled with 50 ml tap water and placed on a lab balance. Spheres ($d=6 \text{ mm}$) were added till the volume increased by 10 ml, while measuring the mass of the added spheres. A total mass of 25.35 g was added. Due to inaccuracy in reading the increase of the volume, a tolerance of $\pm 0.57 \text{ g}$ (\pm two spheres) is assumed. The resulting density agrees with Table 5.

$$\rho_{glass} = \frac{\Delta m_{glass}}{\Delta V_{Water}} = \frac{(25.35 \pm 0.57) 10^{-3}}{10 \cdot 10^{-6}} = 2535 \pm 57 \frac{kg}{m^3} \quad (4.33)$$

4.4.2 Glass Heat Capacity

To validate the heat capacity a calorimeter was used. A defined volume of water and mass of spheres were filled into the vessel. Under continuous stirring water and spheres were heated up using a electrical heater, while measuring the water temperature. The principle of the calorimeter and the experimental setup are shown in [Figure 15](#) and [Figure 16](#) respectively. When neglecting heat losses to the environment the heat capacity of the spheres can be calculated using a heat balance ([Equation 4.34](#)). Here C_c is the calorimeter constant, \dot{Q}_{heater} is the heater power, T is the current temperature, T_0 the initial temperature and c and m are the heat capacity and mass respectively.

$$(C_c + m_W c_W + c_{glass} m_{glass}) (T(t) - T_0) = \dot{Q}_{heater} t \quad (4.34)$$

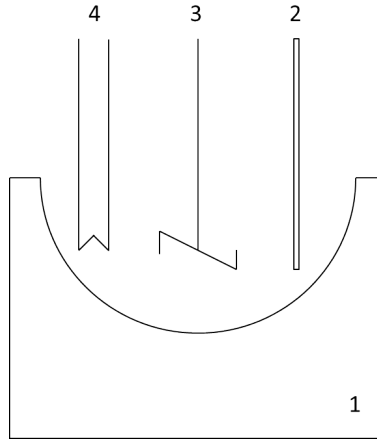


Figure 15: Prinzipel of the calorimeter used to determine the heat capacity. 1...isolated Dewar vessel, 2...temperature probe, 3...stirrer, 4...electrical heater

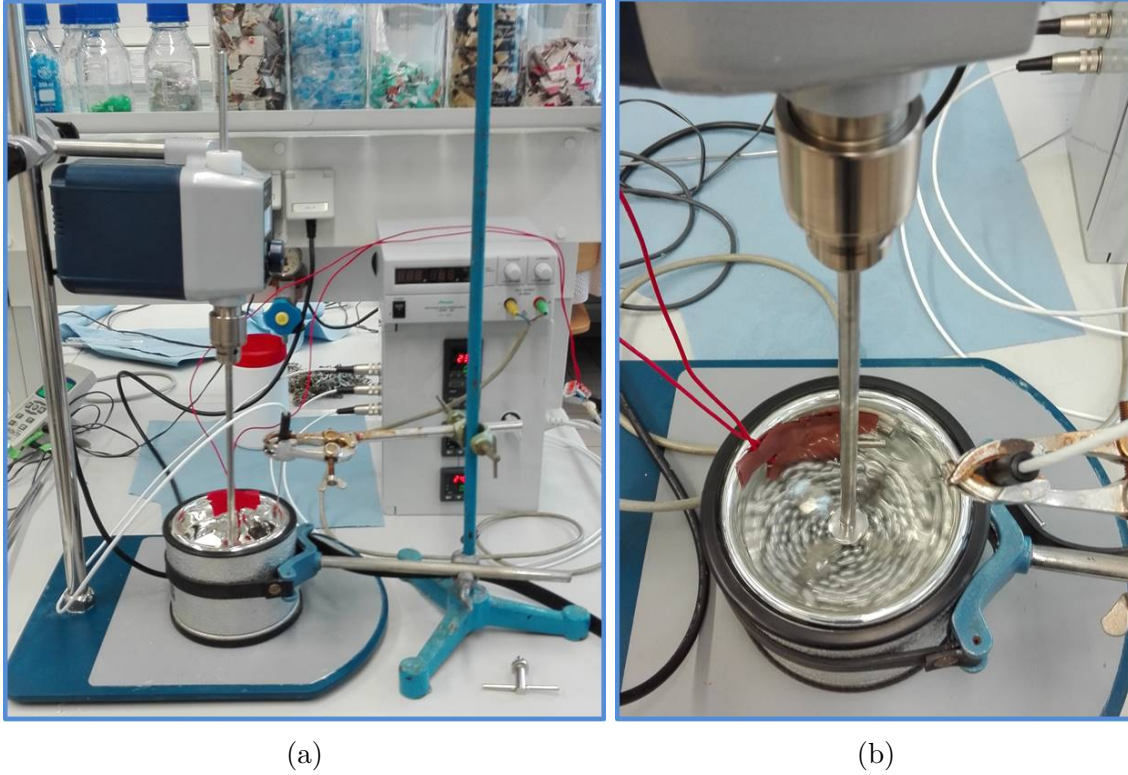


Figure 16: Experimental setup to identify the heat capacity of different spheres.

To get c_{glass} the calorimeter constant C_c is needed. It is determined by heating up a defined amount of water. When doing so, the heat balance is reduced to Equation 4.35 and C_c can be calculated. The density and heat capacity of water used are $\rho_W = 997 \frac{kg}{m^3}$ and $c_w = 4184 \frac{J}{kg K}$ respectively.

$$(C_c + m_W c_W) (T(t) - T_0) = \dot{Q}_{heater} t \quad (4.35)$$

To get C_c 220 ml water was heated for four minutes, using a electrical current of $I = 2.8 A$ and a voltage of $V = 10 V$ under constant stirring (200 rpm). The temperature was recorded every 30 seconds. This procedure was repeated four times. The final calorimeter constant was calculated as mean value of all recorded time steps to $C_c = 273.58 \pm 33.3 \frac{J}{K}$ assuming a temperature reading error of $\pm 0.1 K$. The standard deviation can be calculated to $s = 30.9 \frac{J}{K}$. Experimental data is presented in Table 6.

A validation experiment using 200 ml water and 119 g aluminium spheres was carried out, using this calculated calorimeter constant. The heat capacity of aluminium was calculated according to Equation 4.34 to $c_{Al} = 750.5 \pm 116.5 \frac{J}{kg K}$ which is in the range of values found in literature (VDI-Wärmeatlas (2013): $c_{Al} = 837 \frac{J}{kg K}$). The high derivation is based on the uncertainty of reading the temperature.

After a successful validation of the method, experiments to determine the heat capacity of $d = 6 mm$ and $d = 10 mm$ glass spheres have been carried out using the same procedure. 200 ml water plus 114 g and 144.6 g of 6 mm and 10 mm glass spheres have been used,

respectively. The experimental results, the heat capacity and their standard deviation for both types of spheres are presented in Table 7. Both values are less than half of the value that the manufacturer gives. Mann, Field, and Viskanta (1992) presented the heat capacity and the thermal conductivity of float glass determined from a dynamic temperature data and compared them to values from literature. In this study the heat capacity was determined at temperatures from 150 °C to 700 °C and then extrapolated to 0 °C. At 25 °C the extrapolated heat capacity was found to be approximately $700 \frac{J}{kg K}$. Based on the findings of the calorimeter experiments for both, aluminium and glass spheres, and the presented values from Mann, Field, and Viskanta (1992), it was decided to use a heat capacity of $c_{glass} = 600 \frac{J}{kg K}$ for glass spheres in the simulations.

Since the thermal conductivity could not be validated in this study, it was decided to use the value presented in table Table 5. The value of $k_{glass} = 1.129 \frac{W}{mK}$ is in good agreement with the thermal conductivity of $k_{glass} = 1.15 \frac{W}{mK}$ at 25 °C presented in Mann, Field, and Viskanta (1992).

Table 6: Temperatures and calorimeter constants at each time step for all four runs

t, min	T, °C	C_c , J/K
Run 1		
0	22.7	-
2	25.5	308.0
2.5	26.3	264.3
3	26.9	293.1
3.5	27.6	288.9
4	28.2	308.0
Run 2		
0	23.5	-
1	24.8	355.2
1.5	25.7	264.3
2	26.3	308.1
2.5	27.0	299.0
3	27.7	293.1
3.5	28.4	301.1
4	29.0	308.1
Run 3		
0	23.9	-
0.5	24.7	264.3
1	25.4	264.3
1.5	26.1	223.5
2	26.9	231.4
2.5	27.6	223.5
3	28.4	233.5
3.5	29.1	217.9
4	29.7	243.5
Run 4		
0	24.4	-
0.5	25.2	264.3
1	25.9	264.3
1.5	26.6	264.3
2	27.3	264.3
2.5	28.0	264.3
3	28.7	264.3
3.5	29.4	264.3
4	30.1	285.8
mean C_c		273.58 ± 30.9

Table 7: Temperatures and heat capacities of 6 mm and 10 mm glass spheres in the calorimeter experiment

t, min	T, °C	C_{glass} , J/(kg K)
d=10 mm glass spheres		
0	24.7	-
1.5	27.3	647.16
2	28.1	716.5
2.5	29.0	592.7
3	29.9	512.5
3.5	30.8	569.6
4	31.7	512.5
mean C_{glass}		591.8 ± 72.7
d=6 mm glass spheres		
0	24.5	-
0.5	25.3	650.2
1	26.0	650.2
1.5	26.7	650.3
2	27.4	650.1
2.5	28.2	362.2
3	28.9	409.1
3.5	29.6	442.9
4	30.4	468.3
mean C_{glass}		535.43 ± 118.2

4.5 Sphere Diameter

The diameter of twenty spheres of each material and diameter was measured, using a micrometer ($\pm 10^{-3}$ mm accuracy) to check the average diameter. From Table 8 it can be seen that all the sphere diameter lie within a reasonable range of their nominal diameter. Only for $D = 10$ mm glass spheres the mean diameter is slightly higher.

Using the voidage model of *deKlerk* (2003) and the α_w -Model, the average voidage and the bed's surface temperature after 30 minutes for 10.2762 mm and 10.0 mm glass spheres, were calculated to $\Psi_{10.0} = 0.4468$, $\Psi_{10.2762} = 0.4479$, $T_{10.0} = 29.659$ °C and $T_{10.2762} = 29.677$ °C, respectively. This results show that the deviation of the sphere diameter does not influence the average voidage and surface temperature significantly. Thus, for simulation purpose, a sphere diameter of 6 mm and 10 mm was used.

Table 8: Mean diameter and standard deviation of used spheres

	$D = 6$ mm			$D = 10$ mm		
	Aluminium	Steel	Glass	Aluminium	Steel	Glass
mean diameter, mm	5.9574	5.9476	5.9554	9.9962	10.0419	10.2762
standard deviation, mm	0.00679	0.00254	0.1116	0.007	0.00779	0.09835

4.6 Heat Transfer Experiments

To validate the different simulations, using different particle sizes, materials, bridge sizes and effective thermal conductivities for the bridges, experiments had to be carried out. In this experiments a defined number of spheres was randomly packed on a aluminium plate in an isolated plastic container (inner dimensions: 85x51.4 mm, [Figure 17](#)). The plate was heated from below with two self regulating heaters, with a nominal power of 20 W each and a maximal temperature of approximately 70 °C. The heaters and the surface temperature of the packed bed was monitored using Pt1000 temperature probes. In addition to that the temperature distribution on the surface was monitored using a FLIR® E6 infrared camera. Heating was continued till the surface temperature increased by a certain amount, usually 8 to 10 °C. The temperature was measured at a height of 40 mm in the centre of the packing's surface. The most important dimensions of the experimental setup are shown in [Table 9](#).

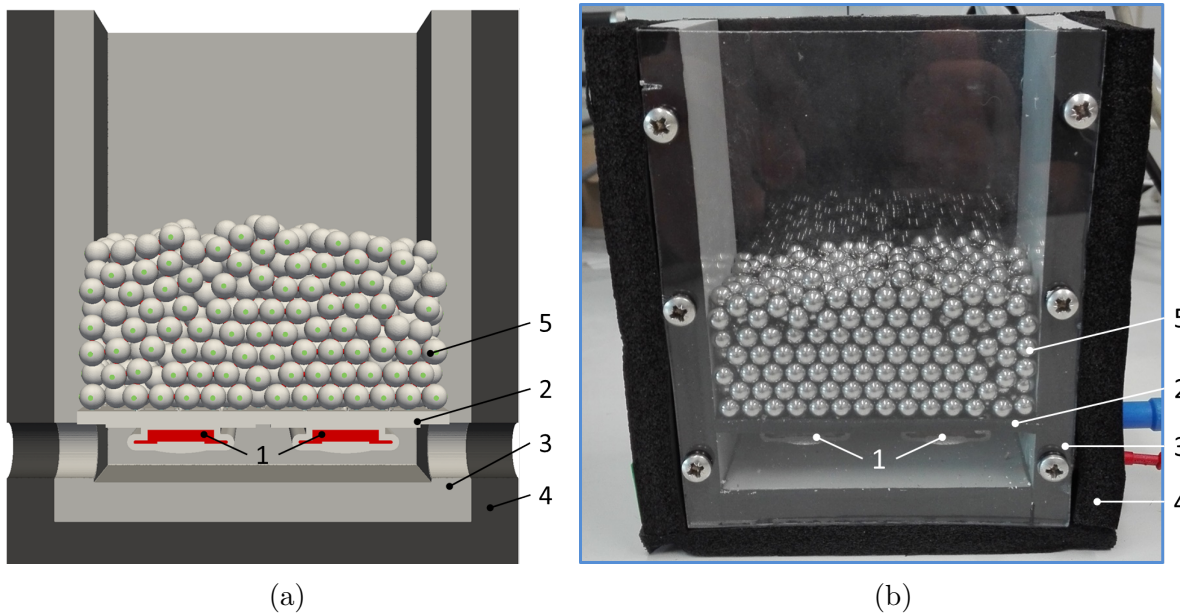


Figure 17: (a) Drawing of the experimental geometry; (b) Experimental geometry
1...heating elements, 2...aluminium bottom plate, 3...plastic container, 4...isolation, 5...sphere packing

Table 9: Inner dimensions of the experimental setup

length aluminium plate	85 mm
width aluminium plate	51.4 mm
thickness aluminium plate	3 mm
thickness plastic parts	10 mm
thickness plexi front and back	3 mm
thickness isolation	12 mm
height from bottom plate to top	98 mm

A total number of 10 experiments have been carried out ([Table 10](#)), though not all of them

have been used in the Design of Experiments. Figure 18 shows the experimental data of $D = 6 \text{ mm}$ aluminium spheres with one repetition. One can see that the heaters temperature increases steeply in the first few minutes. This behaviour is typically for PTC (positive thermal coefficient of resistance) heaters. The surface temperature starts to increase slowly after a breakthrough time. The goal is to get the same surface temperature evolution in the simulations, when using the measured heater temperature as boundary condition.

Table 10: List of experiments carried out in this work.

Material	Diameter	Nr. of Spheres	Figure Nr. (Appendix)	Repetition	Used in
Aluminium	6 mm	619	Figure A.1	0	DoE
Aluminium	6 mm	920	Figure A.2	1	necks method
Aluminium	6 mm	920	Figure A.3	0	necks method
Aluminium	10 mm	200	Figure A.4	0	DoE
Glass	6 mm	619	Figure A.5	2	DoE
Glass	6 mm	~ 2000	Figure A.8	0	-
Glass	10 mm	200	Figure A.7	0	DoE
Steel	6 mm	920	Figure A.9	0	-
Steel	8 mm	496	Figure E.1	0	validation
Steel	10 mm	200	Figure A.10	0	validation

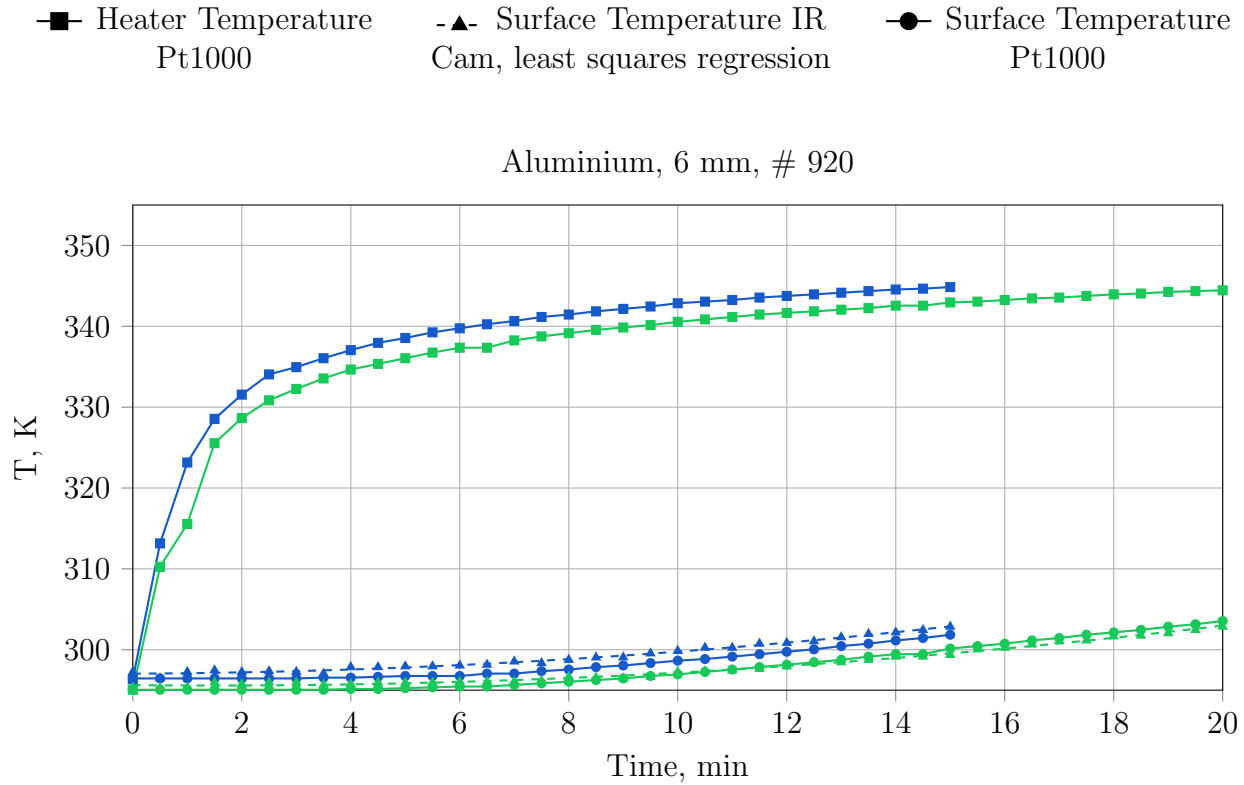


Figure 18: Heater temperature, and surface temperature of a packed bed containing 920 $D = 6 \text{ mm}$ aluminium spheres. Different colours refer to different experimental runs.

5 Computational Fluid Dynamics (CFD)

Flows and related phenomena can be described by partial differential equations ([Equation 5.36a](#)-[Equation 5.36c](#)), which generally speaking cannot be solved analytically. Discretization methods have to be used to get an approximate solution. The approximations are applied to small domains in space (grid or mesh) and/or time, which leads to discrete locations in space and time. The discretized governing equations of fluid dynamics can then be solved on a computer. The accuracy of numerical solutions depends on the quality of discretizations used.

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \quad (5.36a)$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i + F_i \quad (5.36b)$$

$$\frac{\partial (\rho h_i)}{\partial t} + \frac{\partial (\rho u_i h)}{\partial x_i} = \frac{\partial}{\partial x_i} \lambda \frac{\partial T}{\partial x_i} - \frac{\partial \Sigma_j h_j J_j}{\partial x_i} + \frac{\partial p}{\partial t} + \sigma_{ik} \frac{\partial u_i}{\partial x_k} + S_k \quad (5.36c)$$

There are many approaches for discretization, but the most important of which are: finite difference, finite volume and finite elements methods. Since only the first two are used in this thesis, the finite elements method is not dealt with any further.

For the finite difference method, the starting point is the conservation equation in differential form. The solution domain is covered by a grid. At each grid point, the differential equation is approximated by replacing the partial derivatives by a Taylor series expansion or a polynomial fitting of the partial derivatives. The result is one algebraic equation per grid node, in which the variable value at that node and a certain number of neighbour nodes appear as unknowns. On structured grids, the finite difference method is very simple and effective. The disadvantage is that the conservation is not enforced unless special care is taken. Also, the restriction to simple geometries is a disadvantage in complex flows.

The finite volume method uses the integral form of the conservation equations as its starting point. The solution domain is divided into a finite number of control volumes (CV's), and the conservation equations are applied to each of them. Interpolation is used to express variable values at the CV surface in terms of the nodal values. The finite volume method can be used at any type of grid, so it is suitable for complex geometries. (*Ferziger and Peric (2002)*)

In this work the free, open source software OpenFOAM® is used for simulation. For mesh generation snappyHexMesh and cfMesh are used.

In the subsequent chapters first the creation of the packing and the meshing procedure of snappyHexMesh is described, then two different methods of contact point treatment are discussed. The first method is to use bridges at particle-particle contact points and gaps or bridges at particle-wall contact points. Here the bridges are of the same material as the packing. This method, but using bridges with an effective thermal conductivity according to [Equation 2.27](#) at particle-wall contact points instead of gaps, was also used by *Dixon et al. (2012)*. The authors ascribe this method the name 'necks-method'. Gaps between particle and walls have been introduced to avoid boundaries at particle-wall contact points.

A second way of treating contact points is then discussed in [Section 6.5](#). Here cylindrical bridges are introduced at all particle-particle and particle-wall contact points. An effective thermal conductivity using [Equation 2.27](#) is used in all the simulations. This method is later used to carry out the Design of Experiments described in [Section 3](#) and [Section 6.5](#).

6 Methods, Results and Discussion

6.1 Creation of a Randomly Packed Bed of Spheres

To create packed beds with properties (voidage) close to reality, an automatic generation is necessary. The method of choice to create such packings in this work is the Discrete Element Method (DEM). The DEM is an extension of the Lagrangian modelling approach, in which inter-particle contact forces are included into the equation of motion. In the soft-particle formulation, the contact forces, obtained from particle overlaps, are proportional to the overlap, the particle material and the geometric properties (*Wehinger (2016)*).

In this work an inhouse DEM-code, developed by Hamidreza Norouzi, is used. The procedure of the bed creation is shown in [Figure 19](#). (a): A defined number of particles is randomly inserted with a given initial velocity at an insertion plane into the given geometry. (b): The code was always running long enough to allow all particles to settle. No shaking or vibrating was carried out. (c): Using the output of the DEM-code, a Stereolithography (STL)-file of the final packed bed can be created, using a tool built in the DEM-code or external software (e.g. SALOME). (d): This file can later be used for meshing in snappyHexMesh or cfMesh.

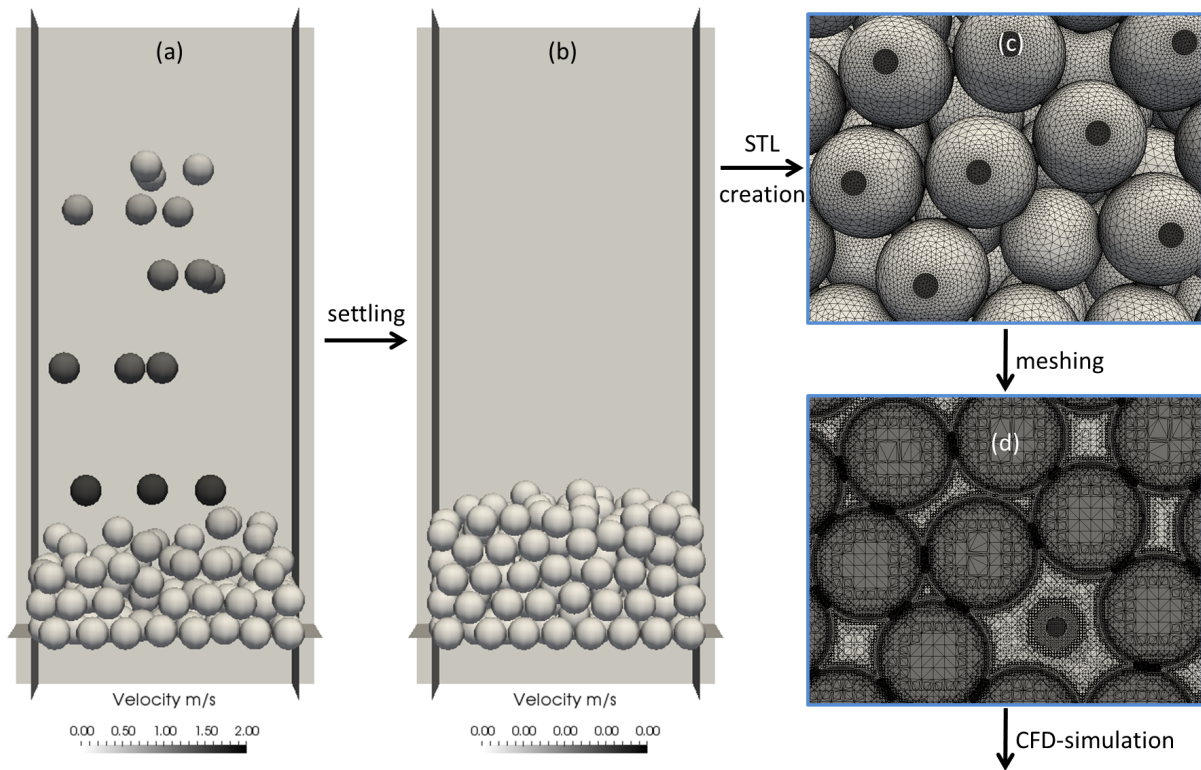


Figure 19: Procedure of creating a randomly packed bed of spheres. Bridges are inserted in SALOME before exporting the STL-files.

6.2 Meshing Procedure of snappyHexMesh

The mesh generation process of snappyHexMesh has direct influence on the resulting mesh (in particular on the form of the bridges between particles). For that reason this process is

described here very briefly.

In snappyHexMesh a mesh of hexahedral cells (background mesh), which covers the entire region, is manipulated to approximate a given triangulated surface geometry (Stereolithography (STL) or Wavefront Object (OBJ) format). Essentially mesh generation is done in four steps:

1. cell splitting of the background mesh
2. cell removal
3. snapping to surface
4. introduction of mesh layers

In a first step the cells of the background mesh are divided into smaller sub-cells (four in 2D, 8 in 3D meshes) at edges (using a eMesh file containing the feature line specification) and surfaces. Considering that cell splitting is defined by levels of refinement (number of times the cells are splitted), the cell size of the background mesh has a significant influence on the final cell size.

After cell splitting cells are removed, if 50% or more of their volume lies outside of the specified region. So only cells which are inside the defined region are retained.

The cell vertex points of the remaining cells, which lie in close proximity of the triangulated surface, are moved to that surface in a third step. This displacement of vertex points is continued until certain mesh quality criteria are satisfied. After this step the geometry is represented by finite volumes and basically ready for simulation.

In some cases it is desired to have additional boundary layers of hexahedral cells at surfaces. These boundary layers are introduced in a fourth step by shrinking the existing mesh and inserting layers of cells.

6.3 Discretization of the α_w -Model

For the purpose of later use Equation 2.16 shown in 2.1.5 is discretized using a central finite different scheme (Forward in Time, Centred in Space = FTCS) in Equation 6.37a. Here the superscript n stand for the discretization in time, the subscript j is the discretization in space.

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = a \frac{T_{j+1}^n - 2T_j^n + T_{j-1}^n}{\Delta x^2} \quad (6.37a)$$

with

$$a = \frac{\Lambda_{ax}}{(1 - \Psi) \rho_p c_p + \Psi \rho_f c_f} \quad (6.37b)$$

The thermal diffusivity a of the bed is according to Equation 6.37b.

For the boundary conditions Equation 2.17, the temperature \underline{T} along the packing height at a given time step $n+1$ can be calculated to:

$$\underline{T}^{n+1} = \underline{A} \underline{T}^n + \underline{T}_{bottom}^{n+1} \quad (6.38a)$$

with

$$\underline{T}_{bottom}^{n+1} = \begin{bmatrix} \frac{T_{bottom}^O - T_{bottom}^U}{t^{n+1} - t^n} \Delta t \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (6.38b)$$

$$\underline{\underline{A}} = \begin{bmatrix} 1 & 0 & 0 & \dots & \dots & \dots & 0 \\ s & 1-2s & s & \ddots & \dots & \dots & 0 \\ 0 & s & 1-2s & s & \ddots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & \ddots & s & 1-2s & s & 0 \\ \vdots & \dots & \dots & \ddots & s & 1-2s & s \\ 0 & \dots & \dots & \dots & 0 & s & 1-s \end{bmatrix} \quad (6.38c)$$

where

$$s = a \frac{\Delta t}{\Delta x^2} \quad (6.38d)$$

T_{bottom}^O and T_{bottom}^U are the measured heater temperatures at different times. $\underline{T}_{bottom}^{n+1}$ is the interpolated heater temperature at time step $n+1$. Equation 6.38 was implemented in python to be able to compare the results to the experiments and CFD-simulations. To get a stable simulation s has to be smaller than 0.5.

The python code of the discretized α_w -Model is presented in Appendix H.

6.4 The necks method

One of the simplest ways of simulating the heat transfer in packed beds is the so called necks method. Here bridges, with same thermal properties as the bed, are introduced at particle-particle contact points. This method has been used by *Dixon et al. (2012)*. In the present study no cylindrical bridges are inserted at contact points, but bridges are automatically generated in the meshing process. The goal here was to find an easy way to mesh the packing and to find the optimal bridge size, where heat transfer is predicted with enough accuracy and the mesh size is still reasonable.

To reduce meshing times a test case, consisting of four regions, was created. The test geometry can be seen in Figure 20. First different methods of single-region meshing, then different ways of multi-region meshing are discussed.

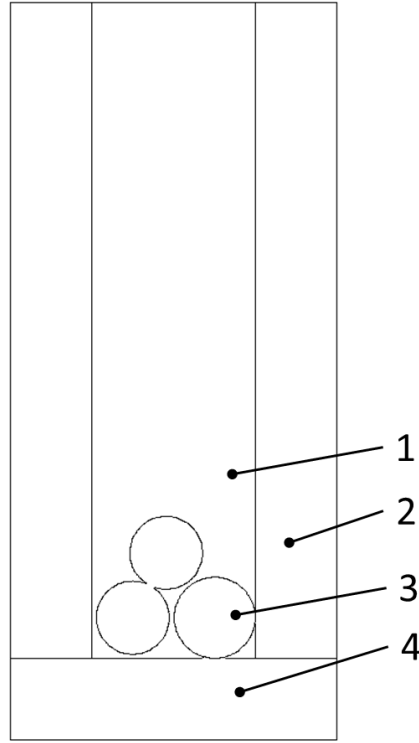


Figure 20: Test case for meshing. 1...fluid; 2...plastic container; 3...packing, consisting of five $D = 10 \text{ mm}$ spheres; 4...heater

6.4.1 Single Region Approach

In the single region approach all regions of the test geometry are meshed separately. To get a bridged sphere packing, first the fluid region is meshed. Because of the maximum refinement level (equivalent to the minimum cell size), contact points can not be resolved completely, resulting in a bridged packing. The packing surface is then exported as STL-file, using the *surfaceMeshTriangulate* function of *snappyHexMesh*, which is then used to mesh the packing.

The advantage of single region meshing over multi region meshing, is that the mesh quality is usually higher. The disadvantage here is, that small gaps have to be introduced between packing and walls to avoid boundaries, which leads to an underestimation of the heat transfer from heated wall to packing. Assuming that the main path for heat transfer from a heated wall to a spherical particle near the contact point is heat transfer through the stagnant fluid, small gaps might not have a big impact on the result. Tested gap sizes are 0.05 mm, 0.1 mm, 0.2 mm and 0.3 mm (Figure 21). It is important that the cell size at particle-particle contact points can be changed independently of the cell size at particle-wall contact points.

For the single region approach different ways to resolve the small particle-wall gaps have been tested.

- Refinement boxes at particle-wall contact points
- Surface refinement
- Gap refinement

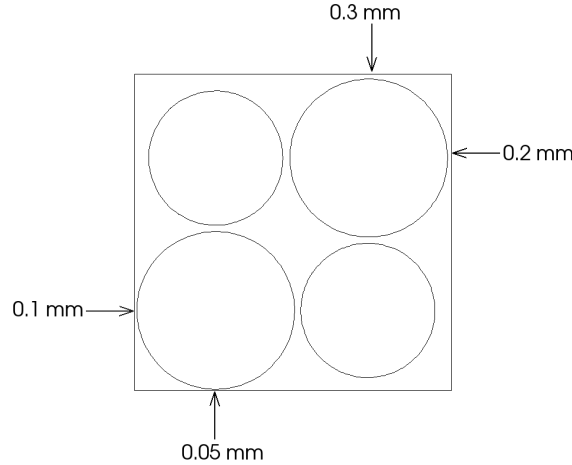


Figure 21: Gap sizes at different walls

However none of this methods led to satisfying results.

When using region based refinement at walls not only cells in close proximity to particle-wall contact points, but all cells inside the defined region are refined. This leads to a impracticable high number of cells.

By defining a range of e.g. *level (2 8)* for surface refinement, *snappyHexMesh* should apply a appropriate refinement-level at gaps. However *snappyHexMesh* was only able to capture all the gaps when using a uniform refinement level for walls and packing (Figure 22). This again leads to high cell counts, especially for small gaps.

When using the *gapLevelIncrement* function of *snappyHexMesh* to refine only cells in small gaps, the cell size at particle-particle contact points can not be changed independently of the cell size at particle-wall contact points. This is because a maximum gap refinement level has to be defined at all surfaces. This refinement level is then applied at all gaps between these surfaces (particle-wall and particle-particle). A minimum refinement level is needed to resolve gaps at particle-wall contacts, and the same refinement level is applied at particle-particle contact points, which leads to small bridges. A gap is defined as a region between geometry surfaces that are facing one another, that is spanned by only one or two cells. However changing the bridge size between particles independently of the gap size at walls is essential for heat transfer simulations.

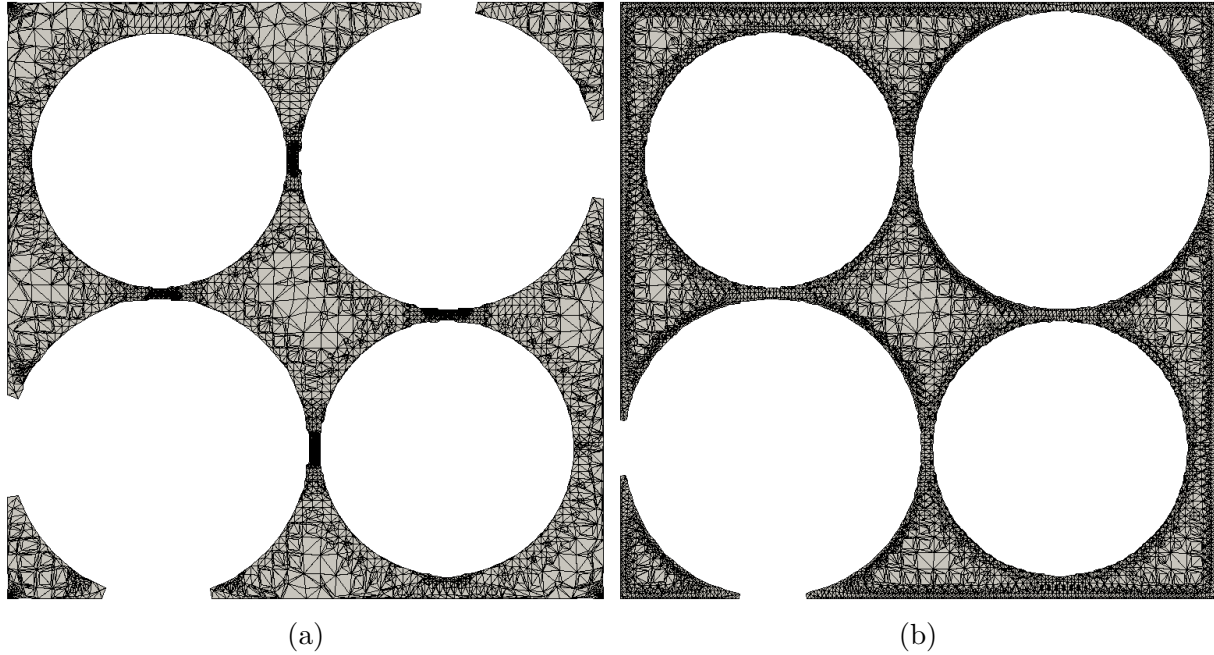


Figure 22: Gap refinement via surface refinement (a) *level (2 8)* and *level (4 8)* applied at surface geometry and packing respectively. None of the gaps are captured by the mesh. (b) uniform refinement *level (5 5)* at all surfaces.

6.4.2 Multi Region Approach

In the multi region approach all regions are meshed simultaneously. Using this approach, no gaps at walls are needed because particle-wall boundaries are created automatically. This boundaries can later be used in the simulation to enable heat transfer from walls to particles. Here bridges are not only created at particle-particle contact points, but also between walls and particles, as no gaps are used.

When using the multi region approach, it is necessary to define uniform surface refinement levels for packing and adjacent walls to avoid cells which could cause problems in the simulation. This solution leads to higher cell counts, compared to local refinement at contact points. In [Figure 23](#) one can see that the problematic cells ([Figure 23a](#); colored yellow) are located where cell levels are changing. Apparently this cells are highly non-orthogonal, which is defined as the angle α between the cell-cell vector and the face normal vector ([Figure 24](#)). Orthogonality is a important factor for calculating the face normal gradient. The basis of the gradient calculation at a face is to subtract the value at the cell centre on one side of the face from the value in the centre on the other side and divide by the distance. The calculation is second-order accurate for the gradient normal to the face, if the vector connecting the cell centres is orthogonal to the face. To maintain second-order accuracy for non-orthogonal meshes, an explicit non-orthogonal correction can be added to the orthogonal component. The correction increases in size as the non-orthogonality increases. As α tends towards 90 °C, e.g. beyond 70 °C, the explicit correction can be so large to cause a solution to go unstable (*The OpenFOAM Foundation (OpenFOAM v5 User Guide)*).

That's exactly what happened in simulations containing non-orthogonal faces at mesh region surfaces. The temperature in the non-orthogonal cells started to oscillate and simula-

tion crashed after few iteration after the oscillation started. By setting a uniform refinement level at surfaces, this non-orthogonal cells could be avoided.

Another way to get a stable simulation is to remove problematic cells. This can be done by grouping all cells exceeding a certain *maxNonOrtho*-value and using the *splitMeshRegions* function of *snappyHexMesh*. The result of this is a separate region containing only problematic cells. This cells can then be deleted or isolated in the simulation. This procedure is a good alternative to the non-orthogonal corrections, if only a small number of cells has to be removed.

It was also tried to get rid of this cells, by changing mesh quality parameter in *snappyHexMesh*. Unfortunately none of the tried settings could get rid of the problematic cells. Mesh quality parameter used for generation of all grids are shown in [Appendix B](#). Notably the *maxNonOrtho* and the *minDeterminant* were set to 55 (0 = hexahedral cells) and 0.05 (1 = hexahedral cells) respectively to avoid skewed cells and increase the mesh quality. By further decreasing *maxNonOrtho* and increasing *minDeterminant* snapping of the mesh to the STL-surface gets worse. So the used quality parameters are a trade-off between mesh quality and surface approximation ([Figure 25](#)).

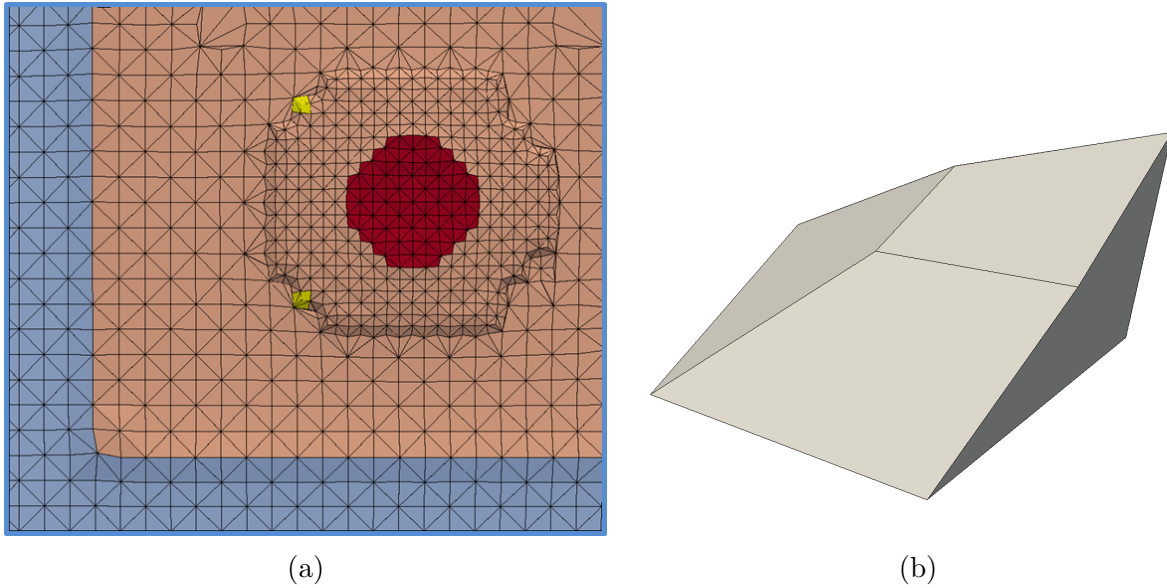


Figure 23: Multi region surface refinement *level (2 3)* and *level (4 5)* for heater and packing respectively (a) yellow: location of problematic cells; colored: different boundaries (b) closer view of problematic cell.

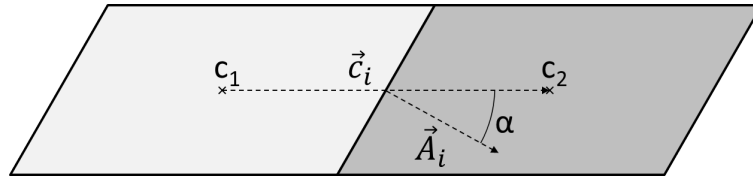


Figure 24: Definition of face non-orthogonality α . c_1 and c_2 : cell centres; \vec{c}_i : vector connecting both cell centres; \vec{A}_i : face normal vector; α angle between c_i and A_i

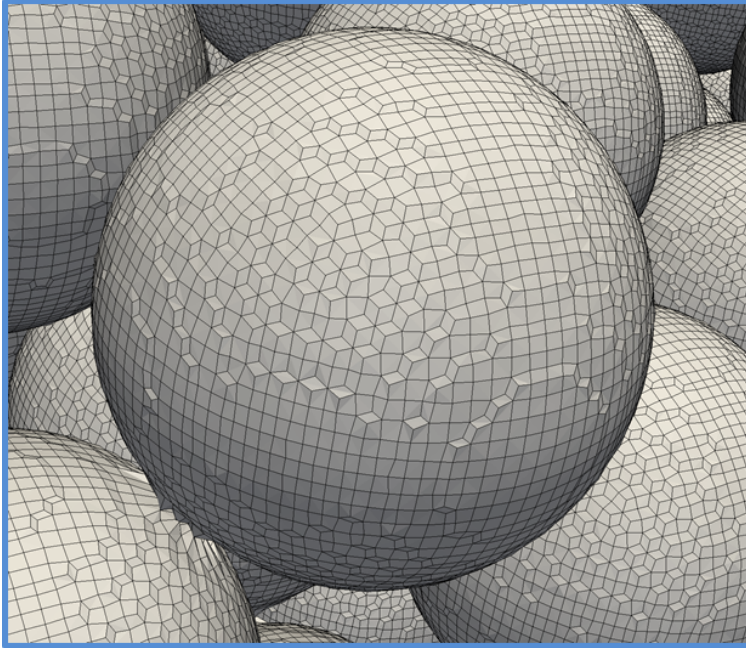


Figure 25: Close look on the surface of a sphere meshed using the multi-region approach with mesh quality parameter shown in [Appendix B](#).

6.4.2.1 Bridge size

To get the same ratio of bridge to particle diameter d/D for all the particles it is necessary to use different background mesh sizes for different particle sizes D . Therefore a parameter-variation study for different particle sizes was executed, using the test geometry shown in [Figure 20](#) and different sphere diameter. Since *snappyHexMesh* does not create perfect cylindrical bridges, the bridge diameter d is calculated based on the cross section A_b ([Equation 6.39](#)). Results are represented in [Figure 26](#). It can be seen that different snapping methods lead to different bridge sizes. A combination of implicit/explicit feature snapping and different background mesh sizes can be used to get desired bridge size.

Explicit feature snapping uses the created eMesh file with the feature line specification. Using the implicit feature definition, *snappyHexMesh* does not read eMesh files, but defines sharp edges itself (*OpenCFD Ltd. 2017 (OpenFOAM® Documentation: Extended Code Guide)*).

$$d = \sqrt{\frac{4 * A_b}{\pi}} \quad (6.39)$$

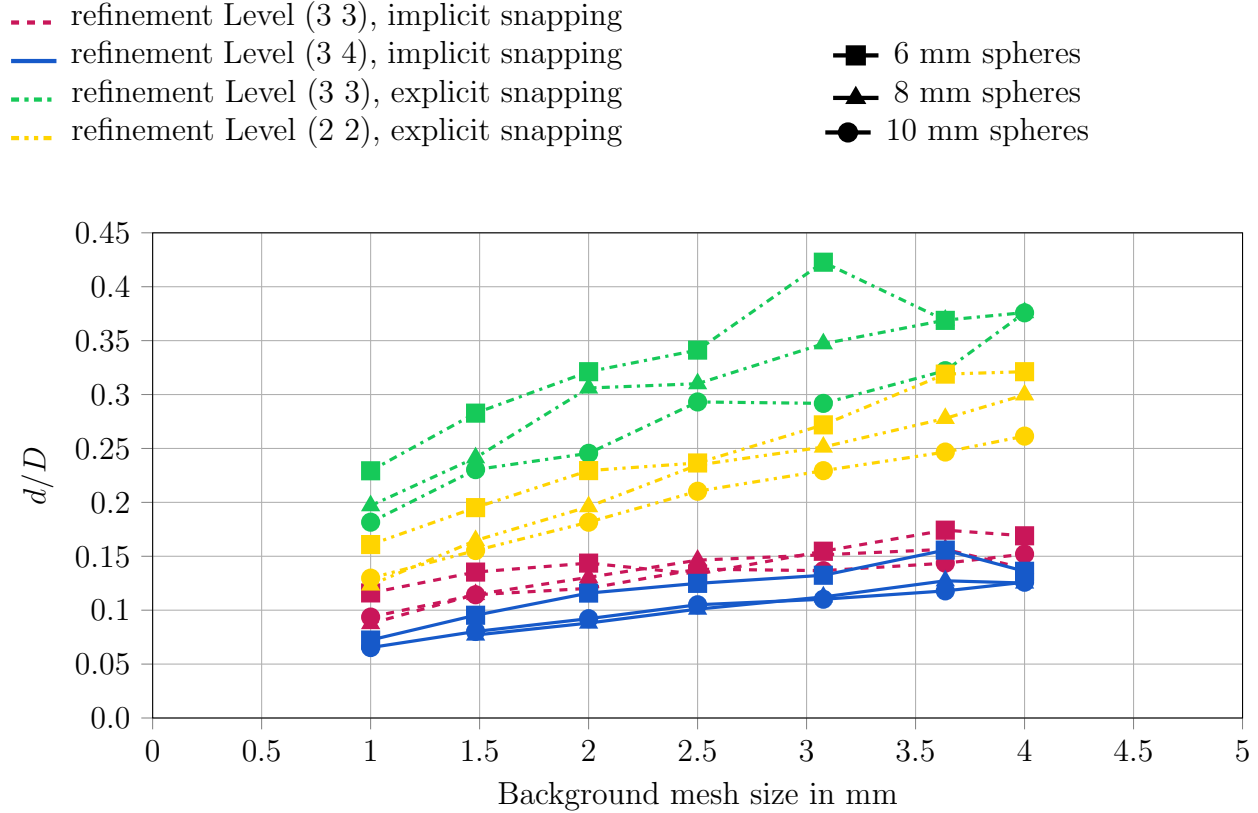


Figure 26: Bridge size d/D for different snapping methods, background mesh sizes and surface refinement levels for the sphere packing.

Bridges between particles look different depending on their orientation relative to the background mesh, as one can see in [Figure 27](#). This is a direct consequence of the snapping phase of the mesh creation. In this phase all cell vertex points within a given range of a surface are moved to this surface geometry. In planes parallel to the background mesh, all the vertices are closer to a surface than in nonparallel planes. Therefore the bridges get much smoother (see also [Section 6.2](#)). The location of the bridges shown in [Figure 27](#) is presented in [Figure 28](#).

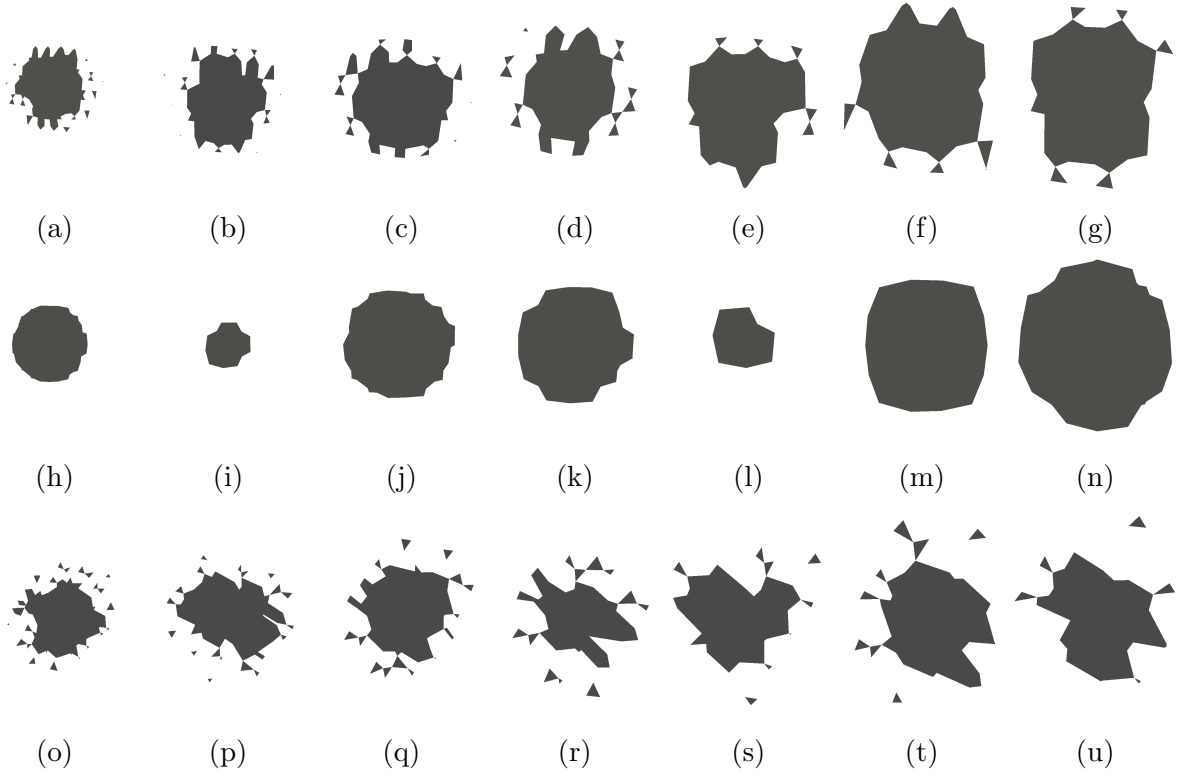


Figure 27: Cross section of bridges between spheres using background mesh sizes from 1 mm (left) to 4 mm (right). (a)-(f): Spheres located in different sphere layers, explicit snapping; (h)-(n): Spheres located in same layer, explicit snapping; (o)-(u): Spheres located in different layers, implicit snapping

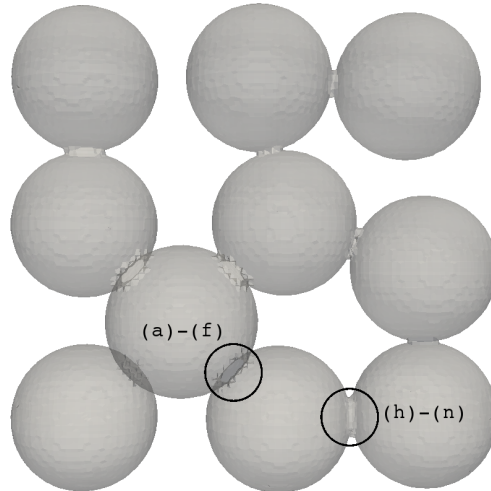


Figure 28: Location of bridges shown in Figure 27. (a)-(f): location of bridge between particles in different sphere layers, (h)-(n) location of bridge between particles in the same sphere layers

6.4.3 Testing the Volumetric Heat Source

To automatically capture the self regulation of the PTC heating elements, a temperature dependent heat source term was added to the *chtMultiRegionFoam* solver in OpenFOAM®. The power of the heating elements is decreasing with higher temperatures and is zero for the heaters maximum temperature (here approx. 80 °C). The advantage of using this temperature dependent source term, compared to using a time dependent one or a time dependent temperature boundary at heaters is, that once the temperature dependent behaviour is known, it can be used in simulations for different bed materials. First simulations (not shown here) showed that the heaters temperature did not evolve correctly over time when using a volumetric, temperature dependent heat source term according to [Section 4.2](#) at both heaters. In the first few simulated seconds, where time steps were still small, the heaters temperature in the simulation did increase very similar to the experiment. When the time step size increased, the heaters temperature in the simulation did not increase as steeply as in the experiments. Based on this finding it was decided to test the behaviour of the implemented heat source for different fixed time step sizes.

To test the time step size dependent behaviour a test case, including only the aluminium plate and the two heaters, ([Figure 29](#)) was created. The heat source term was set to a constant value $a_0 = 10^7 \text{ W/m}^3$, so the produced energy can easily be calculated according to [Equation 6.40](#). Energy loss was disabled in the simulation using zero gradient boundary conditions at all surfaces. The energy produced by two heaters in the simulation ([Equation 6.40](#)) was compared to the theoretically produced energy ([Equation 6.41](#)).

$$E_{th} = a_0 2V_{heater} t \quad (6.40)$$

where V is one heaters volume ($V = 2.02727 \cdot 10^{-6} \text{ m}^3$), t is the time in seconds and a_0 is the volume specific heat source term.

$$E_{sim} = \int T \rho c_p dV \quad (6.41)$$

where ρ is the density, c_p is the specific heat capacity and T is the cell temperature.

All in all nine simulations with different, fixed time steps have been carried out ([Table 11](#)) and the theoretical energy was compared to the produced energy for all the cases. The error was calculated according to [Equation 6.42](#) and can be seen in [Figure 30](#).

$$\Delta E = \frac{E_{th}(t) - (E_{sim}(t) - E_{sim}(t = 0))}{E_{th}(t)} 100 \quad (6.42)$$

Table 11: Time steps used in the simulation and resulting error

Case Nr.	Δt , sec	error ΔE , %
1	10^{-1}	~ 87.2 %
2	$5 \cdot 10^{-2}$	~ 76.4 %
3	10^{-2}	~ 39.4 %
4	$5 \cdot 10^{-3}$	~ 24.5 %
5	$2.5 \cdot 10^{-3}$	~ 13.9 %
6	10^{-3}	~ 6.1 %
7	$5 \cdot 10^{-4}$	~ 3.1 %
8	$2.5 \cdot 10^{-4}$	~ 1.6 %
9	10^{-4}	~ 0.65 %

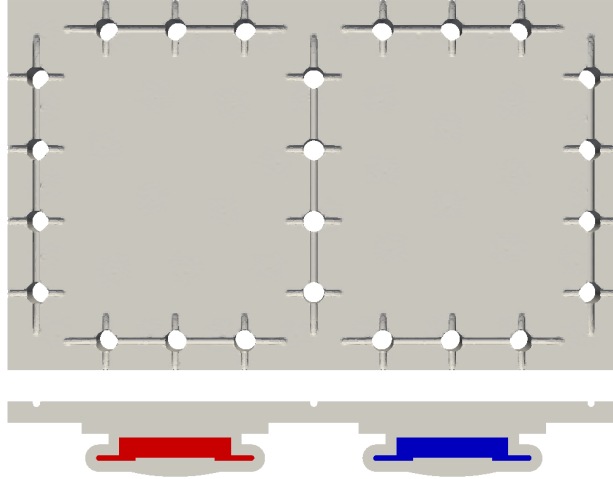


Figure 29: Test case to examine the time step size dependency of the heaters temperature when using a volumetric heat source term. Grey: Aluminium plane, colored: part of the heaters in which a time or temperature dependent, volumetric heat source term is applied.

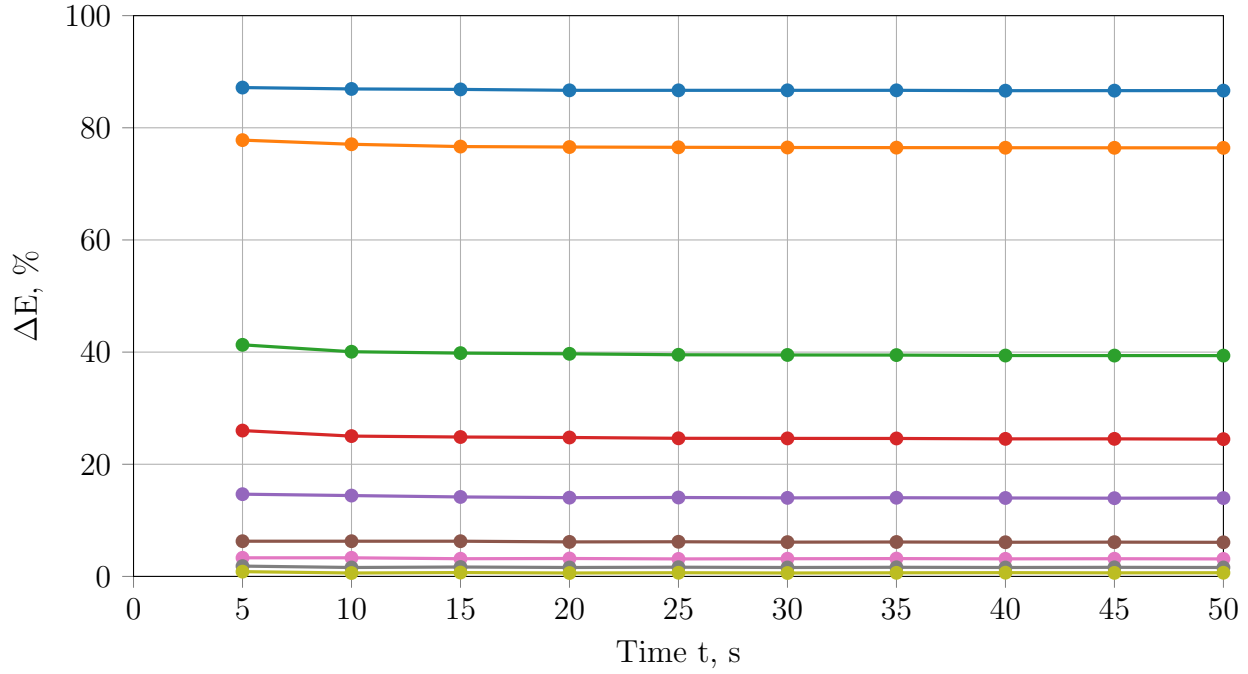


Figure 30: Relative error of the produced energy for nine different time step sizes and a constant heat source term. The time step size decreases from top to bottom (Case Nr. 1-9)

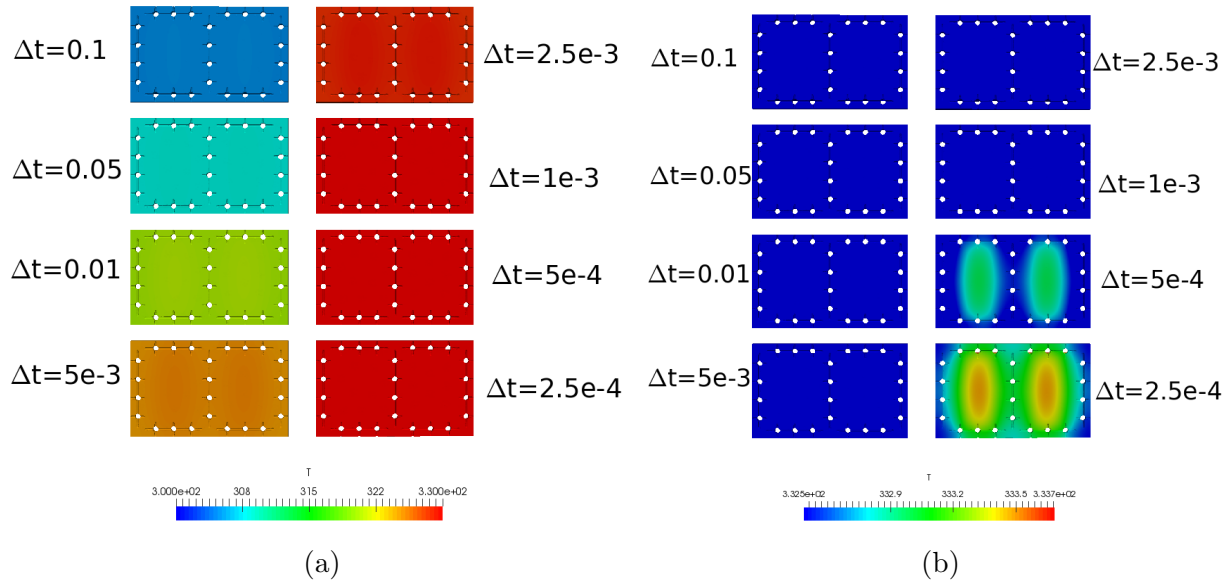


Figure 31: Surface temperature of the aluminium plate for different time step sizes (Δt in seconds) after 50 simulated seconds. (a) Temperature range: 300 – 330 K, (a) Temperature range: 332.5 – 333.7 K

The surface temperatures of the aluminium plate for the different simulated time step sizes

are shown in Figure 31. It can easily be seen that the error increases with increasing time step size.

To check if the same behaviour can be observed for simpler cases, a test case consisting of two adjacent cubes was set up. One of the cubes was heated using time dependent, constant or linear decreasing heat sources. Several different meshes (different cell sizes, expansion ratios, boundary refinement) have been tested, but only for a fine mesh (cell size 0.5 mm) a significant error in produced energy was observed. Time step sizes and the according error after 100 simulated seconds can be seen in Table 12 and Figure 32. In this case a constant volumetric heat source term of $a_0 = 10^6 \text{ W/m}^3$ and a cell sizes of 0.5 mm was used. The error in produced energy was again calculated according to Equation 6.42.

Table 12: Time steps used in the simulation of two adjacent cubes with an over all fine mesh and according error after 100 simulated seconds

Case Nr.	Δt , s	error ΔE , %
1	10^{-3}	~ 0.2 %
2	10^{-2}	~ 2.0 %
3	10^{-1}	~ 17.0 %
4	10^{-0}	~ 64.0 %

$\Delta t = 10^{-3} \text{ s}$, $\Delta E = \sim 0.2$ % $\Delta t = 10^{-1} \text{ s}$, $\Delta E = \sim 17.0$ %
 $\Delta t = 10^{-2} \text{ s}$, $\Delta E = \sim 2.0$ % $\Delta t = 10^{-0} \text{ s}$, $\Delta E = \sim 64.0$ %

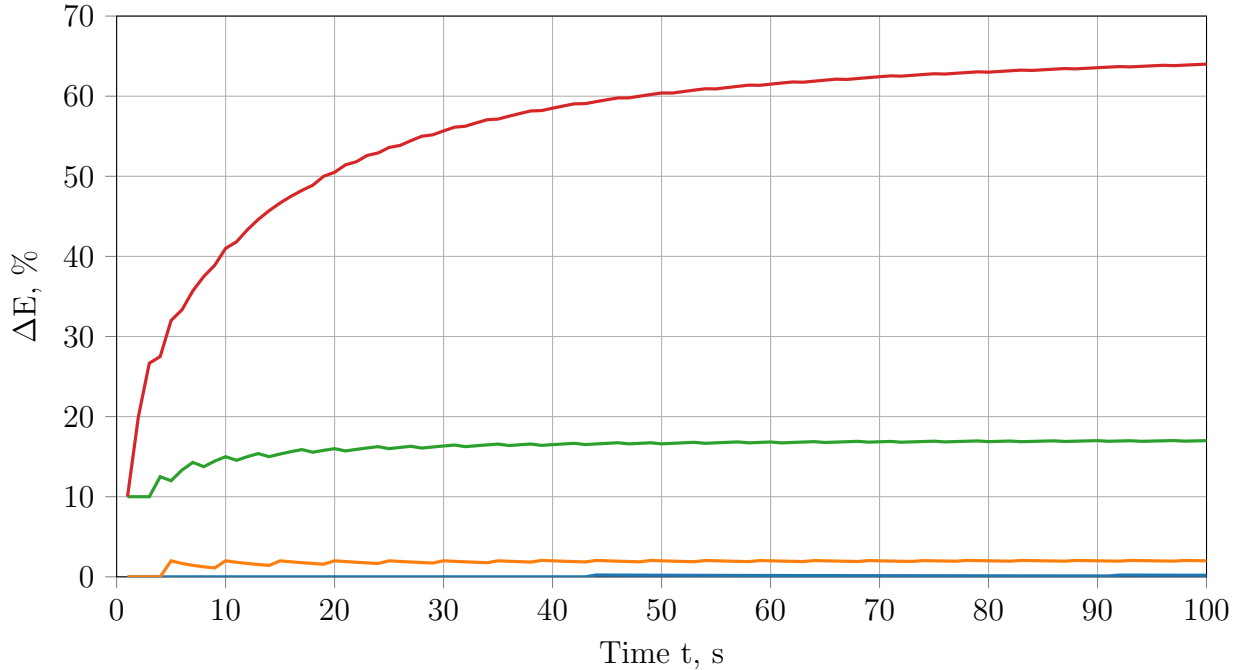


Figure 32: Relative error of the produced energy for four different time step sizes for a constant heat source term. The time step size increases from top to bottom (Case Nr. 1-4)

As no error in the implementation and application of the heat source term could be found, it was decided not to use this kind of energy source. Instead the heaters temperature is measured in each experiment and is then applied in the simulation as a time dependent temperature boundary.

6.4.4 CFD-Simulation

Two packings of randomly packed spheres with sphere diameter of 6 mm and 10 mm, were generated using an inhouse DEM-code. The number of inserted spheres was 920 and 200 for 6 mm and 10 mm spheres respectively. Exactly the same number of spheres was used in the experiments. Meshing was carried out in snappyHexMesh with mesh quality parameters described in [Section 6.4.2](#) using the multi region approach. Bridges of the size of $d/D = 0.2$, $d/D = 0.15$ and $d/D = 0.12$ have been generated using different background mesh sizes. The number of cells for the different packings and each region can be found in [Table 13](#).

Simulations were carried out in OpenFoam® version 4.1 using the unsteady, compressible *chtMultiRegionFoam* solver. As only natural convection is considered, a laminar simulation was carried out. Time step size was limited using the Courant criteria and a Courant number of 0.8. The measured heaters temperature was applied as a time dependent temperature boundary condition at heaters in the simulation for each packing. Material properties can be found in [Table 14](#).

Table 13: Number of cells for the different packings and for each region ($\cdot 10^6$)

	$D = 10 \text{ mm}$ $d/D = 0.12$	$D = 6 \text{ mm}$ $d/D = 0.2$	$D = 6 \text{ mm}$ $d/D = 0.15$	$D = 6 \text{ mm}$ $d/D = 0.12$
Bottom plate	0.472	0.136	0.290	0.473
Container	2.577	0.958	1.525	2.577
Isolation	1.680	0.629	1.026	1.680
Packing	1.567	0.640	1.385	2.175
Fluid	2.945	1.006	2.077	3.385
Total	9.243	3.370	6.305	10.292

Table 14: Material properties used in the simulation

Bottom plate (Aluminium)		
Specific heat capacity c_p	837	$J/(kg\ K)$
Thermal conductivity κ	236	$W/(m\ K)$
Density ρ	2700	kg/m^3
Container (HDPE)		
Specific heat capacity c_p	2050	$J/(kg\ K)$
Thermal conductivity κ	0.38	$W/(m\ K)$
Density ρ	1000	kg/m^3
Isolation		
Specific heat capacity c_p	2000	$J/(kg\ K)$
Thermal conductivity κ	0.03	$W/(m\ K)$
Density ρ	200	kg/m^3
Fluid (air)		
Specific heat capacity c_p	1004.5	$J/(kg\ K)$
Prandtl number Pr	0.7	-
Dynamic viscosity μ	$1.8 \cdot 10^{-5}$	$kg/(m\ s)$
Packing (Aluminium)		
Specific heat capacity c_p	837	$J/(kg\ K)$
Thermal conductivity κ	236	$W/(m\ K)$
Density ρ	2700	kg/m^3
Packing (Glass)		
Specific heat capacity c_p	1335 and/or 600	$J/(kg\ K)$
Thermal conductivity κ	1.129	$W/(m\ K)$
Density ρ	2500	kg/m^3
Packing (Steel)		
Specific heat capacity c_p	430	$J/(kg\ K)$
Thermal conductivity κ	54	$W/(m\ K)$
Density ρ	7850	kg/m^3

The fluid properties are defined using the the Prandtl number Pr . It is defined as followed:

$$Pr = \frac{\mu/\rho}{k/(c_p\rho)} = \frac{c_p \mu}{k} \quad (6.43)$$

One of the most important parameters in terms of heat transfer in packed beds, is the voidage of the packing. The voidage of meshed sphere packings for different bridge sizes is shown in [Figure 33](#). The mesh voidage is compared to theoretical voidage according to *Bey* and *Eigenberger* (1997), *deKlerk* (2003) and *Giese* (1998) shown in [Section 2.1.4](#). Ψ_∞ was varied until the integral profile of the Bey-Eigenberger-equation ([Equation 2.5](#)) did fit average voidage according to *Jeshar* ([Equation 6.44](#)). The values are $\Psi_\infty = 0.3888$ and $\Psi_\infty = 0.383$ for 10 mm and 6 mm spheres respectively. For the column diameter D the width of the bed

($D = 85 \text{ mm}$) was used. The average voidage of each voidage profile $\Psi(r)$ was calculated according to Equation 6.45 (deKlerk (2003)).

In Figure 33 and Figure 34 it can be seen that while the correlations of *Bey* and *Eigenberger* (1997) and *Giese* (1998) do under and over predict the average voidage of the packing respectively, the correlation used by *deKlerk* (2003) does fit to the average voidage of the bridged packing with $d/D = 0.12$. This is because this correlation is able to predict the voidage near the walls with high accuracy, as this correlation was developed for low container to particle diameter ratios of $N = 2 - 20$. SnappyHexMesh was not able to capture all spheres when using bridge sizes of $d/D = 0.2$, thus the voidage here is higher than for small bridges.

$$\Psi_k = 0.375 + 0.34 \frac{1}{N} \quad (6.44)$$

$$\Psi = \frac{\int_0^R \Psi(r) (2\pi r) dr}{\pi R^2} \quad (6.45)$$

- | | |
|---|--|
| — bridged (necks) packing, $d/D = 0.2$,
$\Psi = 0.457$ | — Bey and Eignenberger, $\Psi = 0.401$ |
| — bridged (necks) packing, $d/D = 0.12$,
$\Psi = 0.421$ | — de Klerk, $\Psi = 0.425$ |
| | — Giese, $\Psi = 0.441$ |

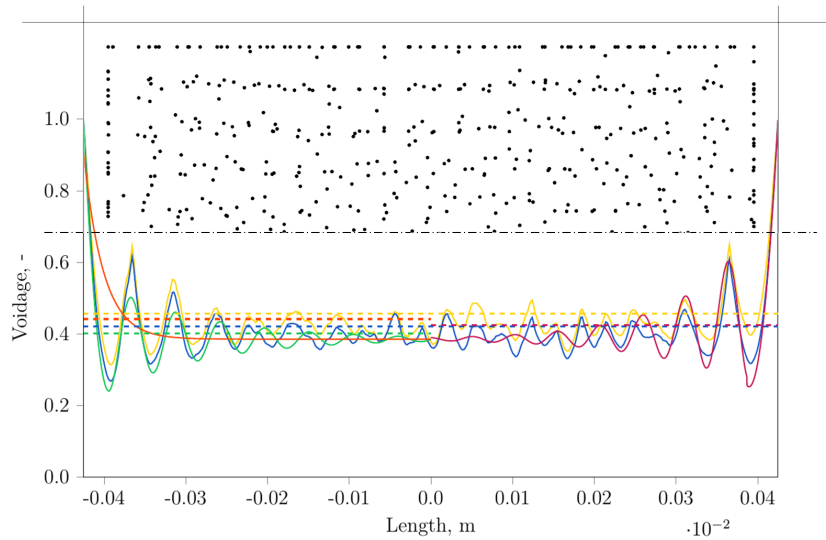


Figure 33: Voidage profile and sphere centres (top view) of a half packing of 920 spheres with diameter of 6 mm. The mesh voidage is compared to correlations discussed in Section 2.1.4. Dashed lines do represent the average bed voidage.

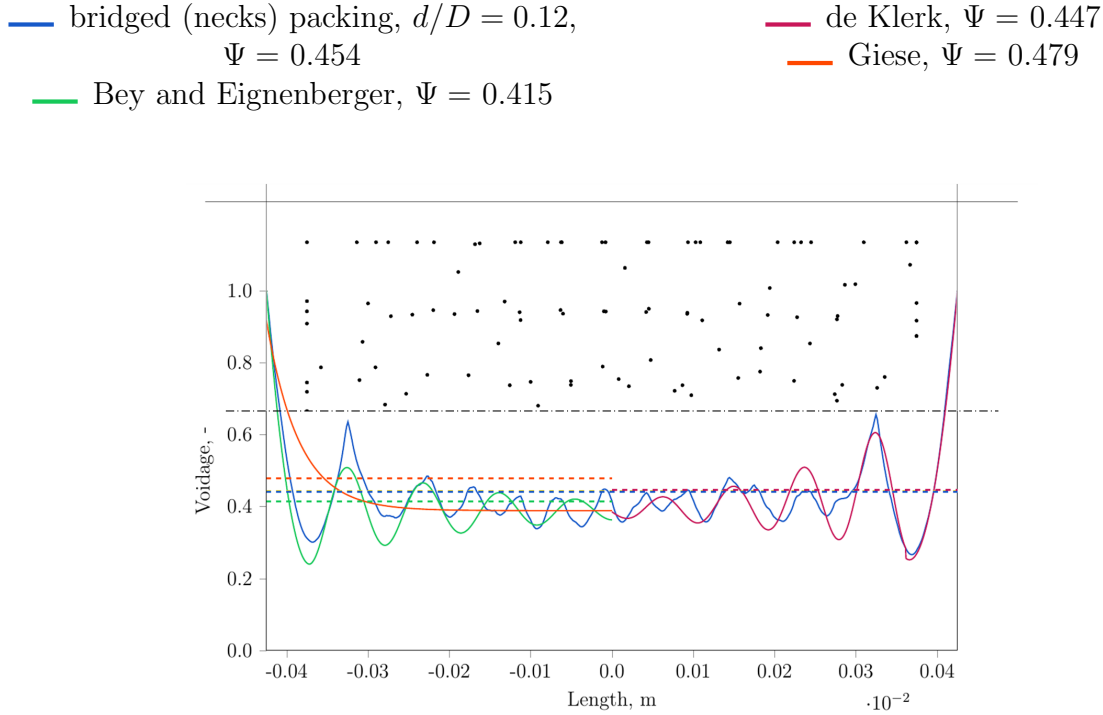


Figure 34: Voidage profile and sphere centres (top view) of a half packing of 200 spheres with diameter of 10 mm. The mesh voidage is compared correlations discussed in [Section 2.1.4](#). Dashed lines do represent the average bed voidage.

In the following sections the simulation results for different materials are discussed. The surface temperature of the experiments is compared to the results of the CFD-simulation using the necks method and the numerical results of the Zehner/Bauer/Schlünder model ([Section 2.1.5](#), [Equation 6.38](#)). Here the flattening coefficient φ was set to zero, the emission coefficient ϵ was set to 0.35 (emission coefficient for polished steel) and a height-factor x of 2 was used. For Ψ the average voidage of the deKlerk model is used. The fluid properties are $\lambda_f = 25.69 \cdot 10^{-3} \text{ W/(m K)}$, $c_{p,f} = 1006.43 \text{ J/(kg K)}$. The density of air was calculated using the perfect gas equation. Λ_{ax} was calculated as the mean of Λ_{ax} at $T = 20^\circ\text{C}$ and at $T = 70^\circ\text{C}$. It is worth to note, that for such low temperatures the values of Λ_{ax} differ by approximately 0.5% from $T = 20^\circ\text{C}$ to $T = 70^\circ\text{C}$.

The surface temperature of the simulation was calculated as the area weighted average temperature of the packing at a height of 40mm. This is exactly the position of the temperature probe in the experiments. An example for the CFD-simulation is shown in [Figure 35](#). Important solver settings for CFD simulation (OpenFoam®) are shown in [Appendix C](#).

Simulations have been carried out on the Vienna Scientific Cluster (VSC3). Due to the high number of cells, simulations have been carried out in parallel, splitting the simulation domain in 96 sub-domains. Depending on the mesh size and simulated time the simulation did take up to 36 hours to finish.

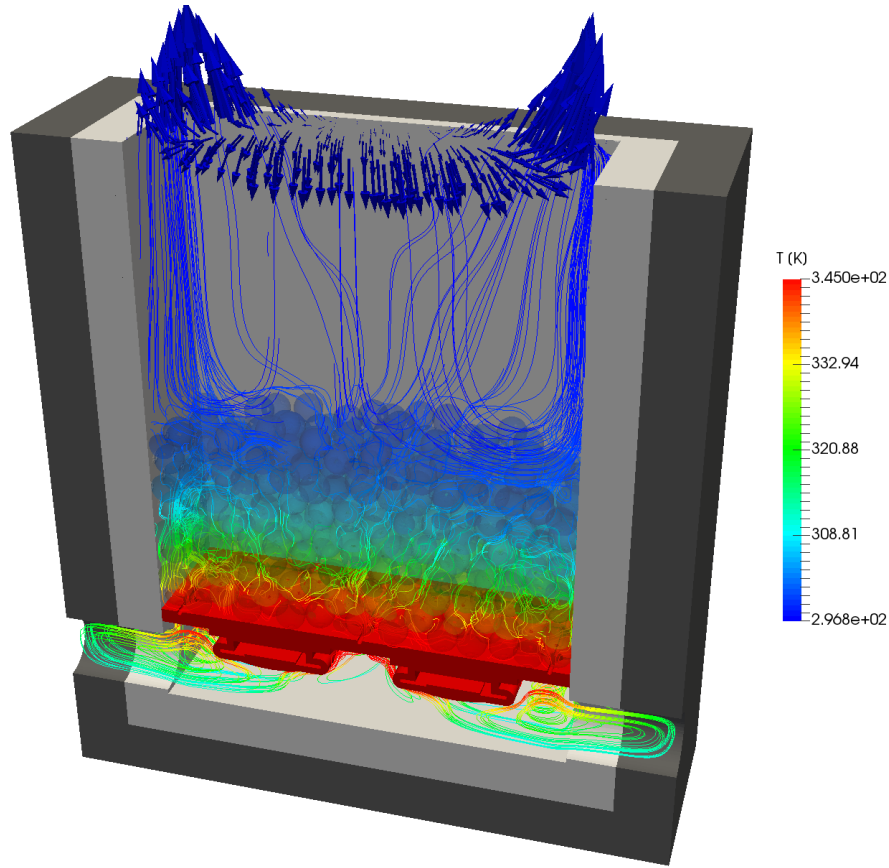


Figure 35: Temperature distribution, stream lines and velocity vectors at outlet of a packing consisting of 920 $D = 6 \text{ mm}$ glass spheres after 30 simulated minutes.

6.4.4.1 6mm Aluminium Spheres

Experimental and simulation results for $D = 6 \text{ mm}$ aluminium spheres are shown in [Figure 36](#). The simulation for necks as the method for contact point treatment, with different neck-sizes, completely overpredicts the surface temperature. This is because the main resistance for heat transfer for particle with high thermal conductivities occurs at particle-particle contact points. When using necks in the simulation, this resistance is reduced significantly. When reducing the size of the necks the results do get better, but are still far off the experimental results. On the other hand the ZBS-model gives highly accurate results. For the aluminium packing the effective thermal conductivity was calculated to $\Lambda_{ax} = 0.4955 \text{ W/(m K)}$.

The starting temperature (at $t = 0 \text{ min}$) is $T_0 = 23.3 \text{ }^\circ\text{C}$ (296.45 K). After 15 minutes the temperatures increase to $T_{exp} = 28.7 \text{ }^\circ\text{C}$ (301.85 K), $T_{ZBS} = 28.27 \text{ }^\circ\text{C}$ (301.42 K) and $T_{CFD} = 69.87 \text{ }^\circ\text{C}$ (343.02 K) in the experiment, the ZBS-model and the CFD-simulation respectively.

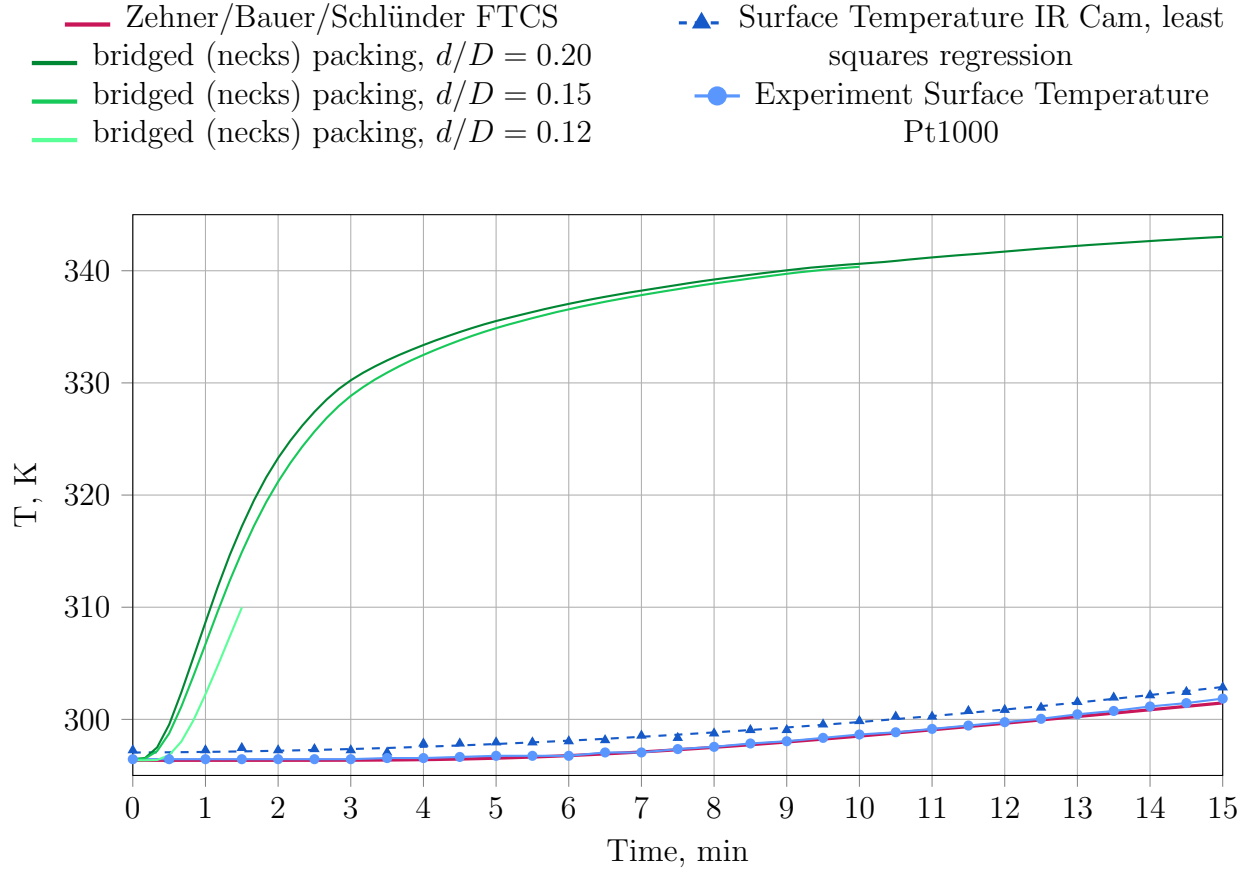


Figure 36: Surface temperature of a packed bed consisting of 920 $D = 6 \text{ mm}$ aluminium spheres. Comparison of experiments, ZBS-model and CFD-simulation. The Temperature was measured at a height of 40 mm.

6.4.4.2 6mm Glass Spheres

As already mentioned in [Section 4.4](#) the surface temperatures of the packing in the CFD simulation are lower than in the experiments, when using material properties suggested by the manufacturer ($c_p = 1335 \text{ J/(kg K)}$). When using the specific heat capacity of $c_p = 600 \text{ J/(kg K)}$, measured using a calorimeter ([Section 4.4.2](#)), the surface temperature is slightly overpredicted, but more accurate than for aluminium spheres ([Figure 37](#)). For spheres with low thermal conductivity (such as glass), the fraction of thermal resistance of contact points is lower than for high conductivities. Thus the CFD-results for the necks method is more accurate for glass spheres than for aluminium spheres.

The ZBS-model does underpredict the surface temperature in this case. This might be because of the chosen flattening coefficient of $\varphi = 0$. It is an empirical parameter and a function of many undetermined factors, such as the elasticity of the material, external mechanical stress and surface roughness. When setting $\varphi = 0$, flattening at contact points and the surface roughness is not considered. Furthermore the ZBS-model underestimates the temperature, because only conduction is considered and natural convection is neglected. For glass spheres the effective thermal conductivity was calculated to $\Lambda_{ax} = 0.1713 \text{ W/(m K)}$.

The temperatures after 26 minutes are $T_{exp} = 30.6 \text{ }^\circ\text{C}$ (303.75 K), $T_{ZBS} = 28.4 \text{ }^\circ\text{C}$ (301.5 K), $T_{CFD,1335} = 25.9 \text{ }^\circ\text{C}$ (299.11 K) and $T_{CFD,600} = 31.7 \text{ }^\circ\text{C}$ (304.89 K) for experiment, ZBS-

model, CFD-simulation 1 ($c_p = 1335 \text{ J/(kg K)}$) and CFD-simulation 2 ($c_p = 600 \text{ J/(kg K)}$) respectively. The temperature at $t = 0 \text{ min}$ is $T_0 = 23.6 \text{ }^\circ\text{C}$ (296.75 K).

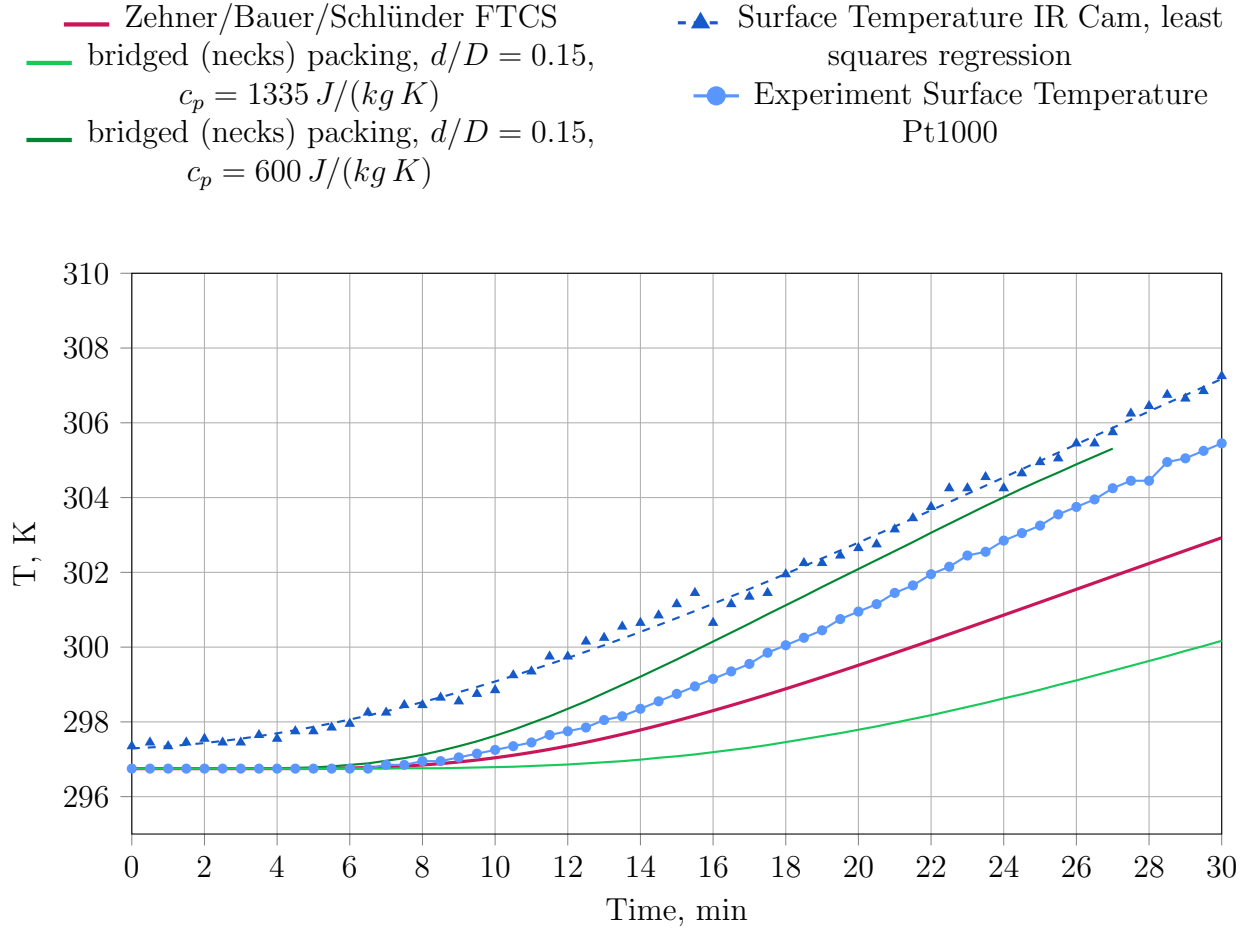


Figure 37: Surface temperature of a packed bed consisting of 920 $D = 6 \text{ mm}$ glass spheres. Comparison of experiments, ZBS-model and CFD-simulation. The Temperature was measured at a height of 40mm.

6.4.4.3 10mm Steel Spheres

For $D = 10 \text{ mm}$ steel spheres the error is in between aluminium and glass spheres, but the CFD-simulation still overpredicts the surface temperature (Figure 38). The ZBS-model underpredicts the surface temperature for the same reasons as for glass spheres. The effective thermal conductivity was calculated to $\Lambda_{ax} = 0.3882 \text{ W/(m K)}$.

The temperatures after 20 minutes are $T_{exp} = 28.9 \text{ }^\circ\text{C}$ (302.05 K), $T_{ZBS} = 26.5 \text{ }^\circ\text{C}$ (299.68 K) and $T_{CFD} = 60.8 \text{ }^\circ\text{C}$ (333.96 K) for experiment, ZBS-model and CFD-simulation. The temperature at $t = 0 \text{ min}$ is $T_0 = 24.0 \text{ }^\circ\text{C}$ (297.15 K).

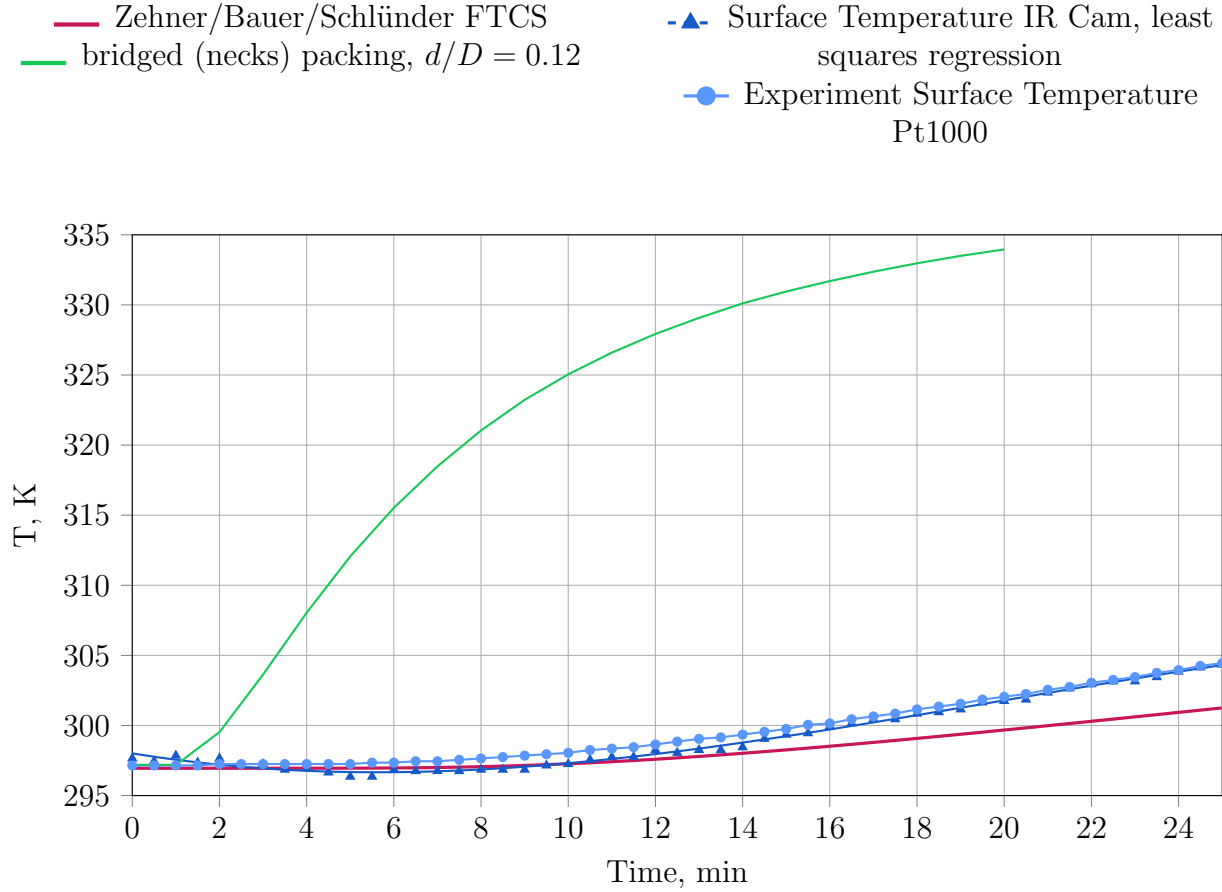


Figure 38: Surface temperature of a packed bed consisting of 200 $D = 10$ mm steel spheres. Comparison of experiments, ZBS-model and CFD-simulation. The Temperature was measured at a height of 40 mm.

6.4.5 The Necks Method: Conclusion

Three different neck-sizes have been tested for different packing materials with a wide range of thermal conductivities and compared to experiments and the semi empirical Zehner/Bauer/Schlünder-Model. In all cases the surface temperature of the packing has been overpredicted in the CFD-simulation. The trend, that the error gets smaller for smaller thermal conductivities, can be seen in [Figure 36](#)-[Figure 38](#). The reason for that is that the main thermal resistance for particles with high thermal conductivity (aluminium) occurs at particle-particle contact points. For lower thermal conductivities the share of contact points in terms of thermal resistance is lower, thus the error gets lower. Moreover it was observed that for smaller necks the simulation results do get closer to the experimental results ([Figure 36](#)).

Although the results look more promising for particles with low thermal conductivities and small bridges, based on the results shown in the last sections the necks method can not be recommended in general. Especially because the specific heat capacity of glass spheres was determined with some degree of uncertainty in [Section 4.4.2](#), there might be deviations from the results shown here for lower thermal conductivities when using the necks method.

The Zehner/Bauer/Schlünder-model in conjunction with an average voidage proposed by *deKlerk* (2003) is able to predict the effective thermal conductivity of packed beds. However it

is necessary to determine the unknown flattening coefficient φ and/or to consider convection in the α_w -Model (use of a heterogenous model) to get accurate results for different packing materials.

6.5 The bridges method

Since the necks method did not give satisfying results, especially for particles with high thermal conductivity, a different approach is studied here. In the so called bridges method, first proposed by *Ookawara et al. (2007)* and *Kuroki et al. (2007)* and later further developed by *Dixon, Nijemeisland, and Stitt (2013)*, cylindrical bridges with a given diameter d are inserted at contact points. An effective thermal conductivity k_{eff} , depending on height and width of the bridge and the thermal conductivity of the particle and the fluid, is assigned to the cylinders, while the particles specific heat capacity c_p and density ρ_p used.

Bridges are inserted using the following procedure (see also [Figure 19](#)):

1. Create the packing using the inhouse DEM-code
2. Load the packing into the geometry module (GEOM) of SALOME
3. Create bridges with given diameter at small gaps/contact-points
4. Subtract bridges from packing to get capped spheres
5. Create surface mesh of packing and bridges using SALOME's mesh module (SMESH)
6. Export meshed packing and bridges as STL-file
7. Mesh the bridged packing in snappyHexMesh using the multi region approach

In step three bridges are only inserted between spheres when a certain contact distance deceeds a given value. To ensure a successful boolean operation in step 4, the sphere diameter had to be reduced by 0.05%. The procedure of inserting the bridges is shown in [Figure 39](#). The python-code for inserting cylindrical bridges at contact points in SALOME is presented in [Appendix F](#).

The surface mesh representation of the bridged packing can be seen in [Figure 40](#). Meshing is carried out using parameters shown in [Section 6.4.2](#) using the multi-region approach. Instead of uniform refinement level on surfaces, here the *gapLevelIncrement*- function of *snappyHexMesh* is used. To avoid convergence problems in non-orthogonal cells, cell limited and non-orthogonal corrected schemes are used for gradient, divergence, laplacian and surface normal gradient discretization in simulations. The number of cells for each of the four different meshes, is presented in [Table 16](#). For $D = 6 \text{ mm}$ particles and $d/D = 0.15$ bridges only a quarter of the geometry is simulated, to get reasonable cell count. [Table 17](#) shows the number of contact points for the different packings.

Material properties are the same as for the necks method ([Table 13](#)).

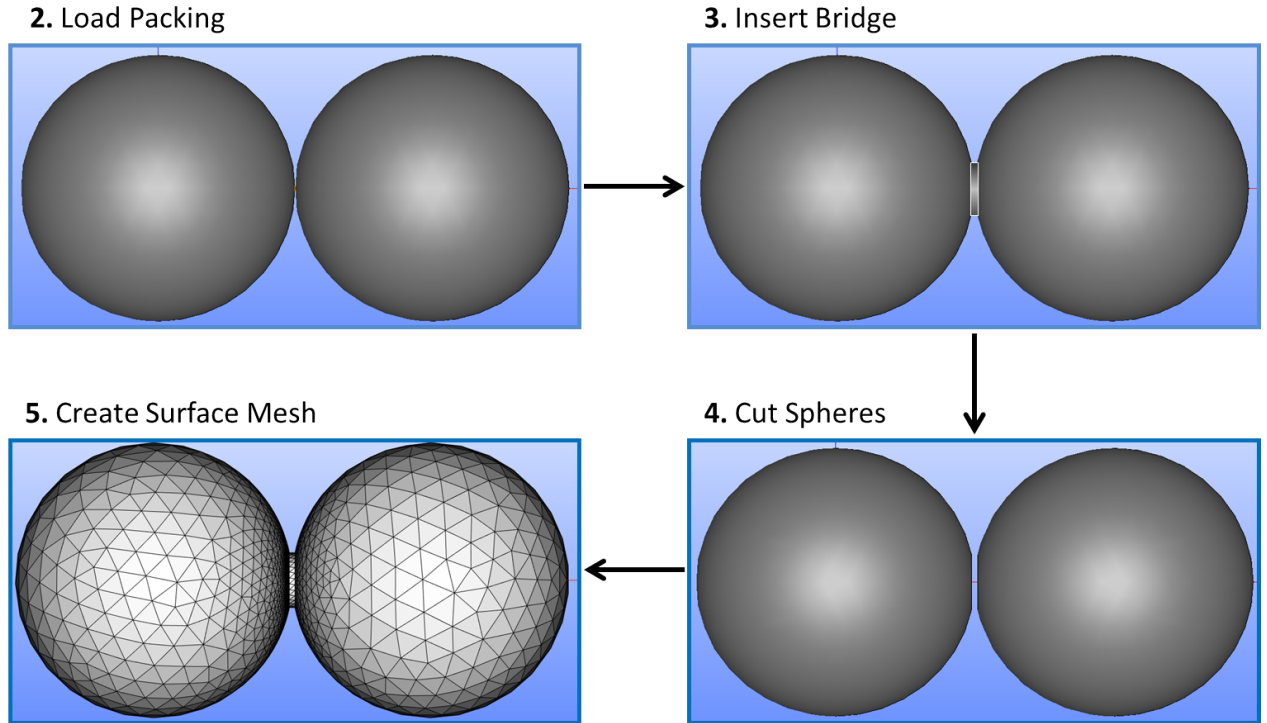


Figure 39: Bridging procedure in SALOME

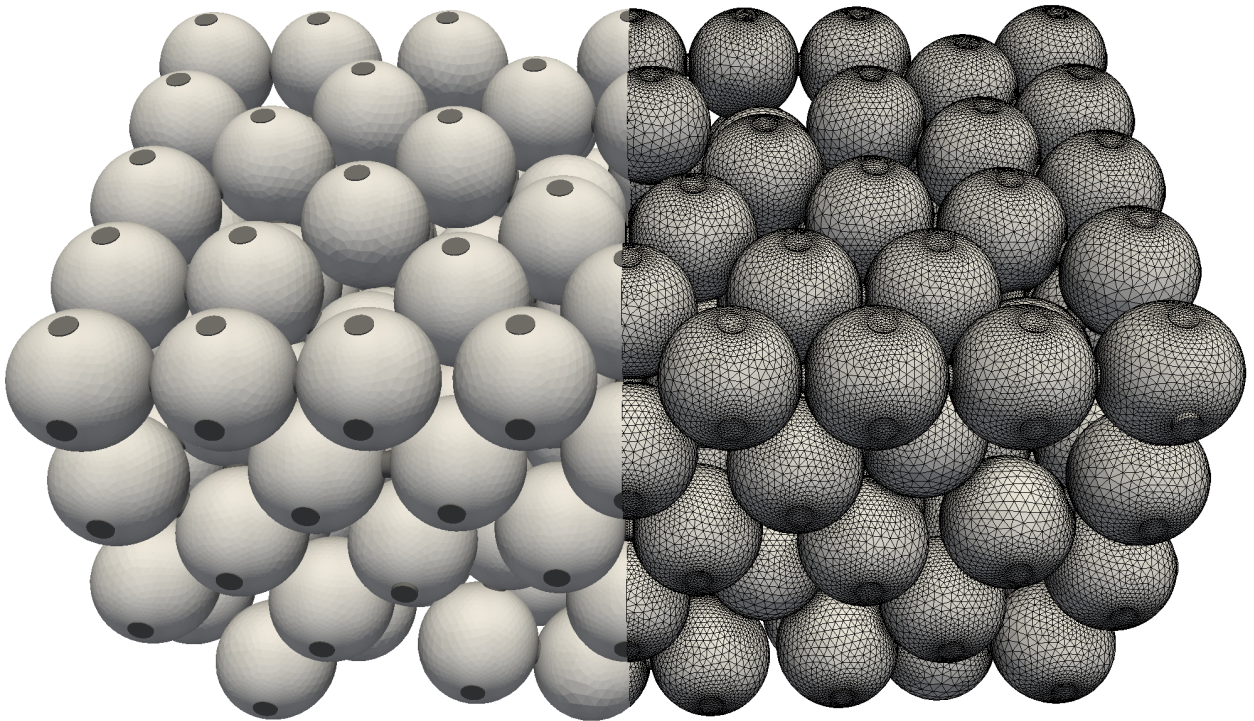
Figure 40: STL-surface representation of 200 $D = 10 \text{ mm}$ spheres with $d/D = 0.2$ bridges at particle-particle and particle-wall contact points.

Table 16: Number of cells for the different packings and for each region ($\cdot 10^6$)

	$D = 10 \text{ mm}$ $d/D = 0.15$	$D = 10 \text{ mm}$ $d/D = 0.20$	$D = 6 \text{ mm}$ $d/D = 0.15$	$D = 6 \text{ mm}$ $d/D = 0.20$
Bottom plate	1.690	0.892	0.669	0.896
Container	6.486	3.894	1.787	3.009
Isolation	1.908	1.908	0.678	0.838
Packing	17.507	11.238	9.113	13.294
Bridges	1.676	0.312	3.973	13.294
Fluid	15.954	10.737	6.584	6.571
Total	50.313	31.798	22.807	31.415

Table 17: Number of particle-particle and particle-wall contact points for the two created packings

	$D = 10 \text{ mm}$	$D = 6 \text{ mm}$
Nr. of spheres	200	619
part.-part. contacts	508	1758
part.-side wall contacts	93	184
part.-bottom wall contacts	40	118
Total	614	2060

6.5.1 Influence of the Bridge Size on the Bed Voidage

The voidage profile of a meshed packing consisting of 619 $D = 6\text{mm}$ spheres with different bridge sizes is shown in [Figure 41](#). For small bridges the average voidage is smaller as for $d/D = 0.2$ bridges. This is because only a quarter of the geometry was meshed. However, as the deviation is less than 1 %, no significant influence on the heat transfer is expected.

The influence of different bridge sizes on the voidage profile and the average voidage can be seen in [Figure 42](#). Apparently the change of the bridge size from $d/D = 0.15$ to 0.2 has very little influence.

As the mesh voidage does not substantially change from the neck method to the bridges method, the same statements as in [Section 6.4.4](#) are true when comparing to the correlations of *Bey and Eigenberger* (1997), *deKlerk* (2003) and *Giese* (1998). For 10 mm spheres the *de Klerk*-correlation gives the best results for the average voidage, while for 10 mm sphere the correlation of *Bey and Eigenberger* and *de Klerk* shows nearly the same error, where the error of the first mentioned is slightly lower. However the *de Klerk*-correlation gives in both cases, 6mm and 10mm spheres, better match in the voidage profile.

The voidage according to *Giese* (1998) gives a to high average voidage in all cases.

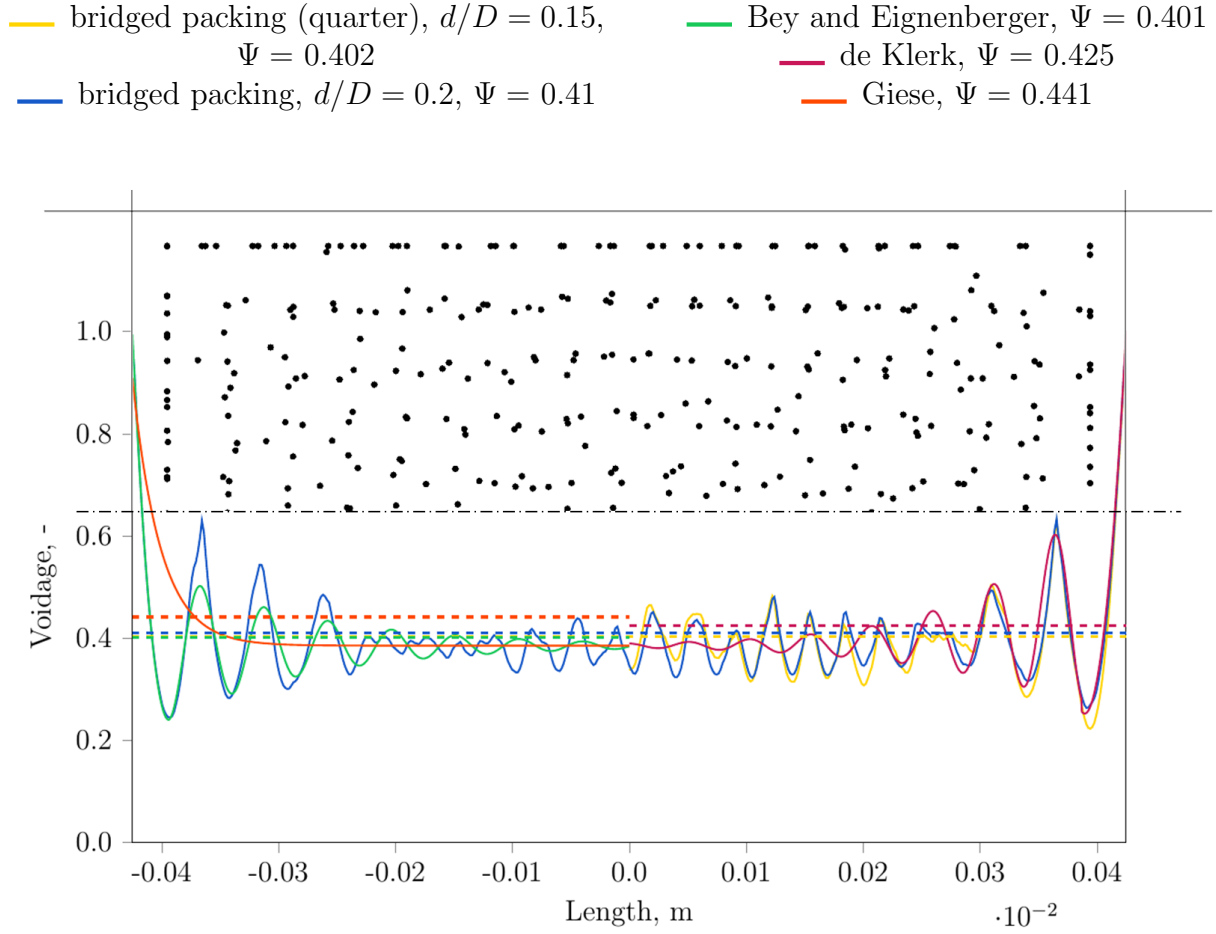


Figure 41: Voidage profile and sphere centres (top view) of a half packing of 619 spheres with diameter of 6 mm, with bridge sizes of $d/D = 0.15$ and $d/D = 0.2$. The mesh voidage is compared to equations discussed in [Section 2.1.4](#). Dashed lines do represent the average bed voidage.

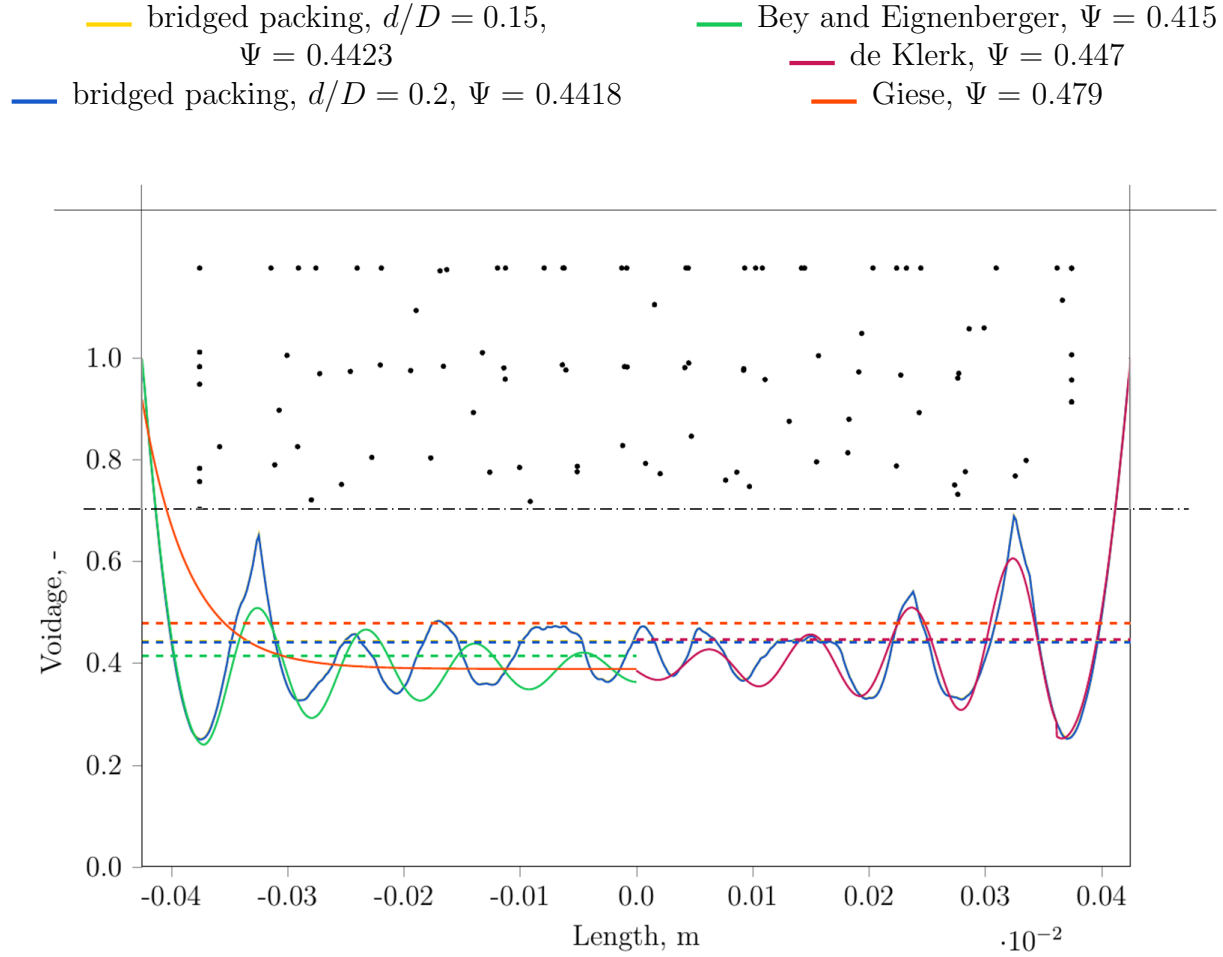


Figure 42: Voidage profile and sphere centres (top view) of a half packing of 100 spheres with diameter of 10 mm, with bridge sizes of $d/D = 0.15$ and $d/D = 0.2$. The mesh voidage is compared to discussed in [Section 2.1.4](#). Dashed lines do represent the average bed voidage.

6.5.2 Fractional Factorial Screening Design

In this study three different particle sizes ($D = 6\text{ mm}$, $D = 8\text{ mm}$ and $D = 10\text{ mm}$) and three different materials with a wide range of thermal conductivities are used.

- Glass: $k_{glass} = 1.129\text{ W/(mK)}$, $\rho_{glass} = 2500\text{ kg/m}^3$, $c_{glass} = 600\text{ J/(kgK)}$
- Steel: $k_{steel} = 54\text{ W/(mK)}$, $\rho_{steel} = 7850\text{ kg/m}^3$, $c_{steel} = 430\text{ J/(kgK)}$
- Aluminium: $k_{alu} = 236\text{ W/(mK)}$, $\rho_{alu} = 2700\text{ kg/m}^3$, $c_{alu} = 837\text{ J/(kgK)}$

Based on the findings in [Section 2.2](#) and [Section 6.4](#) it was decided that the best way to treat contact points for packed beds, consisting of mono-sized spherical particles with stagnant fluid (natural convection only), is to insert cylindrical bridges with adjustable thermal conductivity according to [Equation 2.27](#). To get the optimal bridge sizes d/D and effective thermal conductivities k_{eff} for this range of particle sizes and materials, Design of Experiments is used. In addition to the four factors (particle material, particle size D , bridge size d/D and a multiplier k for the bridges thermal conductivity k_{eff} , [Equation 6.46](#)) an appropriate

response had to be defined. Since the simulations are compared to experiments (Section 4.6), the residuals of the packing's surface temperature at a given point ($\Delta T_{exp} = 4.5 K$, Equation 6.47), were selected as the systems response y . Here the residual is defined as the difference of the measured and the simulated surface temperature. Thus a residual of zero means that the simulation perfectly predicts the surface temperature at a given time. A negative or positive residuum refers to a under or over prediction of the surface temperature, respectively. The first step was to find out which of these factors do have an influence on the response. Based on the outcome of the screening design, an optimization design can be used to get the optimum levels of the remaining factors.

$$k_{eff,mod} = k \cdot k_{eff} \quad (6.46)$$

$$Response \Delta T = T_{exp}(t) - T_{sim}(t) \quad (6.47)$$

at

$$\Delta T_{exp} = T_{exp}(t) - T_{exp}(t = 0) = 4.5 K \quad (6.48)$$

In screening only few experiments have to be carried out, but a lot of different factors can be varied. Here usually fractional factorial designs are used. Since in this study only four factors are varied, a 2^{4-1} two level fractional factorial design is sufficient. For the generation and analysis of the designs, the software package *R* and the GUI *RStudio* was used. The worksheet representation of the screening design, inclusive response, is shown in Table 19.

Since the heaters temperature at $t = 0$ min is used as initial temperature for the simulation, and an initial offset of heaters temperature to the measured surface temperature of the packing is possible, the measured surface temperature was corrected to fit the initial simulation temperature. The R-code of the Design of Experiments in R is shown in Appendix G.

It is worth to note that the calculated effective thermal conductivity in this work does deviate from the values calculated by *Dixon, Nijemeisland, and Stitt (2013)*. This is because *Dixon, Nijemeisland, and Stitt (2013)* used fixed values for γ ($= 0.1$) and Λ ($= 0.07 \cdot 10^{-6} m$). Especially the accommodation coefficient of 0.1 does not agree with the accommodation coefficient calculated according to Equation 2.3 b ($\gamma = 0.899$ for $T = 300 K$). Thus the effective thermal conductivity is higher in this work, as shown in Table 18.

Table 18: Comparison of the accommodation coefficient γ , the mean free path of the gas molecules Λ and the resulting effective thermal conductivity of cylindrical bridges between spherical particles, used from *Dixon, Nijemeisland, and Stitt (2013)* and from this work. k_{eff} is calculated for $D = 10 mm$ aluminium spheres and a bridge size of $d/D = 0.2$

	$\gamma, -$	Λ, m	$k_{eff}, W/(m K)$
<i>Dixon, Nijemeisland, and Stitt (2013)</i>	0.1	$0.7 \cdot 10^{-7}$	0.09515
this work	0.899	$2.7 \cdot 10^{-7}$	0.13061

Again simulations have been carried out in parallel, splitting the simulation domain in up to 480 sub-domains and have been carried out on the Vienna Scientific Cluster (VSC3). Simulation did run up to 84 hours, depending on the mesh size and simulated time.

Table 19: 2^{4-1} screening design for four factors and the response ΔT . $k_{eff,mod}$ is the with k scaled effective thermal conductivity k_{eff} of the bridges.

Run	D, mm	d/D	k	Material	$k_{eff,mod}$, W/(m K)	Response ΔT , K
0	6	0.15	0.5	Glass	0.042195	-0.27
1	10	0.15	0.5	Aluminium	0.057747	-3.21
2	6	0.20	0.5	Aluminium	0.058778	-2.39
3	10	0.20	0.5	Glass	0.046978	-1.3
4	6	0.15	1.5	Aluminium	0.153759	0.69
5	10	0.15	1.5	Glass	0.134393	-0.33
6	6	0.20	1.5	Glass	0.135565	1.51
7	10	0.20	1.5	Aluminium	0.195921	-0.22

All screening plots can be found in [Appendix D](#). Two representative results are shown in figure [Figure 43](#). It can be seen that, while the error for aluminium spheres with $k = 1.5$ is small, for glass spheres the error is nearly zero for smaller effective thermal conductivities for bridges ($k = 0.5$). This indicates that the material properties used in the simulation slightly deviate from the real properties.

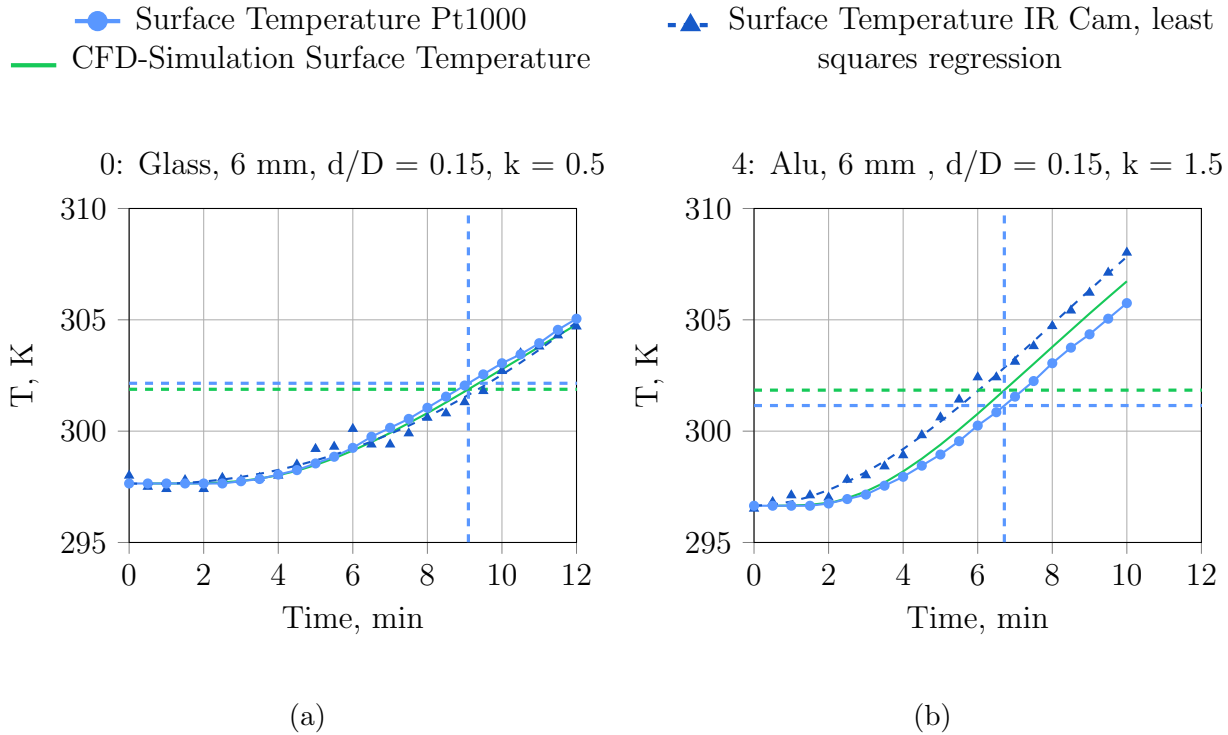


Figure 43: Comparison of CFD-simulation results and experimental surface temperature for 6mm (a) glass and (b) aluminium spheres. The response for screening is the temperature difference of the two dashed horizontal lines (at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K).

From the histogram plot [Figure 44b](#) it is not clear if the response is normally distributed, so

in addition a Shapiro-Wilk Normality Test test is performed. The null hypothesis of this test is that the sample came from a normally distributed population. Thus, if the probability (p) value is less than the chosen significance (α) level, then the zero hypothesis is rejected. Then there is evidence that the data tested are not from a normally distributed population. If the p-value is greater than the chosen α level, then the zero hypothesis that the data came from a normally distributed population cannot be rejected. For an α -level of 0.05, a data set with a p-value of 0.02 rejects the null hypothesis that the data are from a normally distributed population (*JMP®* (2004), *Razali and Wah* (2011), *Wikipedia* ([Shapiro–Wilk test](#))).

In [Listing 1](#) the output of the *Shapiro-Wilk* function of *R* is presented. The p-value is high enough to assume that the response came from a normally distributed population.

In [Figure 44a](#) it can be seen that the surface temperature is clearly underpredicted for lower k_{eff} (Run 0 - 3). In Run 4 - 7, where a factor of $k = 1.5$ is used to modify the effective thermal conductivity of bridges, the error is small compared to Run 0 - 3. But both 6 mm glass sphere cases (Run 0 and Run 6), do not follow this pattern.

To evaluate which of the factors has a significant influence on the systems response, an interaction model is fitted to the response of the screening design. Using an interaction model, all main effect can be calculated, because three factor interactions are considered to be neglectable. Two factor interactions are confounded with each other.

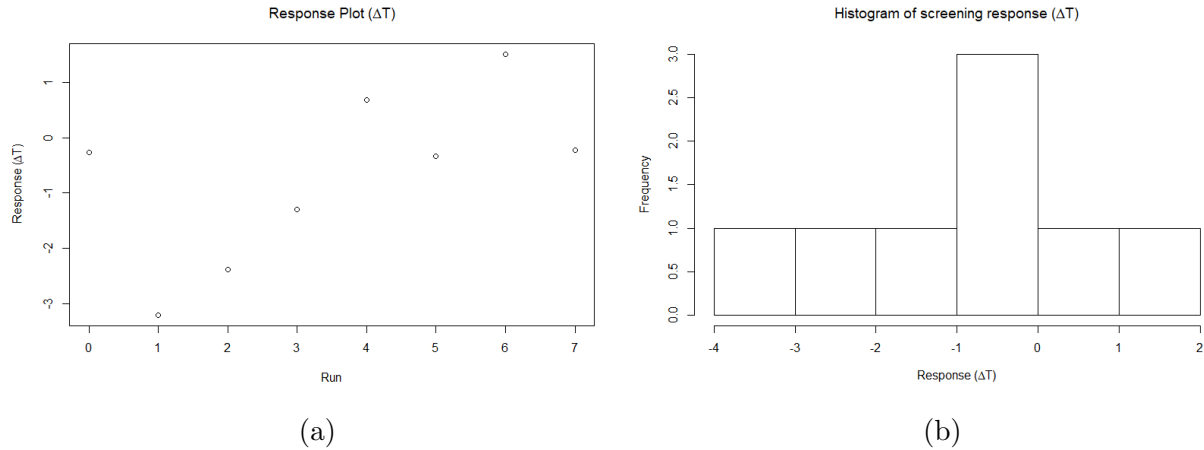


Figure 44: Raw data evaluation: (a) Response plot; (b) Histogram plot

```
> shapiro.test(y)
Shapiro-Wilk normality test

data:  y
W = 0.9584, p-value = 0.7947
```

Listing 1: Output of the Shapiro-Wilk function of *R*

As for a 2^{4-1} fractional factorial design no ANOVA can be performed, a half normal plot is used to find significant factors. If all the coefficients are caused by white noise in the response variable (i.e. no factor has an effect on the response), then the resulting coefficients are normal distributed around a mean value, therefore they are close to the near-zero line in

the half normal plot. All coefficients that are not only caused by white noise are typically displaced well off the line. Figure 45 reveals that none of the factors can be considered as significant, indicated by the fact that all the points are below the dashed line. In addition to that, again, it is not clear if the measured heat capacity c_{glass} and the thermal conductivity k_{glass} are reliable and of the same value for 6 mm and 10mm glass spheres.

Wrong values of the response do influence the outcome of the DoE and can lead to unusable results. Thus it was decided to not consider glass in the DoE, but to carry out a 2^3 full factorial design for aluminium spheres.

Confounding of factors :

D = dD:k:material

dD = D:k:material

k = D:dD:material

material = D:dD:k

D:dD = k:material

D:k = dD:material

D:material = dD:k

Listing 2: Confounding of factors in the fractional factorial screening design

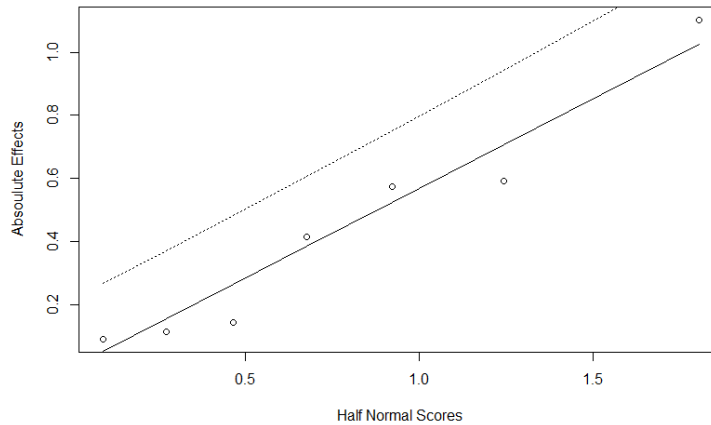
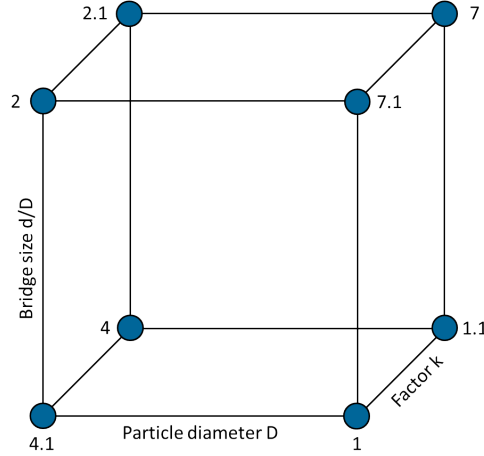


Figure 45: Half normal plot of the interaction model

6.5.3 Full Factorial Screening Design

Using a 2^3 full factorial design, it is possible to estimate interaction models. The geometrical representation of the design is shown in Figure 46, the worksheet representation with response in Table 20. It is easily accessible by replacing the glass spheres in the design used in Section 6.5.2. Comparison of experimental and simulation results are presented in Appendix D. Unsurprisingly, when thinking of the results shown in Section 6.5.2, the response plot Figure 47a shows small errors for $k = 1.5$ (Run 4 - 7), while for $k = 0.5$ (Run 0 - 3) the deviation from experiment to simulation is high. Smaller particle show higher surface temperatures in the simulation than 10 mm spheres, for the same bridge size and k -value. For 10 mm spheres the surface temperature is in all cases underpredicted.

Figure 46: Geometrical representation of the 2^3 full factorialTable 20: 2^3 screening design for three factors and the response ΔT . $k_{eff,mod}$ is the with k scaled effective thermal conductivity k_{eff} of the bridges.

Name	Run	D, mm	d/D	k	$k_{eff,mod}$, W/(m K)	Response ΔT , K
4.1	0	6	0.15	0.5	0.051253	-2.08
1	1	10	0.15	0.5	0.057747	-3.21
2	2	6	0.20	0.5	0.058778	-2.39
7.1	3	10	0.20	0.5	0.065307	-3.49
4	4	6	0.15	1.5	0.153759	0.69
1.1	5	10	0.15	1.5	0.17324	-0.45
2.1	6	6	0.20	1.5	0.176335	1.48
7	7	10	0.20	1.5	0.195921	-0.22

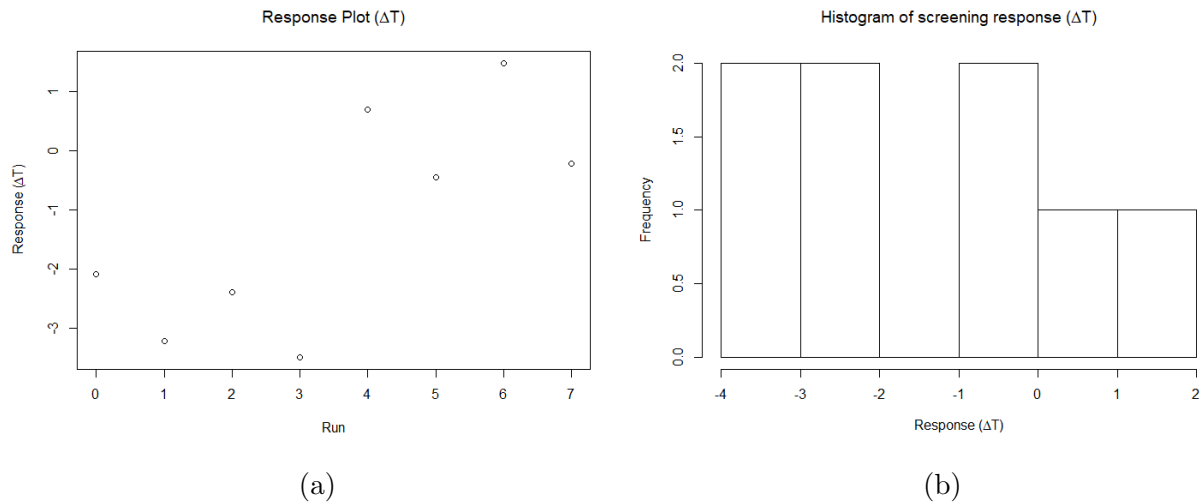


Figure 47: Raw data evaluation: (a) Response plot; (b) Histogram plot

The histogram plot is presented in [Figure 47b](#). Again it is not clear if the response is normally distributed, so in addition a Shapiro-Wilk Normality test is performed. As the p-value here is higher than any reasonable α -level ([Listing 3](#)), the null hypothesis cannot be rejected. Thus the response ΔT is likely to come from a normally distributed population and a interaction model can be fitted to the data. The model, the values of each factor, the residuals of the model, the ANOVA (evaluates the global model quality), which is automatically performed by *R*, and the *R* and *R_{adjusted}* values are presented in [Listing 4](#). The interaction model is shown in [Equation 6.49](#), where D, d/D and k are the coded factor levels (-1, +1 for high and low value) and *D1*, *dD1*, *k1*, *D1 : dD1*, *D1 : k1* and *dD1 : k1* are the main effects and interaction coefficients.

```
> shapiro.test(y)
Shapiro-Wilk normality test

data:  response
W = 0.93512, p-value = 0.5638
```

Listing 3: Output of the Shapiro-Wilk function of R

```
lm.default(formula = y ~ (.)^2, data = screeningFFD)

Residuals:
0      1      2      3      4      5      6      7
0.07375 -0.07375 -0.07375  0.07375 -0.07375  0.07375  0.07375 -0.07375

Coefficients:
              Estimate      Std. Error  t value    Pr(>|t|)
(Intercept) -1.20875      0.07375    -16.390    0.0388 *
D1           -0.63375      0.07375     -8.593    0.0738 .
dD1           0.05375      0.07375      0.729    0.5991
k1            1.58375      0.07375     21.475    0.0296 *
D1:dD1       -0.06625      0.07375     -0.898    0.5341
D1:k1        -0.07625      0.07375     -1.034    0.4894
dD1:k1        0.20125      0.07375      2.729    0.2236

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2086 on 1 degrees of freedom
Multiple R-squared:  0.9982, Adjusted R-squared:  0.9872
```

Listing 4: Summary of most important values of the fitted linear interaction model.

$$\Delta T = (\text{Intercept}) + D1 \cdot D + dD1 \cdot d/D + k1 \cdot k + D1 : dD1 \cdot D \cdot d/D + D1 : k1 \cdot D \cdot k + dD1 : k1 \cdot d/D \cdot k \quad (6.49)$$

From the main effects plot [Figure 48a](#) it can be seen that increasing particle diameter and k-value have a negative and positive influence on the temperature deviation respectively, while the bridge size has only small influence on the temperature. Parallel lines in [Figure 48b](#) indicate that the corresponding two factor interaction does not influence the systems response. Thus the only interaction factor might have an influence is the interaction of k and the

bridge size. This time the assumption is verified using a normal plot (also: Daniel Plot), which essentially shows the same as the half normal plot. From Figure 49 it can be seen that the factors D and k are significant (significance level is 0.05). But it can also be seen that the data in the middle of the plot appear as two lines, not as one straight lines. This indicates that one (or more) of the data points (responses) is influencing the effects upwards and downwards and can be considered as outlier (*Box* (1991)).

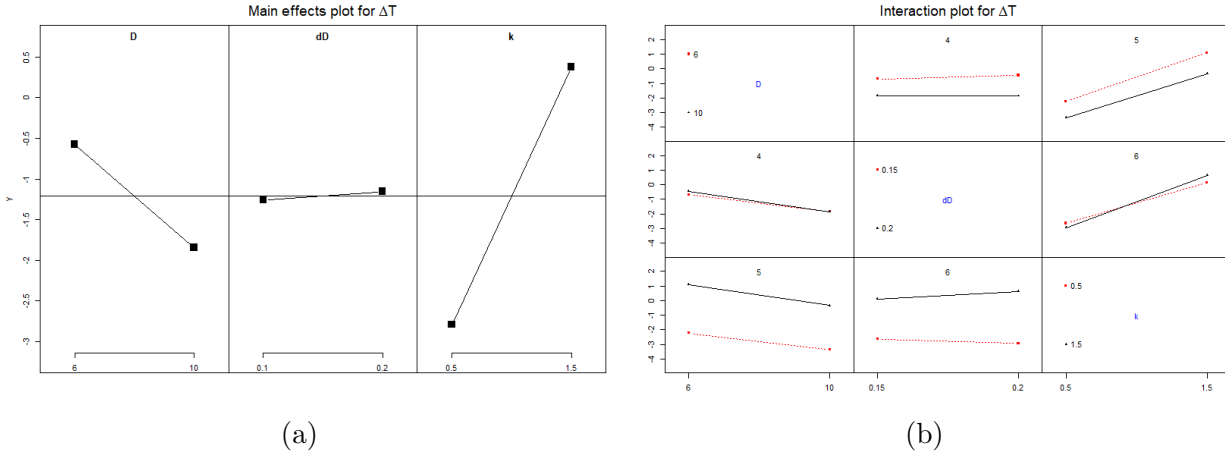


Figure 48: Model evaluation: (a) Main effects plot; (b) Interaction plot

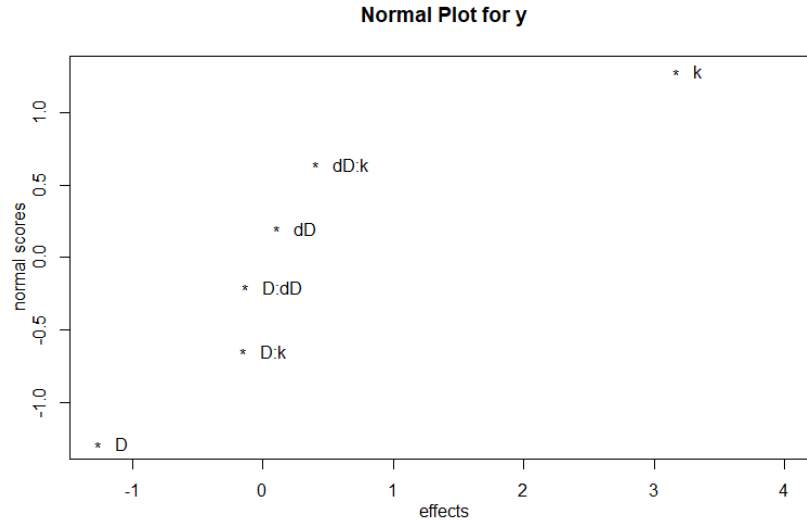


Figure 49: Normal effect plot of the interaction model. D and k are placed well of the middle, and therefore considered as significant.

To detect outliers a gap test is performed, using the in *R* implemented Daniel's Method (an example for this method on a 2^4 factorial design is presented in *Box* (1991)). With this method run 6 is detected as an outlier. Part of the output of the *Gaptest()*-function is

presented in Listing 5. After the gap correction in addition to A ($= D$) and C ($= k$) the two factor interaction B:C ($= dD : k$) can also be considered as a influencing factor, with a significance of 0.05 (Figure 50b, Listing 5). However, when thinking of the way the response is defined, it does not make any sense to treat run 6 as an outlier. The response is calculated as difference in surface temperature of experiment and simulation. Since only one experiment per sphere size has been carried out, the experiment can not influence the response in a sense that only run 6 is an outlier. As only simulation parameter are changed, run 6 shows an unusual high (compared to run 5 and 7) temperature difference, because of the interaction of the effective thermal conductivity in the bridge and the bridge size, and not because of an error in the response.

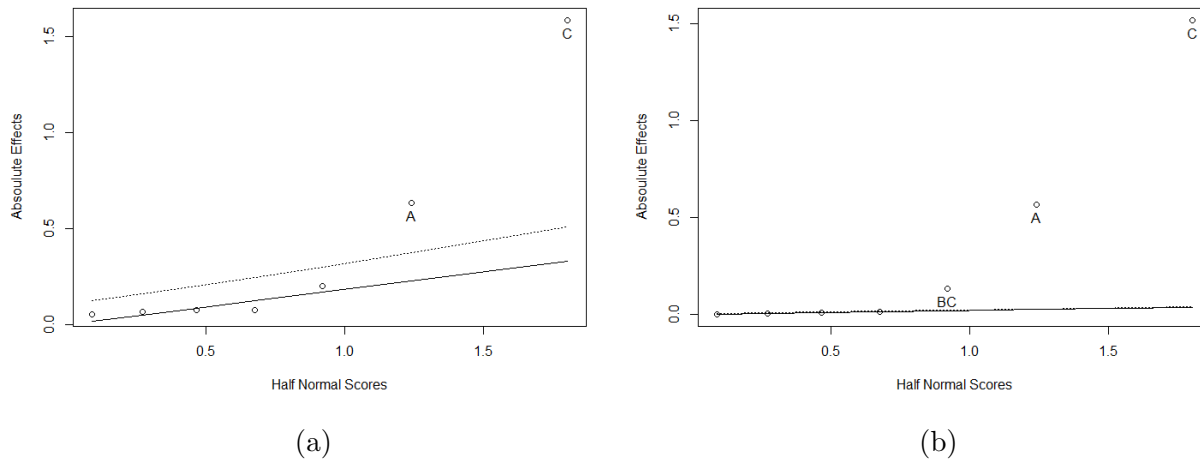


Figure 50: (a) Half normal plot, factors A (particle size D) and C (the multiplier k for modifying the effective thermal conductivity k_{eff}) are considered as significant; (b) Gap corrected half normal plot, in addition to A and C now the two factor interaction B:C (bridge size d/D and k) can be considered as significant.

```
> Gaptest(screening_codedFFD)
Effect Report
Label      Half Effect      Sig(.05)
A          -0.63375         yes
B           0.05375         no
C          1.58375         yes
AB         -0.06625         no
AC         -0.07625         no
BC           0.20125         no
ABC        -0.07375         no
```



```
Corrected Data Report
Response   Corrected Response   Detect Outlier
-2.08      -2.08                no
-3.21      -3.21                no
-2.39      -2.39                no
-3.49      -3.49                no
0.69        0.69                no
-0.45      -0.45                no
```

1.48	0.94	yes
-0.22	-0.22	no
Effect Report		
Label	Half Effect	Sig (.05)
A	-0.56625	yes
B	-0.01375	no
C	1.51625	yes
AB	0.00125	no
AC	-0.00875	no
BC	0.13375	yes
ABC	-0.00625	no

Listing 5: Gap test results; using automated Daniel's Method to detect an outlier

In addition to the full interaction model, a linear model containing only the particle diameter D and the modifier k , was fitted to the experimental results, as only these two factors are significant according to [Figure 49](#). Coefficients, residuals, R and $R_{adjusted}$ values are presented in [Listing 6](#). The model is shown in [Equation 6.50](#), where again coded factor levels have to be used for D and k . The residual plot is presented in [Figure 51](#).

Call:

lm.default(formula = y ~ D + k, data = screeningFFD)

Residuals:

0	1	2	3	4	5	6	7
0.07875	0.21625	-0.23125	-0.06375	-0.31875	-0.19125	0.47125	0.03875

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-1.2087	0.1087	-11.124	0.000102	***
D1	-0.6337	0.1087	-5.833	0.002095	**
k1	1.5837	0.1087	14.576	2.74e-05	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3073 on 5 degrees of freedom

Multiple R-squared: 0.9801, Adjusted R-squared: 0.9722

Listing 6: Summary of most important values of the fitted linear model containig only D and k .

$$\Delta T = (Intercept) + D1 \cdot D + k1 \cdot k \quad (6.50)$$

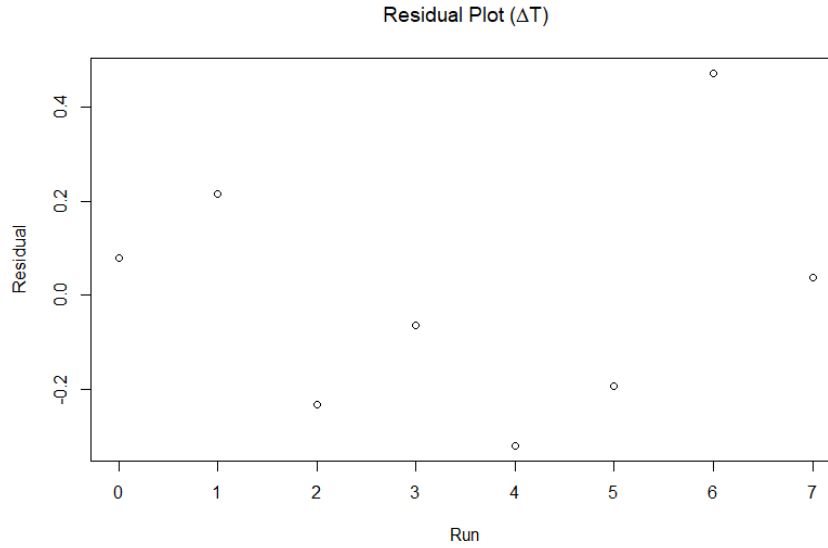


Figure 51: Residual plot for the linear two factor model. Residuals are higher for small particles and high k values (run 4 and run 6).

Equation 6.50 (or Equation 6.49, when also considering non significant factors) can now be used to calculate the effective thermal conductivity for a given sphere and bridge size, by setting the simulation error $\Delta T = 0$.

In Table 21 predictions for k , using the two different models Equation 6.50 and Equation 6.49, are compared for different sphere sizes ($-1 = 6$ mm, $0 = 8$ mm, $1 = 10$ mm) and bridge sizes. The coded results for k are not shown, but directly converted to uncoded values. Finally k is used to calculate the effective thermal conductivity k_{eff} .

It can be seen that the two-factor model's results do not deviate much from the interaction model. Based on the Daniel's plot, the usage of the simple linear model is recommended.

Table 21: Comparison of calculated k values for $\Delta T = 0$ K and resulting k_{eff} values for aluminium spheres. Coded values are used for sphere diameter D and bridge size d/D , while the result k was directly converted from coded to uncoded values.

D	d/D	2-factor model		interaction model	
		k	$k_{eff,2-fac}$	k	$k_{eff,inter}$
-1	-1	1.1815	0.121114	1.2382	0.126924
-1	1	1.1815	0.138898	1.1222	0.131926
1	-1	1.5817	0.182673	1.7005	0.196394
1	1	1.5817	0.206589	1.5428	0.201510
0	-1	1.3816	0.151719	1.4566	0.159954
0	1	1.3816	0.172581	1.3235	0.165327

6.5.4 Validation

To validate the findings of [Section 6.5](#) additional experiments and simulations have been carried out. Since the goal of this work was to find optimal simulation settings for a wide range of materials and because the Design of Experiments was carried out for aluminium spheres and glass, steel spheres are used for validation.

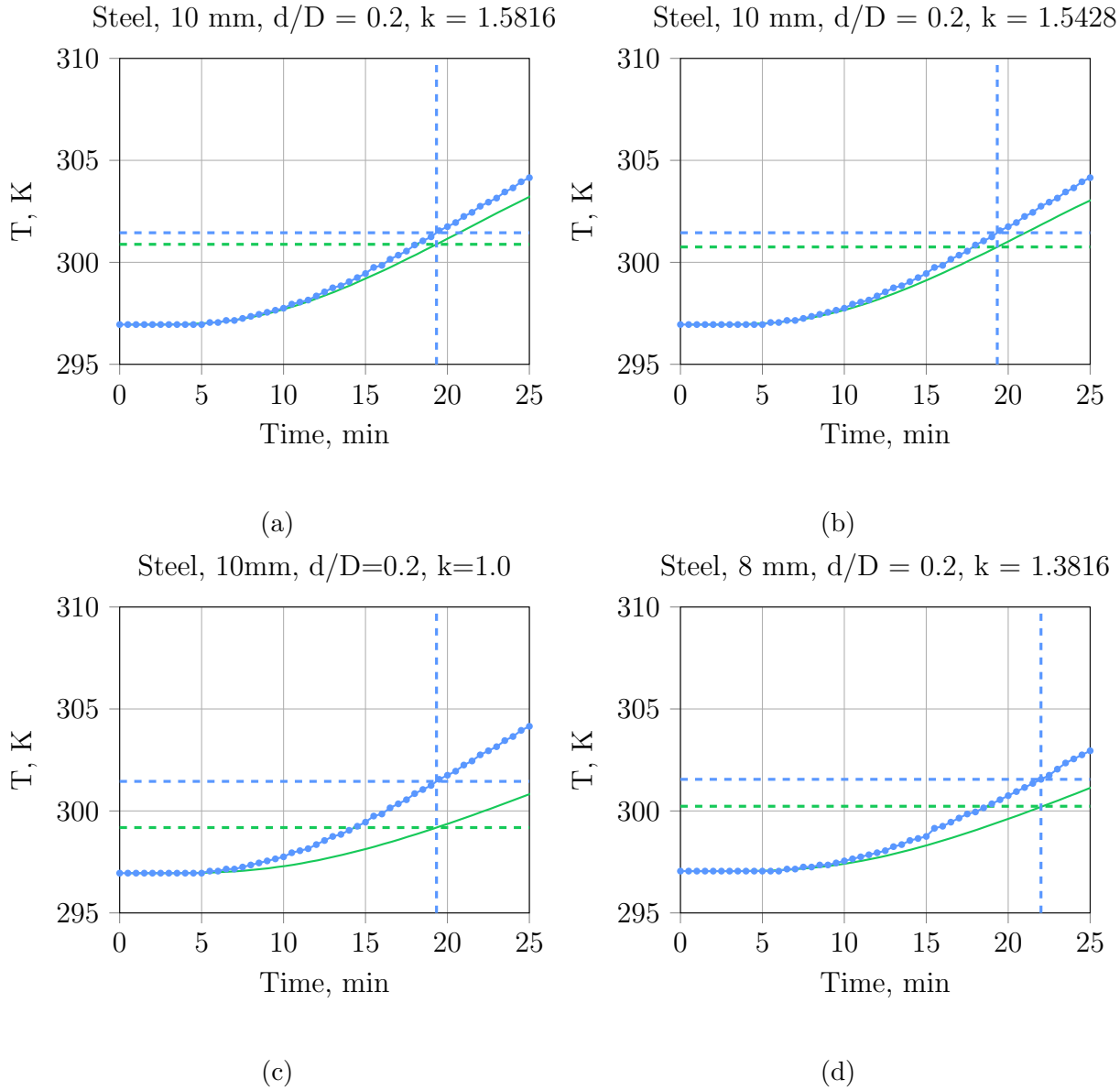


Figure 52: Comparison of CFD-simulation results and experimental surface temperature for steel spheres using an effective thermal conductivity according to (a) [Equation 6.50](#) and (b) [Equation 6.49](#). (c) unmodified effective thermal conductivity (d) [Equation 6.50](#). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5\text{ K}$ are marked with dashed lines in all plots.

In [Figure 52a](#) a comparison of experiments for $D = 10\text{mm}$ steel spheres and the simulation, using bridges of the size of $d/D = 0.2$ and the simple linear model for the effective thermal conductivity in the bridges ([Equation 6.50](#)) is shown. Here the simulation underestimates the surface temperature. The temperature difference after 19 minutes is -0.57 K . Using the simple linear regression model, the error ΔT was reduced by 1.79 K compared to a case with an unmodified effective thermal conductivity ([Figure 52c](#), $\Delta T = -2.26\text{ K}$). In [Figure 52b](#) the same case, but using the full interaction model to calculate the bridges effective thermal conductivity ([Equation 6.49](#)), is presented. The temperature difference after 19 minutes is -0.7 K . Here the simpler model gives better simulation results.

The simple linear model was then used in the simulation of $d = 8\text{ mm}$ steel spheres (bed consisting of 496 spheres), where the bridge size was set to $d/D = 0.2$. 8 mm spheres do represent the centre of the used full factorial design from [Section 6.5.3](#), which is not covered in screening. One can see that the simulated temperature is lower than in the experiments ($\Delta T = -1.32\text{ K}$). That indicates, that the simple linear model might not be sufficient to describe the effective thermal conductivity in this point and additional points should be included in the design (e.g. using a central composite face-centred (ccf) or central composite circumscribed (ccc) design for optimization). When using this design, it is possible to fit a quadratic model to the DoE results, so that estimations for $d = 8\text{ mm}$ spheres can be improved.

As here CFD simulation is compared to experiments, there are some factors that might have influence on the results:

- real thermal properties of bed material
- surface roughness
- bed voidage
- position of the temperature probe
- contact of temperature probe and particle
- inaccuracy of measurement
- particle diameter

6.5.4.1 Temperature Distribution

In all experiments not only the surface temperature in the middle of the packing was measured using a Pt1000 temperature probe, but also the temperature distribution was recorded using an infrared camera. [Figure 53](#) shows the measured and the simulated surface temperature field. The bed was created using an inhouse DEM-code and the simulated using the bridges method with a bridge size of $d/D = 0.2$ (case 6 in [Table 20](#)). The figure shows, that the temperature distribution of experiment and simulation are very similar. One can conclude that not only the temperature at a given point in the packing, but also the temperature distribution across the surface can be simulated when using the bridges method.

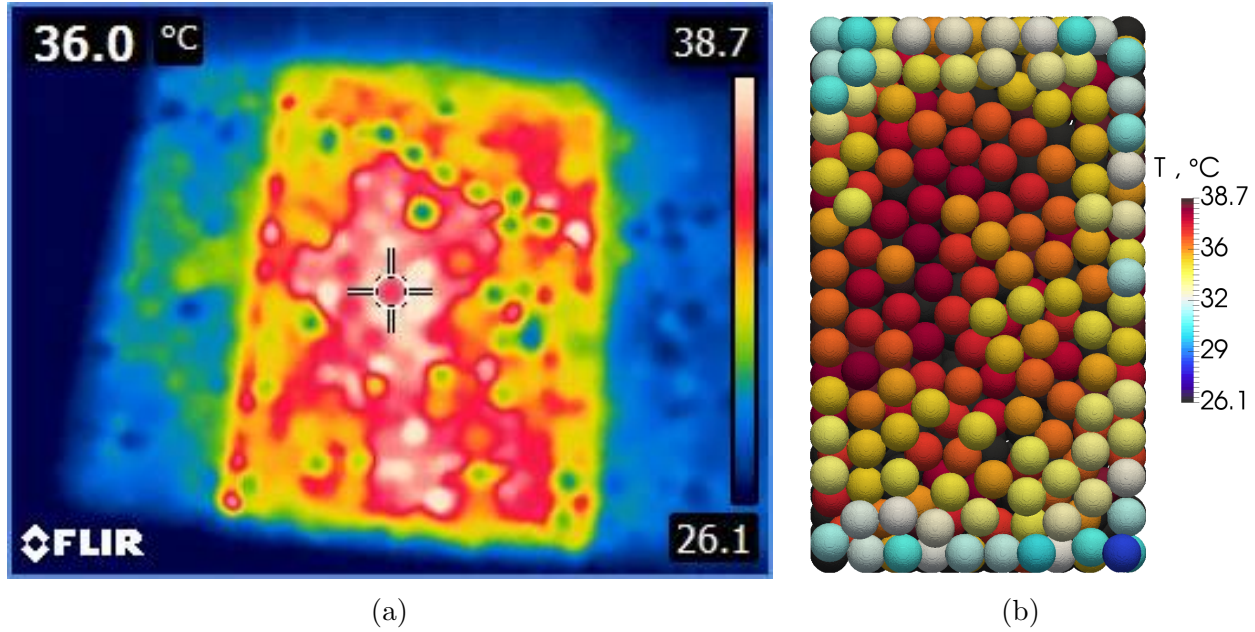


Figure 53: Comparison of measured ((a) IR-camera) and CFD-simulated ((b) bridge method) surface temperature of a packed bed consisting of 619 aluminium spheres ($D = 6 \text{ mm}$) after 10 minutes.

Figure 54(a)-(d) shows the temperature field over the bed height for aluminium spheres ($d = 6 \text{ mm}$), a bridge size of $d/D = 0.2$ and $k = 1.5$ (Run 6 of the Full Factorial Design) for four time steps. For comparison in Figure 54(e) the temperature field for 920 spheres using the necks method after 50 seconds is shown. It is clear that the necks method overpredicts the thermal conductivity of the bed.

The temperature distribution in two adjacent spheres, in a packing consisting of 619 aluminium and glass spheres total, is shown in Figure 55 and Figure 56 respectively. The main resistance for the heat transfer to a particle lies in the bridges for aluminium spheres. Thus there is nearly no temperature gradient inside the spheres. One can conclude that the effective thermal conductivity of the bridges has significant influence on the heat transfer of the bed for particles with high thermal conductivities. For glass spheres the temperature gradient in a single sphere is much higher, thus the effective thermal conductivity of the bridges has lower, but still significant, influence on the heat transfer in the bed.

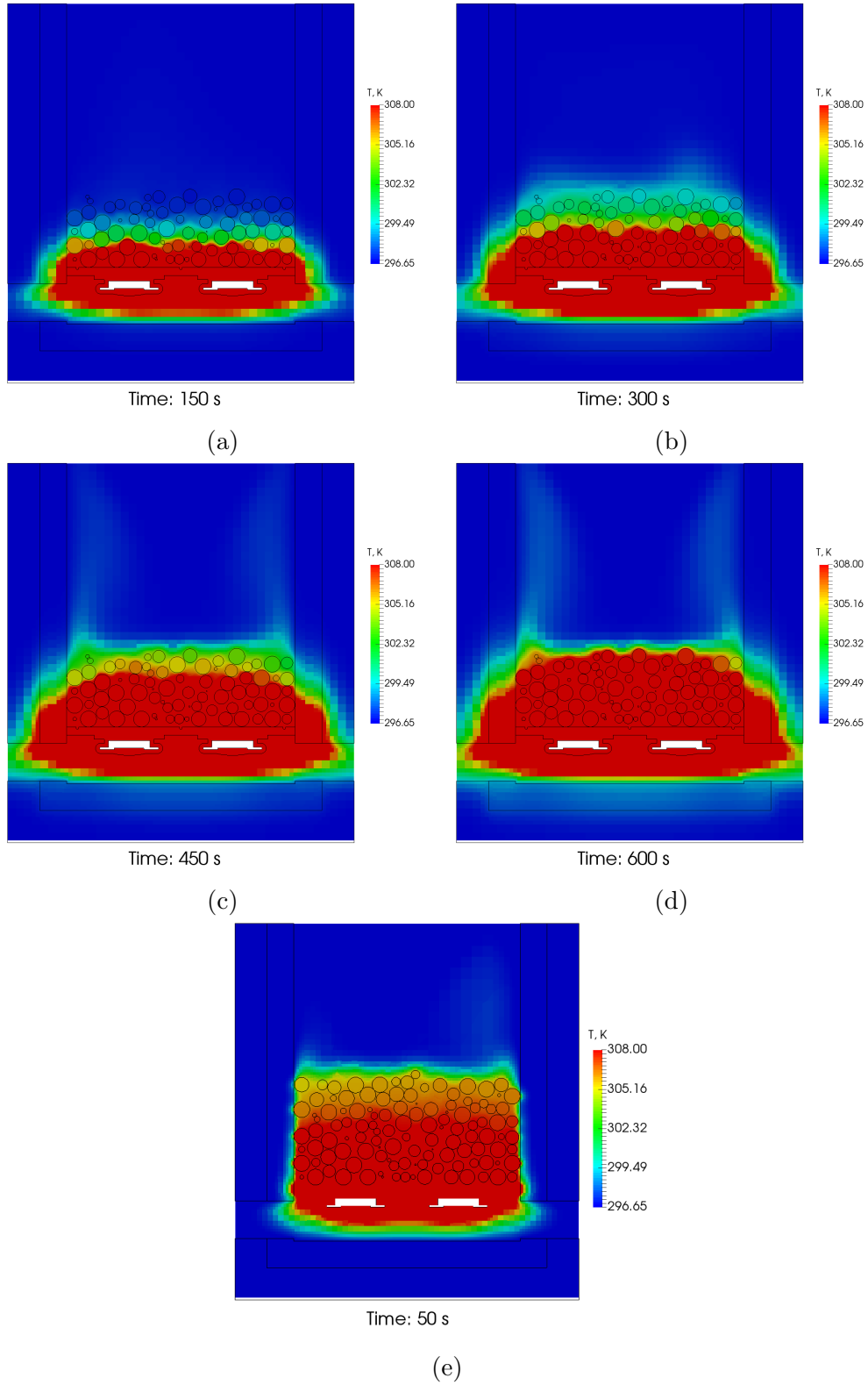
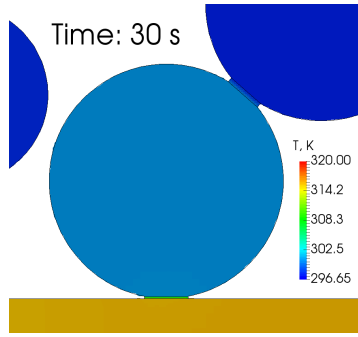
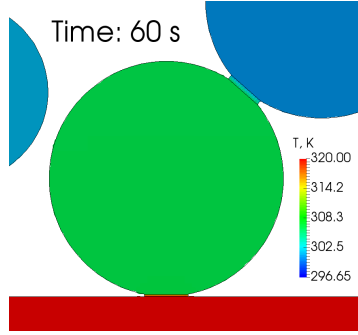


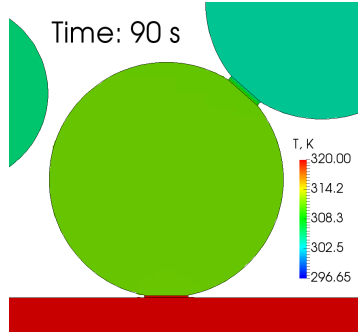
Figure 54: Temperature field over the bed height. (a)-(d): Run 6 of the Full Factorial Design; (e) 920 spheres using the necks method ($d/D = 0.2$)



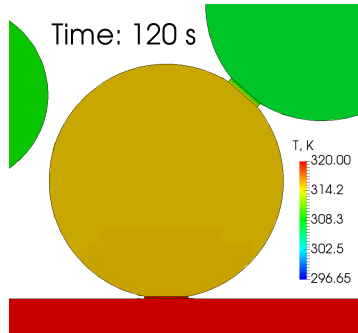
(a)



(b)

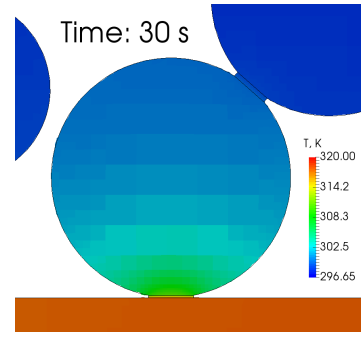


(c)

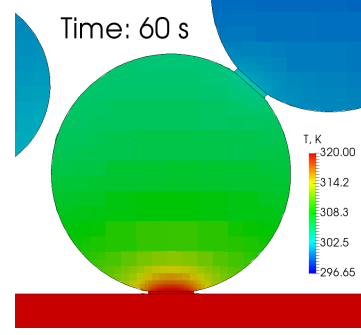


(d)

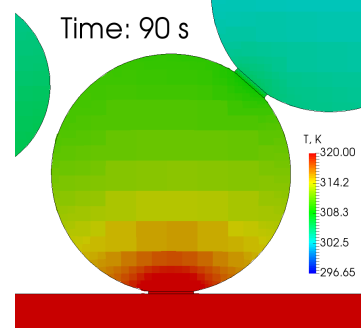
Figure 55: Temperature distribution in two adjacent spheres (aluminium, $d = 6\text{ mm}$, $d/D = 0.2$ and $k = 1.5$) in a packing of 619 spheres at different time steps.



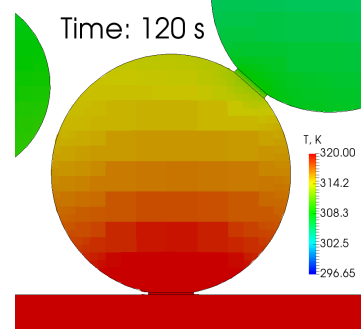
(a)



(b)



(c)



(d)

Figure 56: Temperature distribution in two adjacent spheres (glass, $d = 6\text{ mm}$, $d/D = 0.2$ and $k = 1.5$) in a packing of 619 spheres at different time steps.

7 Conclusion and Outlook

The presented work shows, that the bridges method gives good results for different materials, sphere diameter and bridge diameter, while the necks method is not recommended, when simulating the heat transfer in packed beds without forced convection. When using the necks method (bridge size $d/D = 0.2$) the difference of the packing's surface temperature from experiment to simulation was 31.95 K and 41.69 K for steel and aluminium spheres respectively. This is the temperature difference at time t , when the experimental measured surface temperature has increased by 4.5 K (Equation 7.51).

$$\Delta T_{exp} = T_{exp}(t) - T_{exp}(t = 0) = 4.5\text{ K} \quad (7.51)$$

Using the bridges method, the presented full factorial Design of Experiments reveals, that the bridge size has a negligible influence on the heat transfer in packed beds, for the tested range of particle diameter and bridge sizes.

Validation showed, that the simple linear model, found in Section 6.5.3, can be used to give estimates for the bridges effective thermal conductivity at the tested experimental points and for a wide range of thermal conductivities. The error was reduced from -2.26 K , when using no modification of the effective thermal conductivity in the bridges, to -0.57 K in validation for $d = 10\text{ mm}$ steel spheres and a bridge size of $d/D = 0.2$ using this model.

Surprisingly in all cases the thermal conductivity of the inserted bridge has to be significantly higher than previously reported (*Dixon, Nijemeisland, and Stitt (2013)*). This might be because no forced convection is considered here, so that conduction is the main heat transfer mechanism. For particles with higher conductivities, conduction in the packed bed is the main contributing part. For those materials the conductivity of the bridges has big influence on the heat transfer, as the interparticular heat transfer is the main resistance. So small changes in the thermal conductivity of the bridges do have significant influence on the overall bed conductivity (Section 6.5.4.1). This also shows, that the correlation for the effective thermal conductivity, purposed by *Dixon, Nijemeisland, and Stitt (2013)*, might not be suitable in all cases, as when using this correlation only conduction in the fluid and the solid is considered. To be able to even more generalise the model for the effective thermal conductivity of bridges between particles, the influence of secondary parameter (such as flattening of particles or surface roughness) should be studied in future.

Recommendations for simulating the heat transfer in packed beds, consisting of spheres with a wide range of thermal conductivities and natural convection, have been presented in this thesis. To be able to get even better results when simulating the heat transfer in packed beds without forced convection, an additional optimization design should be performed in future. The graphical representation of a possible optimization design is shown in Figure 57. Especially the interaction of the actual bridge size and the conductivity of the bridge needs further investigations. Thus a wider range of bridge sizes (e.g. $d/D = 0.10 - 0.25$) should be used in optimization.

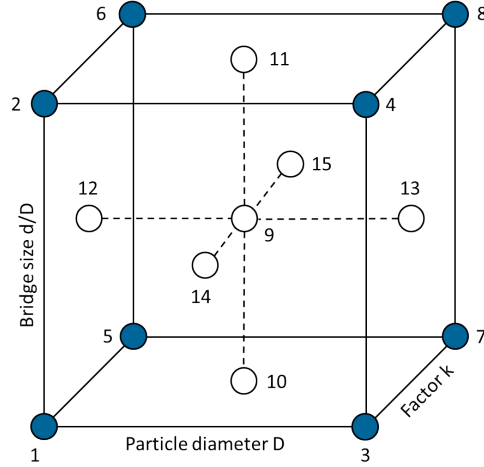


Figure 57: Possible central composite face-centred optimization design

One might also want to consider forced convection by applying different particle Reynolds Numbers Re_P , because the bridge diameter has an influence on the heat transfer and the pressure drop Δp for high Re_P , as other authors showed (*Wehinger (2016)*, *Dixon, Nijemeisland, and Stitt (2013)*, *Bu et al. (2014)*).

Also packings consisting of spheres with different particle size distributions (bimodal, normally distributed), packings of particles of different materials (e.g. aluminium + steel) and packings of non-spherical particles (e.g. cylinders) need more investigations in future. Especially for non-spherical particles, a new way for applying [Equation 2.27a-e](#), the effective thermal conductivity in the bridges, is needed. This is because the bridge size and form is different for each contact point.

List of Symbols

sub-/superscript	description
Al, aluminium	aluminium
ax	axial
bed	packed bed
b	bridge
c	core of the unit cell
corr	corrected value
eff	effective value
exp	experimental value
f	fluid
G	considering pressure dependence
glass	glass
heater	value at heater
i	single fraction in polydispersed bed
ir	infrared
measured	measured value
min	minimal
p	particle
Pt	Platinum resistance temperature probe
r	radial
rad	radiation
s	solid
sim	simulation value
st, steel	steel
th	theoretical value
∞	infinitely extended bed

symbol	description	unit
α	face non-orthogonality	$^{\circ}$
γ	accommodation coefficient	-
ϵ	emissivity	-
ζ	distribution coefficient	m^2/s
ϑ	Temperature	K
λ, Λ	thermal conductivity in Section 2.1	$\frac{W}{m K}$
Λ	mean free path of the gas molecules to calculate the influence of the Smoluchowski effect	m
ρ	density	$\frac{kg}{m^3}$
σ	radiation coefficient of the black body	$\frac{W}{m^2 K^4}$
σ_{ij}	stress tensor	N/m^2
φ	flattening coefficient	-
Ψ	bed porosity	-

symbol	description	unit
a	thermal diffusivity of the bed	m^2/s
a_0	volumetric heat source term	$\frac{W}{m^3}$
a_1	model constant (Bey and Eigenberger)	-
a_2	model constant (Giese)	-
b_1	model constant (Bey and Eigenberger)	-
b_2	model constant (Giese)	-
A	area	m^2
B	deformation parameter	-
c	specific heat capacity	$\frac{J}{kg\ K}$
C	Model constant $f(M_f)$	-
C_c	Calorimeter constant	$\frac{J}{K}$
C_f	shape factor	-
d	particle diameter (Section 2.1.4), bridge diameter (Section 2.2)	m
E	energy	J
H	bed diameter (Section 2.1.4), particle diameter (Section 2.2)	m
$f(\zeta)$	distribution function	-
H	bed height	m
f_p	dimensionless pressure drop	-
g	gravity	$\frac{m}{s^2}$
h	specific enthalpy	$\frac{J}{kg}$
J	diffusion flux	$\frac{kg}{s}$
k	reduced thermal conductivity in Section 2.1	-
k	thermal conductivity	$\frac{W}{m\ K}$
l	modified mean free path of gas molecules	m
m	mass	kg
M	molar mass	$\frac{kg}{kmol}$
N	column to particle diameter ratio in circular columns filled with spherical particles $N = D/d$	-
p	pressure	Pa
Pe	Peclet Number $Pe = u_0 \rho_f c_f d / \lambda_f$	-
Pr	Prandtl Number $Pr = c_p \mu / k$	-
\dot{q}	specific heat flux	W/m^2
\dot{Q}	heat flux	W
Q	volume fraction	-
r	sphere or bed radius	m
r^*	reduced distance from the wall	-
R	universal gas constant	$\frac{J}{kg\ K}$
Re	Reynolds number $Re = \rho u L / \mu$	-
S	source term for heat of chemical reactions	$\frac{J}{s}$
t	time	s
T	thermodynamic temperature	K
u_i	velocity component	$\frac{m}{s}$
V	Volume	m^3
x	height factor in numerical solution of axial heat equation	-
x_i	space coordinate	m
z	nondimensional distance from the wall	m

List of Figures

1	Unit cell of the Zehner/Bauer/Schlünder Model (<i>VDI-Wärmeatlas</i> (2013)) .	3
2	Bed porosities using Equation 2.5, Equation 2.8 and Equation 2.10 with $\Psi_\infty = 0.40$. z is the nondimensional distance from the wall, $z = (R - r)/d$. The unsteadiness of the model according to <i>deKlerk</i> (2003) at $z=0.637$ nearly disappears for $\Psi_\infty = 0.362$	7
3	Relative influence of axial heat transport ($n=1$), fluid-to-particle heat transfer ($n=2$), and intraparticle heat conduction ($n=3$) on the experimentally observable axial dispersion. (<i>VDI-Wärmeatlas</i> (2013))	9
4	Different global and local contact point modification methods	12
5	Geometry of the bridge and definition of the variables used in Equation 2.27 (<i>Dixon, Nijemeisland, and Stitt</i> (2013).)	15
6	Summary of screening and optimization	16
7	Example of common three factor designs used for screening or optimization. Each of the circles represents the levels of the factors for a single experiment. (a) Full Factorial Design, (b) Fractional Factorial Design, (c) Central Composite Face-Centered Design (CCF). (<i>Eriksson et al.</i> (2008))	17
8	Experimental setup. (a) Setup including platinum resistance sensor and infrared camera (b) 1...heating elements, 2...aluminium bottom plate, 3...plastic container, 4...isolation, 5...sphere packing	19
9	Experimental setup to identify the emission coefficient ϵ . (a) Heater plate, Pt1000 element, soldered to a unpainted sphere, and black painted sphere for IR measurement; (b) infrared picture of the setup	21
10	Infrared ($\epsilon = 0.98$) and Pt1000 temperature data for $d=6$ mm steel spheres .	21
11	Infrared ($\epsilon = 0.99$) and Pt1000 temperature data for $d=8$ mm steel sphere .	22
12	Infrared ($\epsilon = 0.98$) and Pt1000 temperature data for $d=10$ mm steel sphere. After 20 minutes ϵ was set to 0.99	22
13	Temperature dependence of the heater power for different voltage values. . .	23
14	Surface temperature of $d = 6$ mm glass spheres, $h_{bed} = 40$ mm. Solid bridges with $d/D = 0.14$ and glass properties have been used as contact point treatment in the simulation.	24
15	Prinziple of the calorimeter used to determine the heat capacity. 1...isolated Dewar vessel, 2...temperature probe, 3...stirrer, 4...electrical heater	25
16	Experimental setup to identify the heat capacity of different spheres.	26
17	(a) Drawing of the experimental geometry; (b) Experimental geometry 1...heating elements, 2...aluminium bottom plate, 3...plastic container, 4...isolation, 5...sphere packing	29
18	Heater temperature, and surface temperature of a packed bed containing 920 $D = 6$ mm aluminium spheres. Different colours refer to different experimental runs.	30
19	Procedure of creating a randomly packed bed of spheres. Bridges are inserted in SALOME before exporting the STL-files.	33
20	Test case for meshing. 1...fluid; 2...plastic container; 3...packing, consisting of five $D = 10$ mm spheres; 4...heater	36
21	Gap sizes at different walls	37

22	Gap refinement via surface refinement (a) <i>level (2 8)</i> and <i>level (4 8)</i> applied at surface geometry and packing respectively. None of the gaps are captured by the mesh. (b) uniform refinement <i>level (5 5)</i> at all surfaces.	38
23	Multi region surface refinement <i>level (2 3)</i> and <i>level (4 5)</i> for heater and packing respectively (a) yellow: location of problematic cells; colored: different boundaries (b) closer view of problematic cell.	39
24	Definition of face non-orthognonality α . c_1 and c_2 : cell centres; \vec{c}_i : vector connecting both cell centres; \vec{A}_i : face normal vector; α angle between c_i and A_i	39
25	Close look on the surface of a sphere meshed using the multi-region approach with mesh quality parameter shown in Appendix B.	40
26	Bridge size d/D for different snapping methods, background mesh sizes and surface refinement levels for the sphere packing.	41
27	Cross section of bridges between spheres using background mesh sizes from 1 mm (left) to 4 mm (right). (a)-(f): Spheres located in different sphere layers, explicit snapping; (h)-(n): Spheres located in same layer, explicit snapping; (o)-(u): Spheres located in different layers, implicit snapping	42
28	Location of bridges shown in Figure 27. (a)-(f): location of bridge between particles in different sphere layers, (h)-(n) location of bridge between particles in the same sphere layers	42
29	Test case to examine the time step size dependency of the heaters temperature when using a volumetric heat source term. Grey: Aluminium plane, colored: part of the heaters in which a time or temperature dependent, volumetric heat source term is applied.	44
30	Relative error of the produced energy for nine different time step sizes and a constant heat source term. The time step size decreases from top to bottom (Case Nr. 1-9)	45
31	Surface temperature of the aluminium plate for different time step sizes (Δt in seconds) after 50 simulated seconds. (a) Temperature range: 300 – 330 K, (a) Temperature range: 332.5 – 333.7 K	45
32	Relative error of the produced energy for four different time step sizes for a constant heat source term. The time step size increases from top to bottom (Case Nr. 1-4)	46
33	Voidage profile and sphere centres (top view) of a half packing of 920 spheres with diameter of 6 mm. The mesh voidage is compared to correlations discussed in Section 2.1.4 . Dashed lines do represent the average bed voidage.	49
34	Voidage profile and sphere centres (top view) of a half packing of 200 spheres with diameter of 10 mm. The mesh voidage is compared correlations discussed in Section 2.1.4 . Dashed lines do represent the average bed voidage.	50
35	Temperature distribution, stream lines and velocity vectors at outlet of a packing consisting of 920 $D = 6$ mm glass spheres after 30 simulated minutes.	51
36	Surface temperature of a packed bed consisting of 920 $D = 6$ mm aluminium spheres. Comparison of experiments, ZBS-model and CFD-simulation. The Temperature was measured at a height of 40 mm.	52
37	Surface temperature of a packed bed consisting of 920 $D = 6$ mm glass spheres. Comparison of experiments, ZBS-model and CFD-simulation. The Temperature was measured at a height of 40mm.	53

38	Surface temperature of a packed bed consisting of 200 $D = 10\text{ mm}$ steel spheres. Comparison of experiments, ZBS-model and CFD-simulation. The Temperature was measured at a height of 40 mm.	54
39	Bridging procedure in SALOME	56
40	STL-surface representation of 200 $D = 10\text{ mm}$ spheres with $d/D = 0.2$ bridges at particle-particle and particle-wall contact points.	56
41	Voidage profile and sphere centres (top view) of a half packing of 619 spheres with diameter of 6 mm, with bridge sizes of $d/D = 0.15$ and $d/D = 0.2$. The mesh voidage is compared to equations discussed in Section 2.1.4 . Dashed lines do represent the average bed voidage.	58
42	Voidage profile and sphere centres (top view) of a half packing of 100 spheres with diameter of 10 mm, with bridge sizes of $d/D = 0.15$ and $d/D = 0.2$. The mesh voidage is compared to discussed in Section 2.1.4 . Dashed lines do represent the average bed voidage.	59
43	Comparison of CFD-simulation results and experimental surface temperature for 6mm (a) glass and (b) aluminium spheres. The response for screening is the temperature difference of the two dashed horizontal lines (at $T_{exp}(t) - T_{exp}(t = 0) = 4.5\text{ K}$).	61
44	Raw data evaluation: (a) Response plot; (b) Histogram plot	62
45	Half normal plot of the interaction model	63
46	Geometrical representation of the 2^3 full factorial	64
47	Raw data evaluation: (a) Response plot; (b) Histogram plot	64
48	Model evaluation: (a) Main effects plot; (b) Interaction plot	66
49	Normal effect plot of the interaction model. D and k are placed well of the middle, and therefore considered as significant.	66
50	(a) Half normal plot, factors A (particle size D) and C (the multiplier k for modifying the effective thermal conductivity k_{eff}) are considered as significant; (b) Gap corrected half normal plot, in addition to A and C now the two factor interaction B:C (bridge size d/D and k) can be considered as significant. . .	67
51	Residual plot for the linear two factor model. Residuals are higher for small particles and high k values (run 4 and run 6).	69
52	Comparison of CFD-simulation results and experimental surface temperature for steel spheres using an effective thermal conductivity according to (a) Equation 6.50 and (b) Equation 6.49. (c) unmodified effective thermal conductivity (d) Equation 6.50 . The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5\text{ K}$ are marked with dashed lines in all plots. . . .	70
53	Comparison of measured ((a) IR-camera) and CFD-simulated ((b) bridge method) surface temperature of a packed bed consisting of 619 aluminium spheres ($D = 6\text{ mm}$) after 10 minutes.	72
54	Temperature field over the bed height. (a)-(d): Run 6 of the Full Factorial Design; (e) 920 spheres using the necks method ($d/D = 0.2$)	73
55	Temperature distribution in two adjacent spheres (aluminium, $d = 6\text{ mm}$, $d/D = 0.2$ and $k = 1.5$) in a packing of 619 spheres at different time steps.	74
56	Temperature distribution in two adjacent spheres (glass, $d = 6\text{ mm}$, $d/D = 0.2$ and $k = 1.5$) in a packing of 619 spheres at different time steps.	74
57	Possible central composite face-centred optimization design	76

A.1	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 619 aluminium spheres ($d = 6$ mm)	A-1
A.2	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 aluminium spheres ($d = 6$ mm) with one repetition.	A-2
A.3	Surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 aluminium spheres ($d = 6$ mm)	A-2
A.4	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 200 aluminium spheres ($d = 10$ mm)	A-3
A.5	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 619 glass spheres ($d = 6$ mm)	A-3
A.6	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 glass spheres ($d = 6$ mm) with two repetitions.	A-4
A.7	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 200 glass spheres ($d = 10$ mm)	A-4
A.8	Top line: temperature of first layer of spheres, next to bottom plate; middle line: temperature at height of 40mm; bottom line: surface temperature	A-5
A.9	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 steel spheres ($d = 6$ mm)	A-5
A.10	Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 200 steel spheres ($d = 10$ mm)	A-6
D.1	Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$).	D-2
D.2	Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	D-2
D.3	Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$).	D-3
D.4	Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	D-3

D.5	Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$).	D-4
D.6	Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines. . . .	D-4
D.7	Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$).	D-5
D.8	Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	D-5
D.9	Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$).	D-6
D.10	Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines. . .	D-6
D.11	Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$).	D-7
D.12	Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	D-7
D.13	Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$).	D-8
D.14	Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	D-8

D.15 Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$).	D-9
D.16 Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines. . . .	D-9
D.17 Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$).	D-10
D.18 Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines. . .	D-10
D.19 Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$).	D-11
D.20 Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines. . . .	D-11
D.21 Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$).	D-12
D.22 Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines. . .	D-12
D.23 Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$).	D-13
D.24 Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines. . . .	D-13

E.1	Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5428$).	E-2
E.2	Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5428$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	E-2
E.3	Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5816$).	E-3
E.4	Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5816$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	E-3
E.5	Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 496 steel spheres ($d = 8$ mm) using the bridges method ($d/D = 0.2$, $k = 1.3816$).	E-4
E.6	Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 496 steel spheres ($d = 8$ mm) using the bridges method ($d/D = 0.2$, $k = 1.3816$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	E-4
E.7	Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.0$).	E-5
E.8	Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.0$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.	E-5

List of Tables

1	Flattening coefficients for different particle shapes and materials (<i>VDI-Wärmeatlas</i> (2013))	5
2	Properties of particles and bed and Reynolds numbers used in <i>Erdim, Akgiray,</i> and <i>Demir</i> (2015)	11
3	Comparison of theoretical and measured temperature of ice water and boiling water.	20
4	Composition of steel spheres using X-ray fluorescence analysis	23
5	Thermal properties of soda lime glass according to the manufacturer. <i>Hilgenberg</i> (<i>Technical data sheet 0133</i>)	24
6	Temperatures and calorimeter constants at each time step for all four runs .	27
7	Temperatures and heat capacities of 6 mm and 10 mm glass spheres in the calorimeter experiment	28
8	Mean diameter and standard deviation of used spheres	28
9	Inner dimensions of the experimental setup	29
10	List of experiments carried out in this work.	30
11	Time steps used in the simulation and resulting error	44
12	Time steps used in the simulation of two adjacent cubes with an over all fine mesh and according error after 100 simulated seconds	46
13	Number of cells for the different packings and for each region ($\ast 10^6$)	47
14	Material properties used in the simulation	48
16	Number of cells for the different packings and for each region ($\ast 10^6$)	57
17	Number of particle-particle and particle-wall contact points for the two created packings	57
18	Comparison of the accommodation coefficient γ , the mean free path of the gas molecules Λ and the resulting effective thermal conductivity of cylindrical bridges between spherical particles, used from <i>Dixon, Nijemeisland,</i> and <i>Stitt</i> (2013) and from this work. k_{eff} is calculated for $D = 10\text{ mm}$ aluminium spheres and a bridge size of $d/D = 0.2$	60
19	2^{4-1} screening design for four factors and the response ΔT . $k_{eff,mod}$ is the with k scaled effective thermal conductivity k_{eff} of the bridges.	61
20	2^3 screening design for three factors and the response ΔT . $k_{eff,mod}$ is the with k scaled effective thermal conductivity k_{eff} of the bridges.	64
21	Comparison of calculated k values for $\Delta T = 0\text{ K}$ and resulting k_{eff} values for aluminium spheres. Coded values are used for sphere diameter D and bridge size d/D , while the result k was directly converted from coded to uncoded values.	69

References

- Antwerpen, W. van, C.G. du Toit, and P.G. Rousseau (2010). “A review of correlations to model the packing structure and effective thermal conductivity in packed beds of mono-sized spherical particles”. In: *Nuclear Engineering and Design* 240, pp. 1803–1818.
- Bauer, T. H. (1993). “A general analytical approach toward the thermal conductivity of porous media”. In: *International Journal of Heat and Mass Transfer* 36.17, pp. 4181–4191.
- Behnam, M. et al. (2013). “A new approach to fixed bed radial heat transfer modeling using velocity fields from computational fluid dynamics simulations”. In: *Industrial and Engineering Chemistry Research* 52, pp. 15244–15261.
- Bey, Oliver and Gerhart Eigenberger (1997). “Fluid flow through catalyst filled tubes”. In: *Chemical Engineering Science* 52, pp. 1365–1376.
- Box, George E. P. (1991). “George’s column: Finding bad values in factorial designs”. In: *Quality Engineering* 3.3, pp. 405–410.
- Brauer, Heinz (1971). *Grundlagen der Einphasen- und Mehrphasenströmungen*. Aarau [u.a.] : Sauerländer.
- Bu, S.S. et al. (2014). “On contact point modifications for forced convective heat transfer analysis in a structured packed bed of spheres”. In: *Nuclear Engineering and Design* 270, pp. 21–33.
- Carman, P. C. (1937). “Fluid flow through granular beds”. In: *Trans. Inst. Chem. Eng.* 15, pp. 32–48.
- Chen, Jhon C. and Stuart W. Churchill (1963). “Radiant Heat Transfer in Packed Beds”. In: *A.I.Ch.E. Journal* 9, pp. 35–41.
- deKlerk, Arno (2003). “Voidage Variation in Packed Beds at Small Column to Particle Diameter Ratio”. In: *AIChE Journal* 49.8, pp. 2022–2029.
- Dixon, A. G. and M. Nijemeisland (2001). “CFD as a design tool for fixed-bed reactors”. In: *Industrial and Engineering Chemistry Research* 40, pp. 5246–5254.
- Dixon, A. G., M. Nijemeisland, and E. H. Stitt (2013). “Systematic mesh development for 3D CFD simulation of fixed beds: Contact points study”. In: *Computers and Chemical Engineering* 48, pp. 135–153.
- Dixon, A. G. et al. (2011). “Systematic mesh development for 3D CFD simulation of fixed beds: Single sphere study”. In: *Computers and Chemical Engineering* 35, pp. 1171–1185.
- Dixon, A. G. et al. (2012). “Experimental validation of high Reynolds number CFD simulations of heat transfer in a pilot-scale fixed bed tube”. In: *Chemical Engineering Journal* 200–202, pp. 344–356.
- Eppinger, T., K. Seidler, and M. Kraume (2011). “DEM-CFD simulations of fixed bed reactors with small tube to particle diameter ratios”. In: *Chemical engineering Journal* 166, pp. 324–331.
- Erdim, Esra, Ömer Akgiray, and Ibrahim Demir (2015). “A revisit of pressure drop-flow rate correlations for packed beds of spheres”. In: *Powder Technology* 283, pp. 488–504.
- Ergun, Sabri (1952). “Fluid flow through packed columns”. In: *Chemical Engineering Progress* 48.2, pp. 89–94.
- Eriksson, L. et al. (2008). *Design of Experiments; Principles and Applications*. Vol. 3., rev. and enl. ed. Umetrics Academy.

- Ferziger, J. H. and M. Peric (2002). *Computational Methods for Fluid Dynamics*. 3rd. Springer, Berlin.
- Giese, Michael (1998). “Strömung in porösen Medien unter Berücksichtigung effektiver Viskositäten”. PhD thesis. München, Techn. Univ.
- Guardo, Alfredo et al. (2004). “CFD Flow and Heat Transfer in Nonregular Packings for Fixed Bed Equipment Design”. In: *Industrial and Engineering Chemistry Research* 43, pp. 7049–7056.
- Hilgenberg. *Technical data sheet 0133*. Online. URL:<http://www.hilgenberg-gmbh.de>, visited on 2018-01-21.
- JMP® (2004). *How do I interpret the Shapiro-Wilk test for normality in JMP?* online. URL: <http://www.jmp.com/support/notes/35/406.html>, visited on 2018-01-21.
- Kuroki, M. et al. (2007). “High-fidelity CFD modeling of particle-to-fluid heat transfer in packed bed reactors”. In: *Proceedings of European Congress of Chemical Engineering (ECCE-6)*.
- Mann, D., R.E. Field, and R. Viskanta (1992). “Determination of specific heat and true thermal conductivity of glass from dynamic temperature data”. In: *Wärme- und Stoffübertragung* 27.27, pp. 225–231.
- Nijemeisland, M. and A. G. Dixon (2001). “Comparison of CFD simulations to experiment for convective heat transfer in a gas–solid fixed bed”. In: *Chemical Engineering Journal* 82, pp. 231–246.
- Ookawara, S. et al. (2007). “High-fidelity DEM-CFD modeling of packed bed reactors for process intensification”. In: *Proceedings of European Congress of Chemical Engineering (ECCE-6)*.
- OpenCFD Ltd. 2017. *OpenFOAM® Documentation: Extended Code Guide*. Online. URL: <https://www.openfoam.com/documentation/>, visited on : 2018-01-24.
- Razali, Nornadiah Mohd and Yap Bee Wah (2011). “Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests”. In: *Journal of Statistical Modeling and Analytics* 2.1, pp. 21–33.
- Shafeeyan, Mohammad Saleh, Wan Mohd Ashri Wan Daud, and Ahmad Shamiri (2014). “A review of mathematical modeling of fixed-bed columns for carbon dioxide adsorption”. In: *Chemical Engineering Research and Design* 92, pp. 961–988.
- Tabib, Mandar V., Stein T. Johansen, and Shahriar Amini (2013). “A 3D CFD-DEM Methodology for Simulating Industrial Scale Packed Bed Chemical Looping Combustion Reactors”. In: *Industrial & Engineering Chemistry Research* 52, pp. 12041–12058.
- The OpenFOAM Foundation. *OpenFOAM v5 User Guide*. Online. URL: <https://cfd.direct/openfoam/user-guide/fvschemes/>, visited on 2018-01-23.
- Tsotsas, E. and H. Martin (1987). “Thermal conductivity of packed beds: A review”. In: *Chemical Engineering and Processing: Process Intensification* 22.1, pp. 19–37.
- VDI-Wärmeatlas (2013). *VDI-Wärmeatlas*. Vol. 11. VDI-Buch. Springer-Verlag Berlin Heidelberg.
- Wehinger, Gregor Dionys (2016). “Particle-resolved CFD simulations of catalytic flow reactors”. PhD thesis. Technischen Universität Berlin.
- Wikipedia. *Shapiro–Wilk test*. online. URL: <https://en.wikipedia.org/wiki/>, visited on 2018-01-21.
- Yagi, Sakae and Daizo Kunii (1960). “Studies on Heat Transfer Near Wall Surface in Packed Bades”. In: *A.I.Ch.E. Journal* 6, pp. 97–104.

- Yagi, Sakae and Noriaki Wakao (1959). “Heat and mass transfer from wall to fluid in packed beds”. In: *AIChE Journal* 5.1, pp. 79–85.
- Yang, Jian et al. (2012). “Experimental analysis of forced convective heat transfer in novel structured packed beds of particles”. In: *Chemical Engineering Science* 71, pp. 126–137.
- Zehner, P. and E. U. Schlünder (1970). “Wärmeleitfähigkeit von Schüttungen bei mäßigen Temperaturen”. In: *Chemie-Ing.-Techn.* 42.14, pp. 933–941.
- Zou, R.P. and A.B. Yu (1994). “The packing of spheres in a cylindrical container: the thickness effect”. In: *Chemical Engineering Science* 50, pp. 1504–1507.

Appendices

A Experimental Data

■ Heater Temperature Pt1000 ▲ Surface Temperature IR Cam, least squares regression ● Surface Temperature Pt1000

Different colours refer to different experiments. Dashed lines represent polynomial interpolated regression lines.

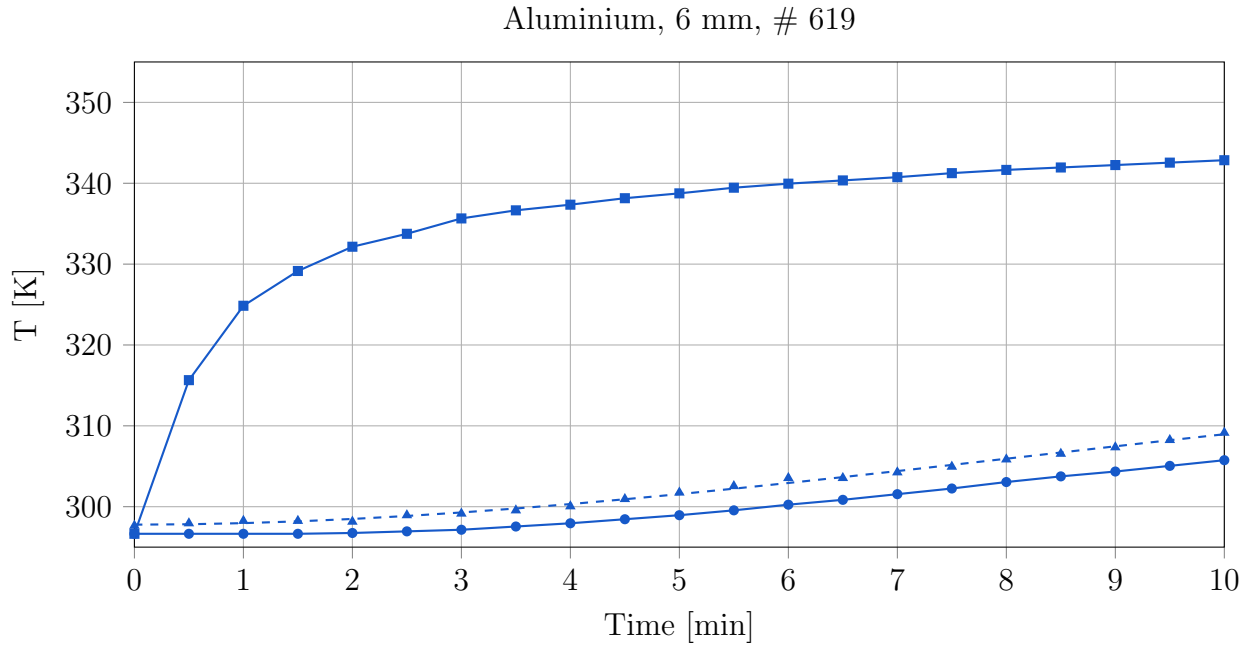


Figure A.1: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 619 aluminium spheres ($d = 6$ mm)

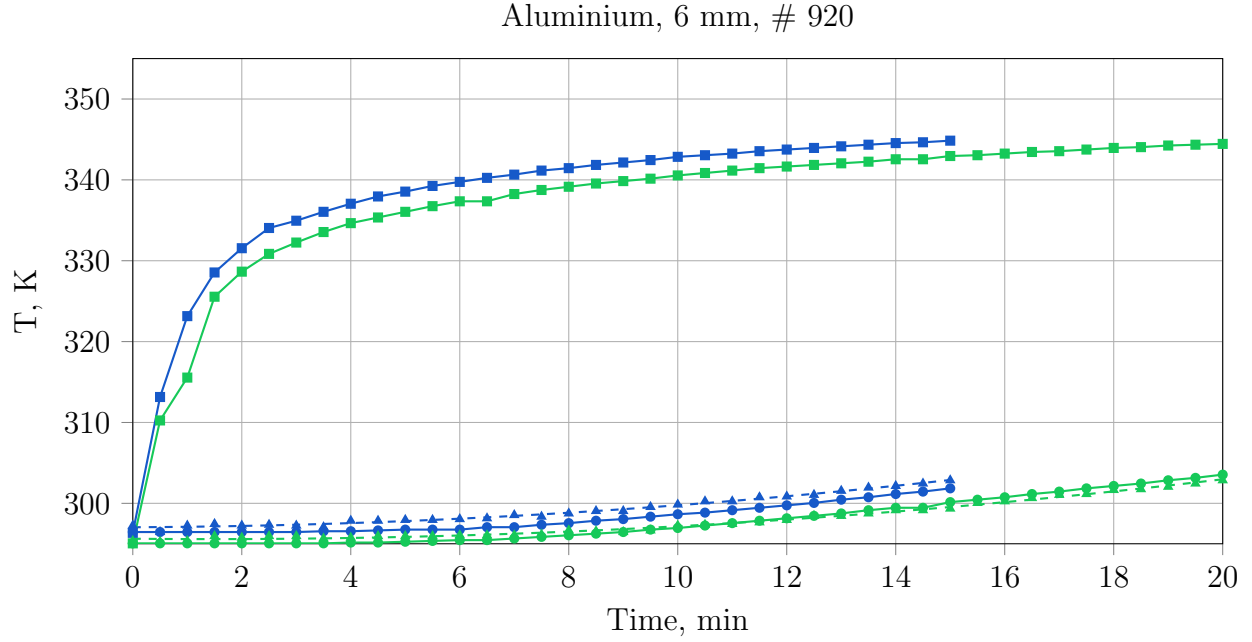


Figure A.2: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 aluminium spheres ($d = 6$ mm) with one repetition.

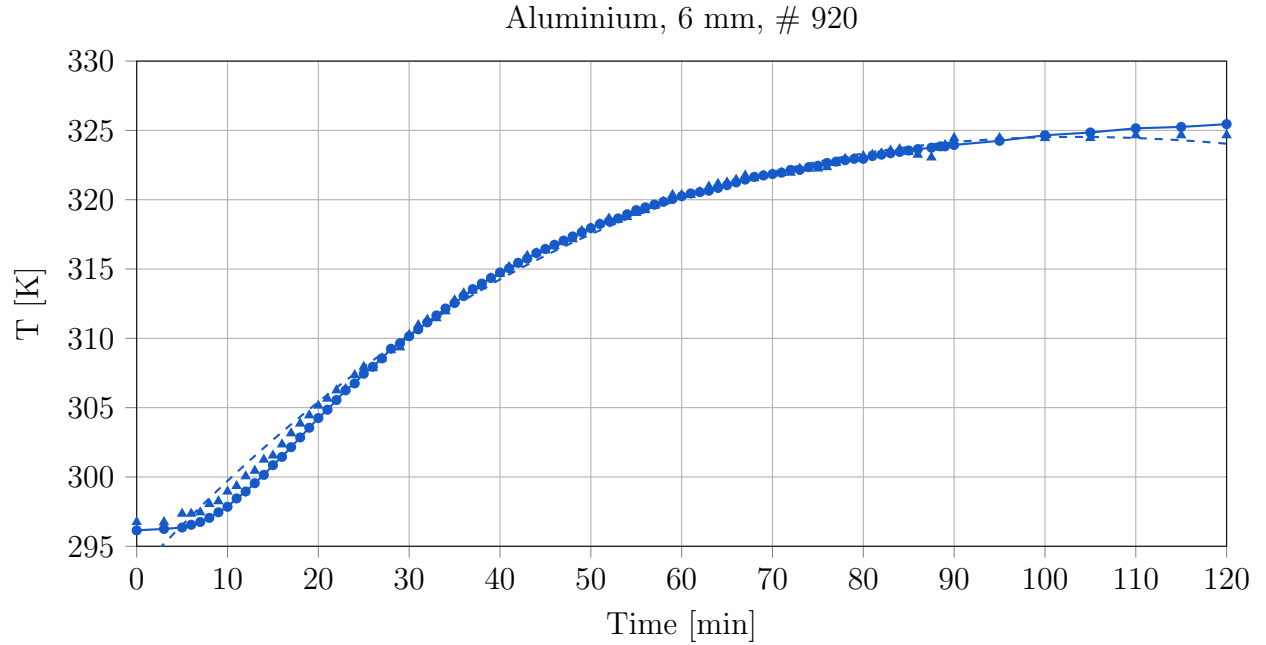


Figure A.3: Surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 aluminium spheres ($d = 6$ mm)

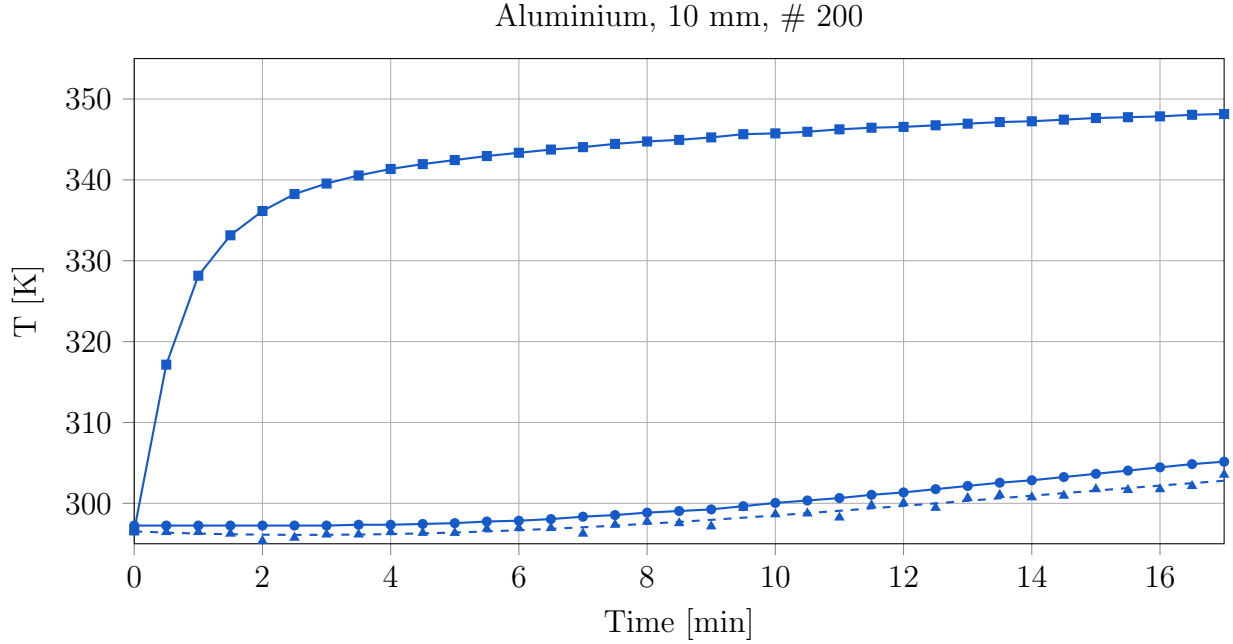


Figure A.4: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 200 aluminium spheres ($d = 10$ mm)

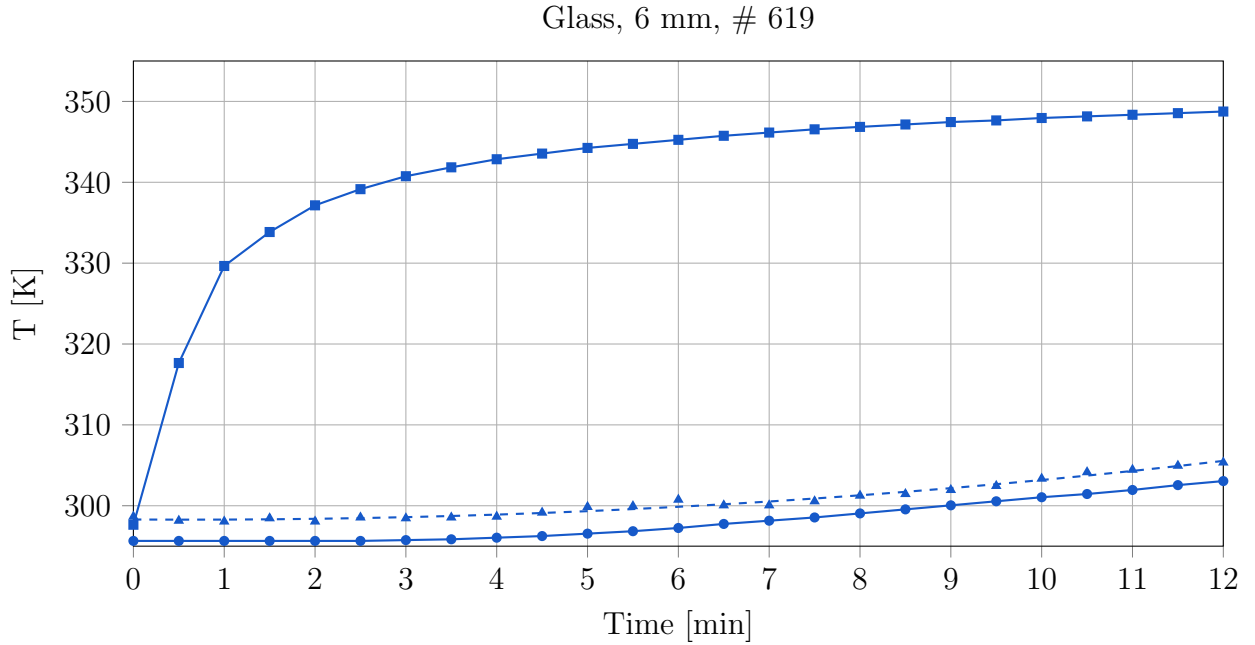


Figure A.5: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 619 glass spheres ($d = 6$ mm)

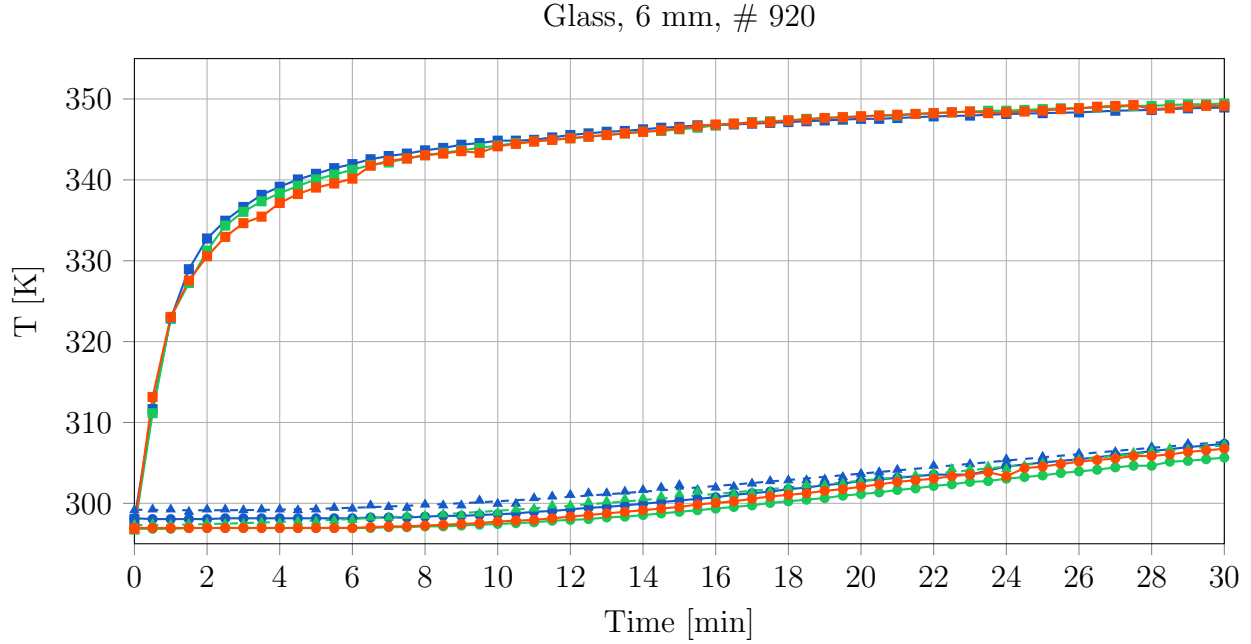


Figure A.6: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 glass spheres ($d = 6$ mm) with two repetitions.

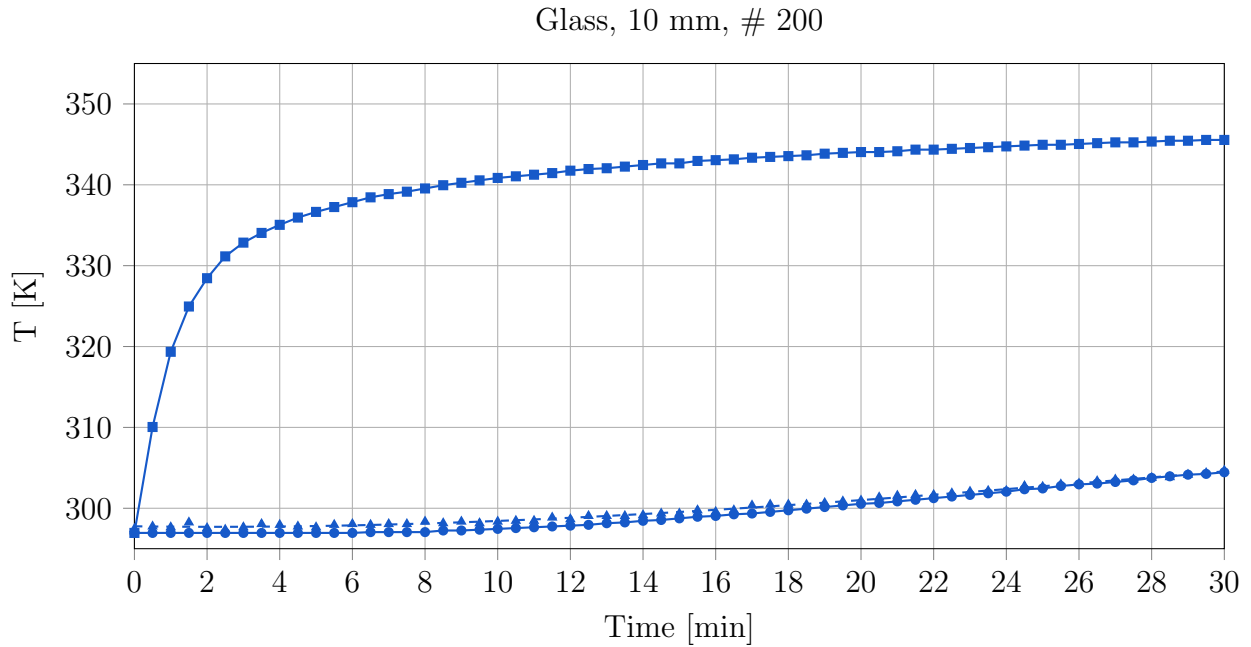


Figure A.7: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 200 glass spheres ($d = 10$ mm)

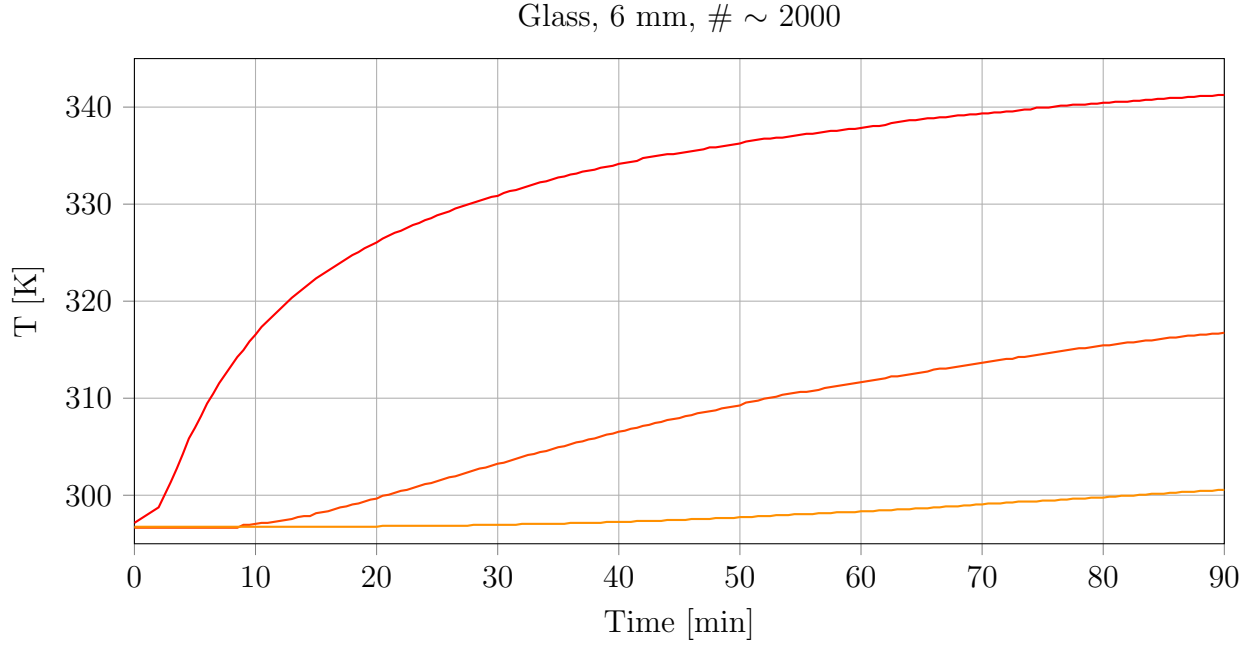


Figure A.8: Top line: temperature of first layer of spheres, next to bottom plate; middle line: temperature at height of 40mm; bottom line: surface temperature

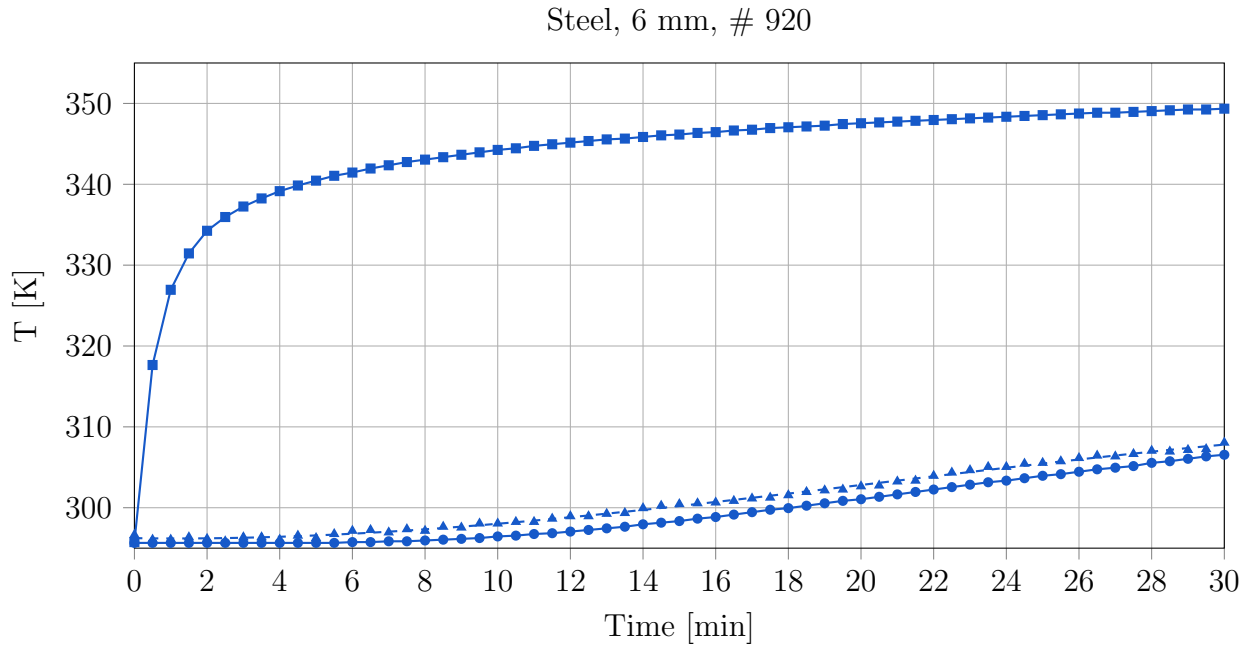


Figure A.9: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 920 steel spheres ($d = 6$ mm)

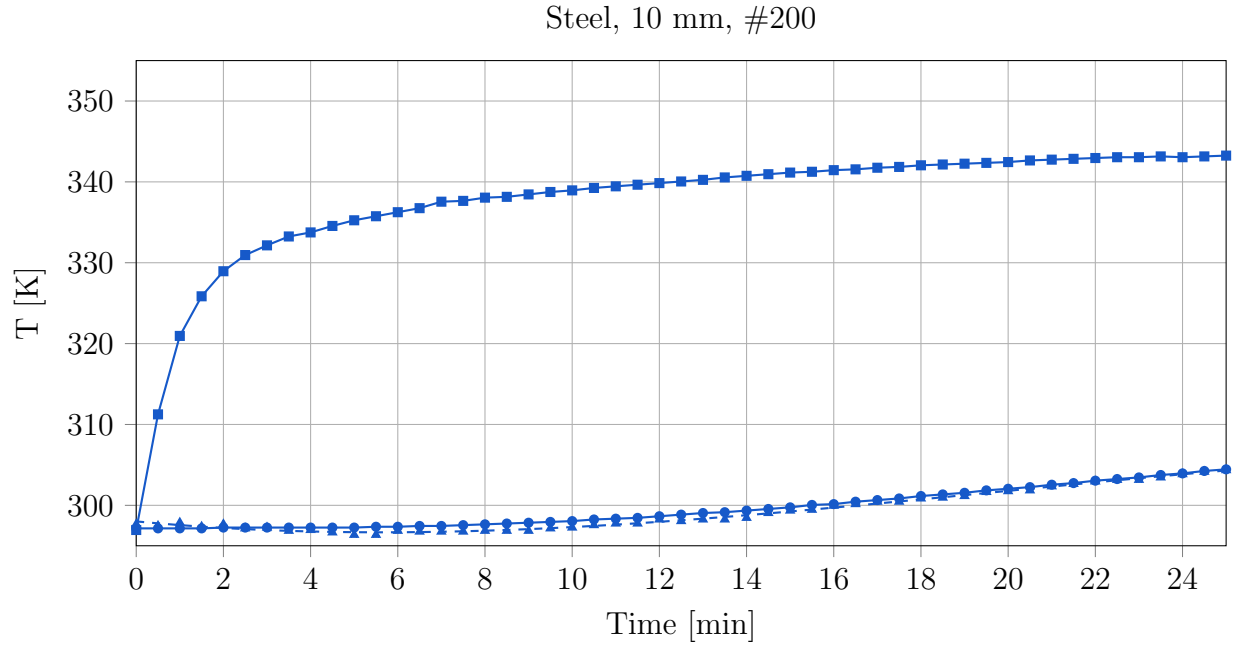


Figure A.10: Heaters temperature and surface temperature (measured using an IR-camera and a PT1000 temperature probe) of a packing consisting of 200 steel spheres ($d = 10$ mm)

B Mesh Quality Settings

```
/*----- C++ -----*/
|=====|
| \ \ \ \ | F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ \ \ \ | O p e r a t i o n | Version: 4.1
| \ \ \ \ | A n d | Web: www.OpenFOAM.org
| \ \ \ \ | M a n i p u l a t i o n |
|=====|
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       meshQualityDict;
}
// ***** //

maxNonOrtho 55;
maxBoundarySkewness 20;
maxInternalSkewness 4;
maxConcave 80;
minVol 1e-13;
minTetQuality 1e-15;
minArea -1;
minTwist 0.02;
minDeterminant 0.05;
minFaceWeight 0.05;
minVolRatio 0.05;
minTriangleTwist -1;

// ***** //
```

Listing B.1: meshQualityDict for refinement box at wall

C Important CFD-solver Settings

```
/*----- C++ -----*/
|=====|
| \ \ \ \ \ | F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ \ \ \ \ | O p e r a t i o n | Version: 4.1
| \ \ \ \ \ | A n d | Web: www.OpenFOAM.org
| \ \ \ \ \ | M a n i p u l a t i o n |
|=====|
/*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       fvSchemes;
}

// * * * * *

ddtSchemes
{
    default Euler;
}

gradSchemes
{
    default      Gauss skewCorrected linear;
    grad(U)      cellLimited Gauss linear 1;
}

divSchemes
{
    default      none;

    div(phi,U)    Gauss skewCorrected upwind;
    div(phi,K)    Gauss skewCorrected linear;
    div(phi,h)    Gauss skewCorrected upwind;
    div(phi,k)    Gauss skewCorrected upwind;
    div(phi,epsilon) Gauss skewCorrected upwind;
    div(phi,R)    Gauss skewCorrected upwind;
    div(R)        Gauss skewCorrected linear;
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss skewCorrected linear;
}

laplacianSchemes
{
    default      Gauss linear limited corrected 0.5;
}

interpolationSchemes
{
    default      skewCorrected linear;
}

snGradSchemes
{

```

```
}    default          limited corrected 0.5;  
// ***** //
```

Listing C.1: Schemes settings for solid region in OpenFoam


```

/*----- C++ -----*/
//
//      F i e l d
//      O p e r a t i o n
//      A n d
//      M a n i p u l a t i o n
//
OpenFOAM: The Open Source CFD Toolbox
Version: 4.1
Web: www.OpenFOAM.org
//
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0/leftSolid";
    object       T;
}
// * * * * *

dimensions      [ 0 0 0 1 0 0 0 ];

internalField    uniform 296.65;

boundaryField
{
    alu_to_bridges_WS_Heater
    {
        type                compressible::turbulentTemperatureCoupledBaffleMixed;
        value                uniform 296.65;
        Tnbr                 T;
        kappaMethod          solidThermo;
    }

    alu_to_heaterL
    {
        type                uniformFixedValue;
        uniformValue table
        (
            ( 0.0 296.65 )
            ( 30.0 315.65 )
            ( 60.0 324.85 )
            ( 90.0 329.15 )
            ( 120.0 332.15 )
            ( 150.0 333.75 )
            ( 180.0 335.65 )
            ( 210.0 336.65 )
            ( 240.0 337.35 )
            ( 270.0 338.15 )
            ( 330.0 339.45 )
            ( 360.0 339.95 )
            ( 390.0 340.35 )
            ( 420.0 340.75 )
            ( 450.0 341.25 )
            ( 480.0 341.65 )
            ( 510.0 341.95 )
            ( 540.0 342.25 )
        )
    }
}

```



```

( 570.0 342.55 )
( 600.0 342.85 )
( 6000.0 342.85 )
);
}

alu_to_plexi
{
    type                compressible::turbulentTemperatureCoupledBaffleMixed;
    value               uniform 296.65;
    Tnbr                T;
    kappaMethod         solidThermo;
}

alu_to_heaterR
{
    type                uniformFixedValue;
    uniformValue table
    (
( 0.0 296.65 )
( 30.0 315.65 )
( 60.0 324.85 )
( 90.0 329.15 )
( 120.0 332.15 )
( 150.0 333.75 )
( 180.0 335.65 )
( 210.0 336.65 )
( 240.0 337.35 )
( 270.0 338.15 )
( 330.0 339.45 )
( 360.0 339.95 )
( 390.0 340.35 )
( 420.0 340.75 )
( 450.0 341.25 )
( 480.0 341.65 )
( 510.0 341.95 )
( 540.0 342.25 )
( 570.0 342.55 )
( 600.0 342.85 )
( 6000.0 342.85 )
);
}

alu_to_air
{
    type                compressible::turbulentTemperatureCoupledBaffleMixed;
    value               uniform 296.65;
    Tnbr                T;
    kappaMethod         solidThermo;
}
}
// ***** //

```

Listing C.3: Boundary conditions for bottom plate (heater boundary)

D Simulation Data, DoE Results



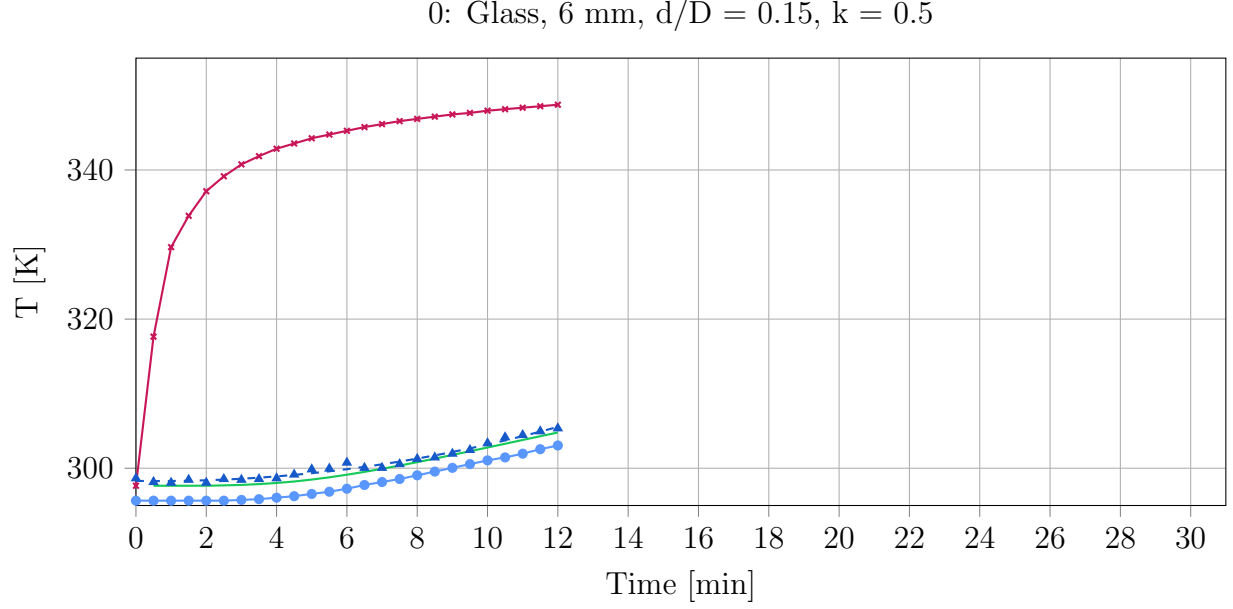


Figure D.1: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$).

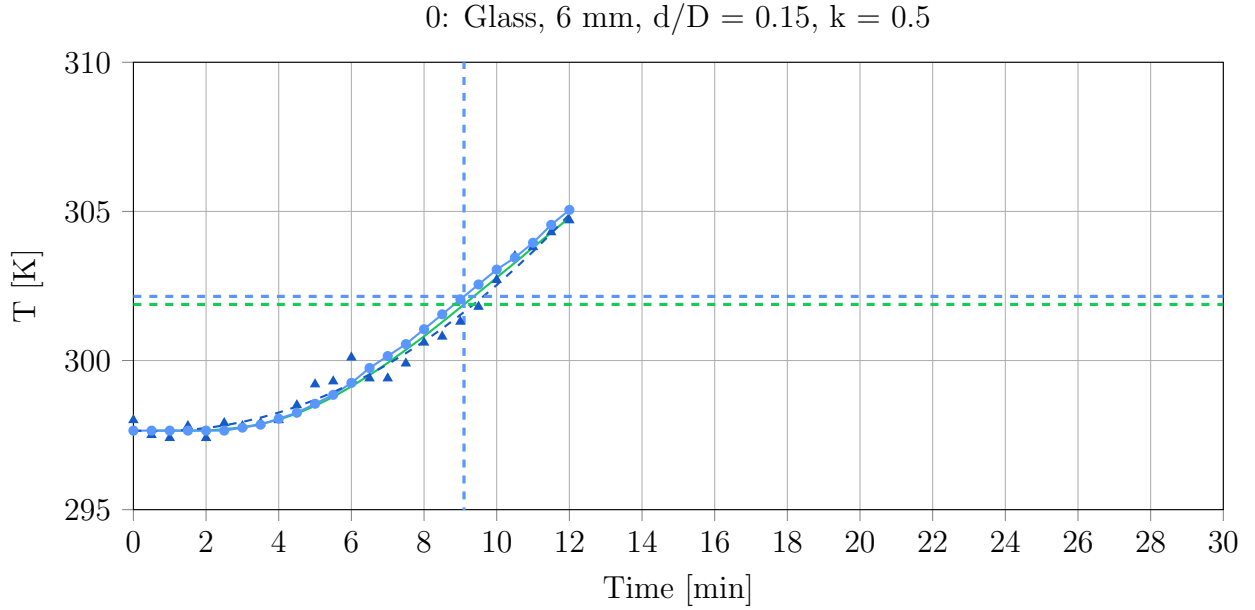


Figure D.2: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

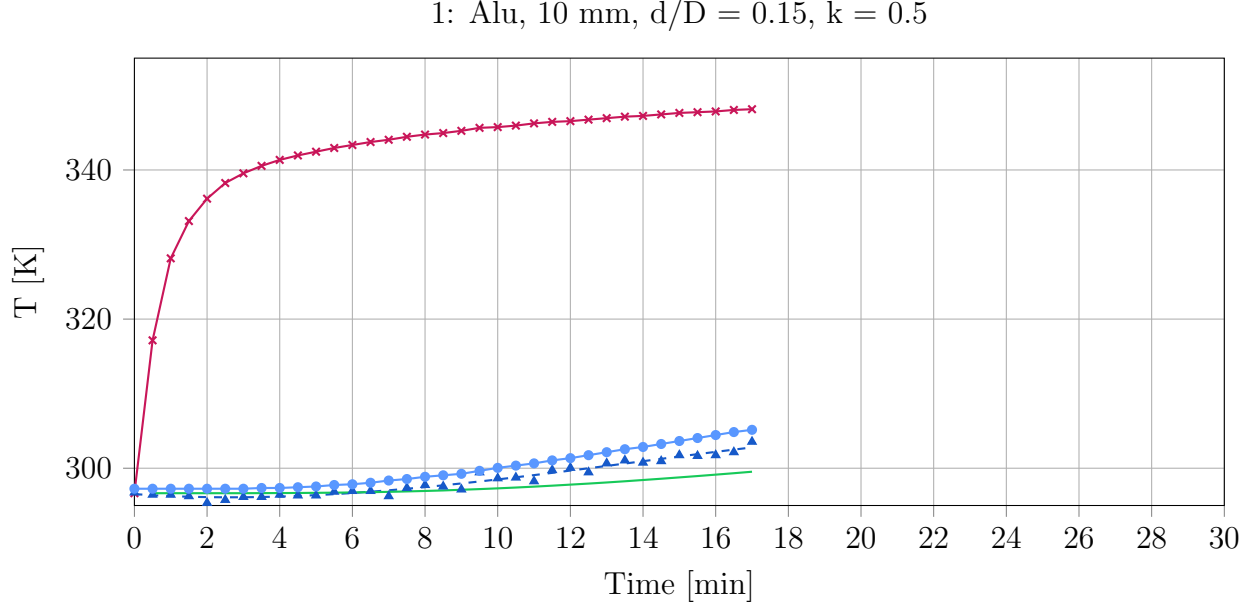


Figure D.3: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$).

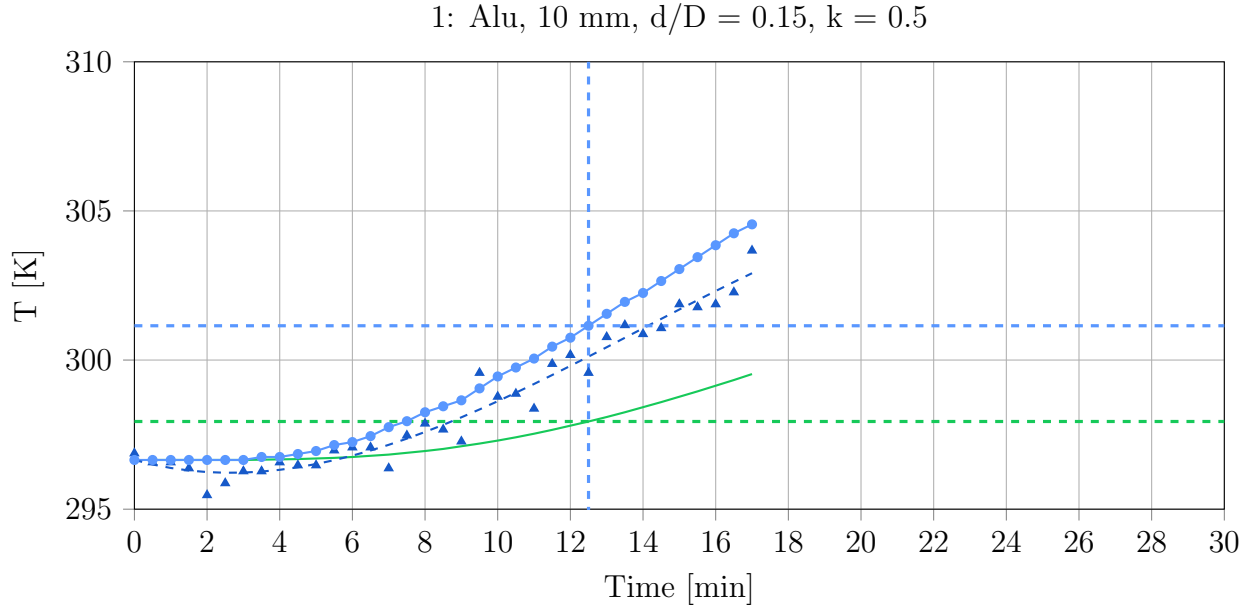


Figure D.4: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

2: Alu, 6 mm, $d/D = 0.2$, $k = 0.5$

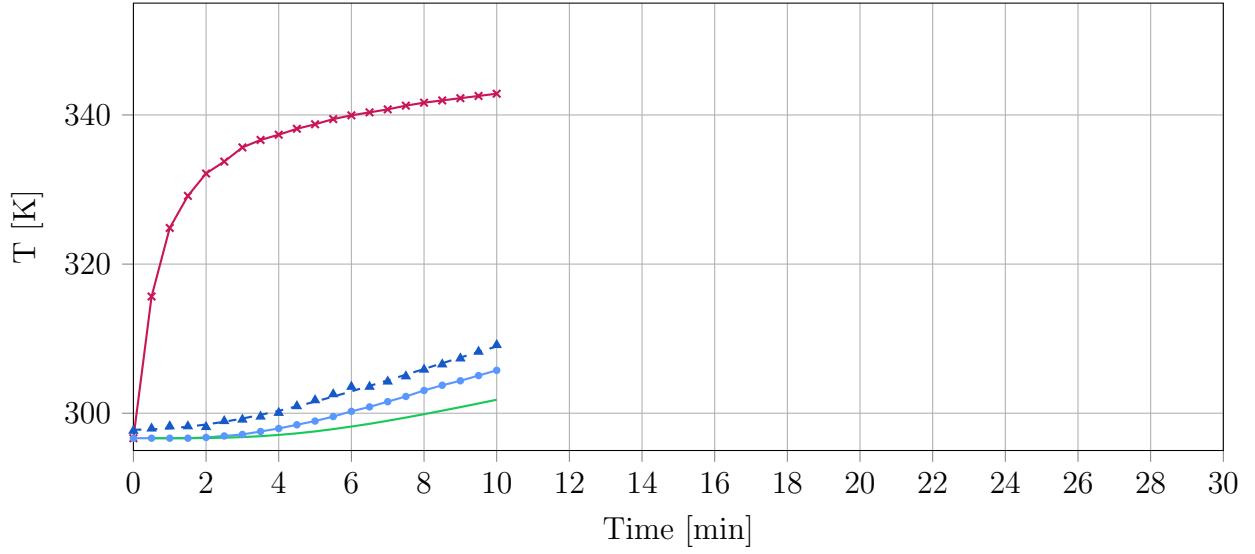


Figure D.5: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$).

2: Alu, 6 mm, $d/D = 0.2$, $k = 0.5$

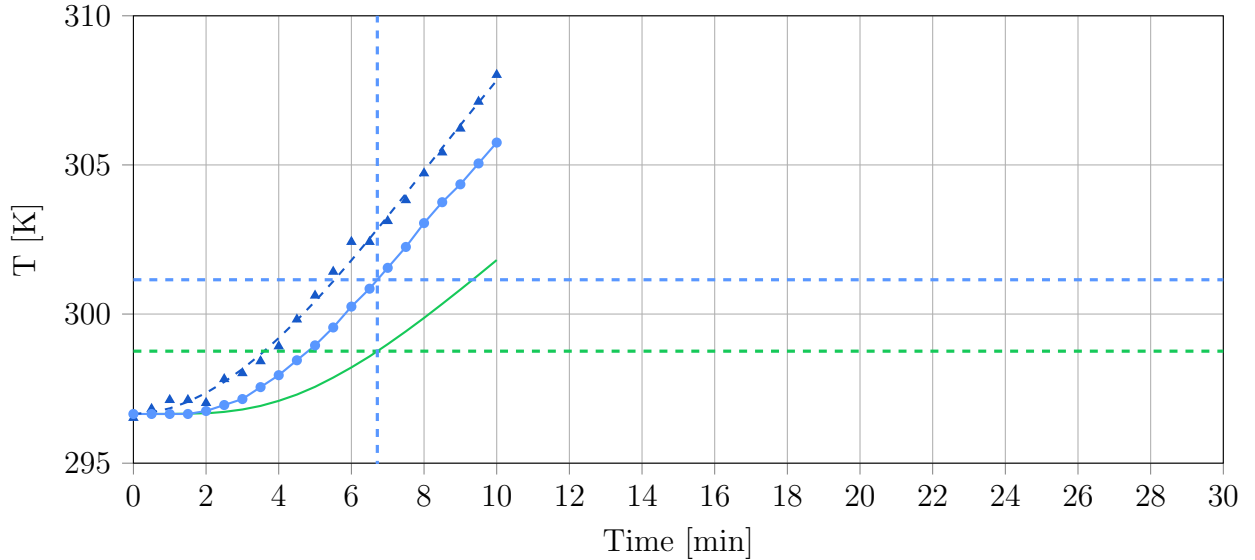


Figure D.6: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

3: Glass, 10 mm, $d/D = 0.2$, $k = 0.5$

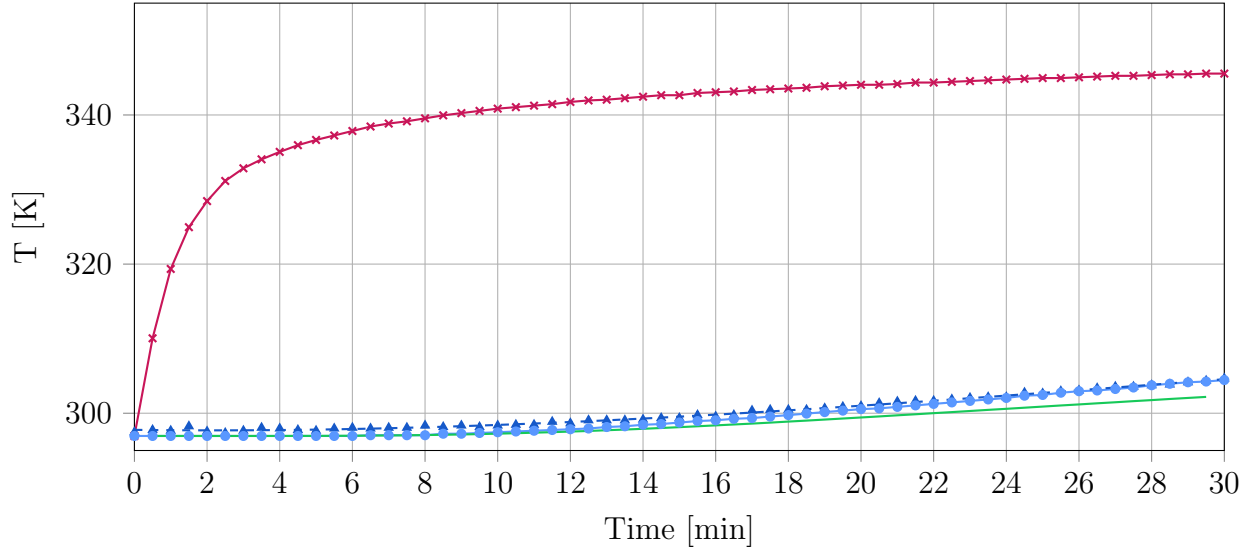


Figure D.7: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$).

3: Glass, 10 mm, $d/D = 0.2$, $k = 0.5$

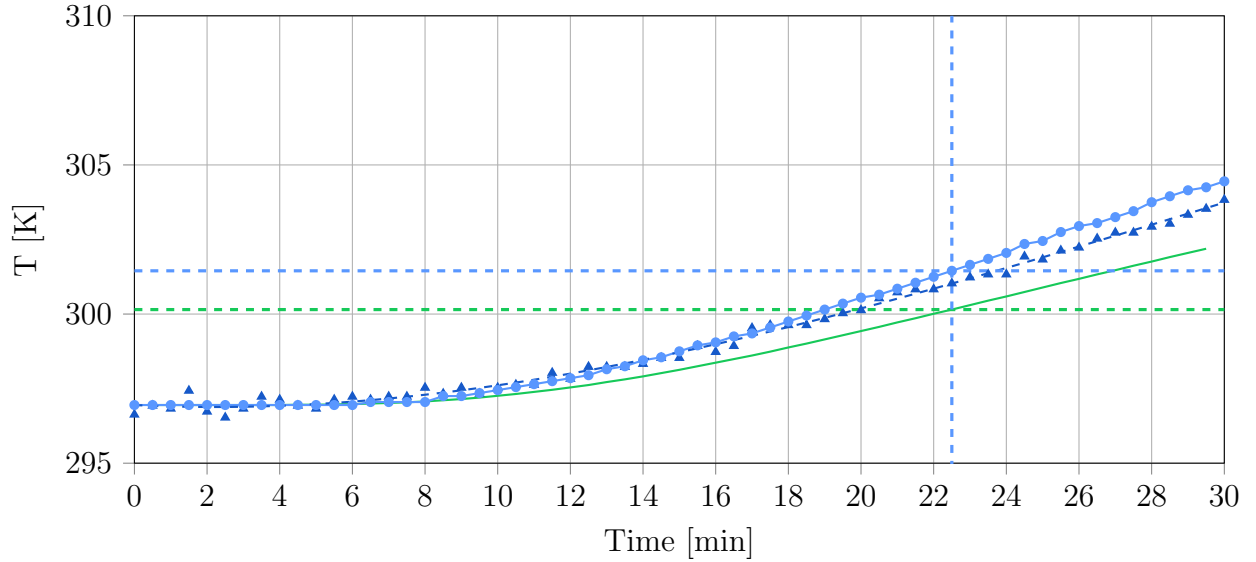


Figure D.8: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

4: Alu, 6 mm, $d/D = 0.15$, $k = 1.5$

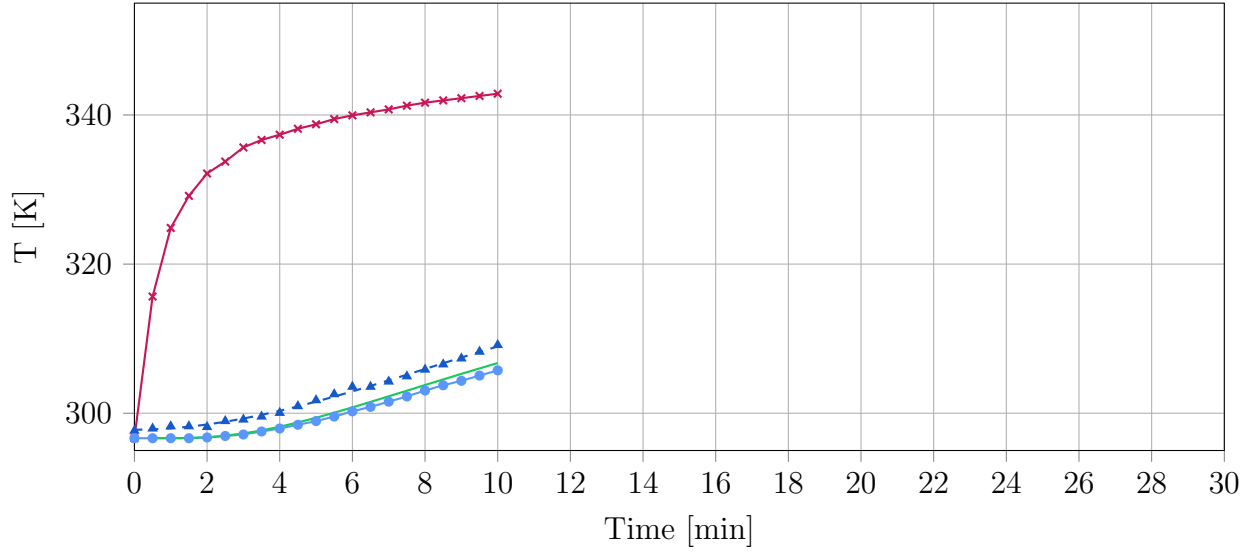


Figure D.9: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$).

4: Alu, 6 mm, $d/D = 0.15$, $k = 1.5$

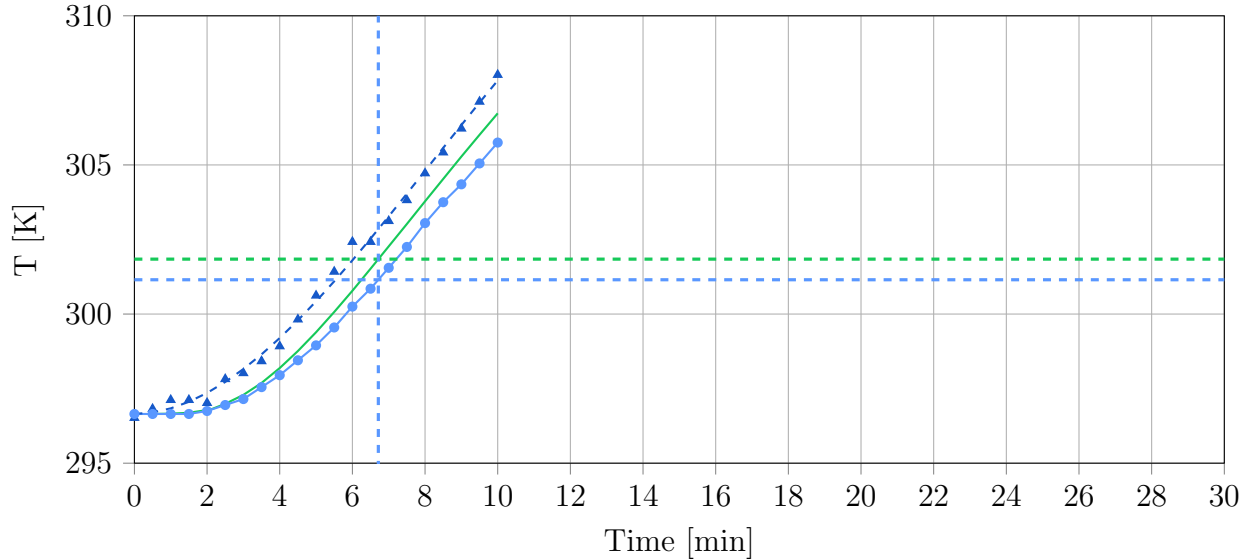


Figure D.10: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

5: Glass, 10 mm, $d/D = 0.15$, $k = 1.5$

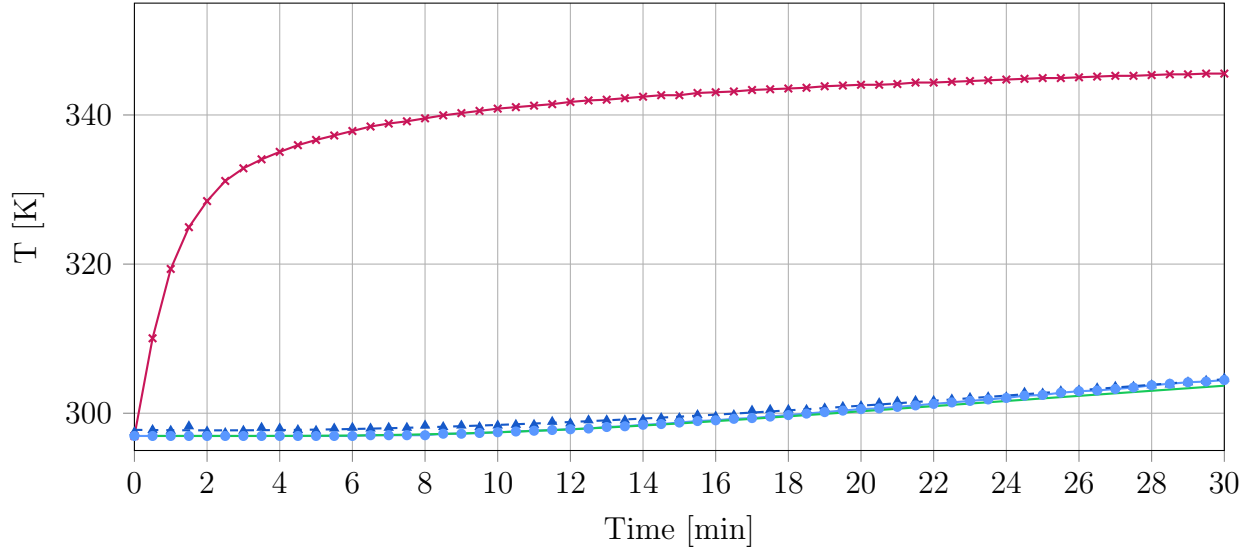


Figure D.11: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$).

5: Glass, 10 mm, $d/D = 0.15$, $k = 1.5$

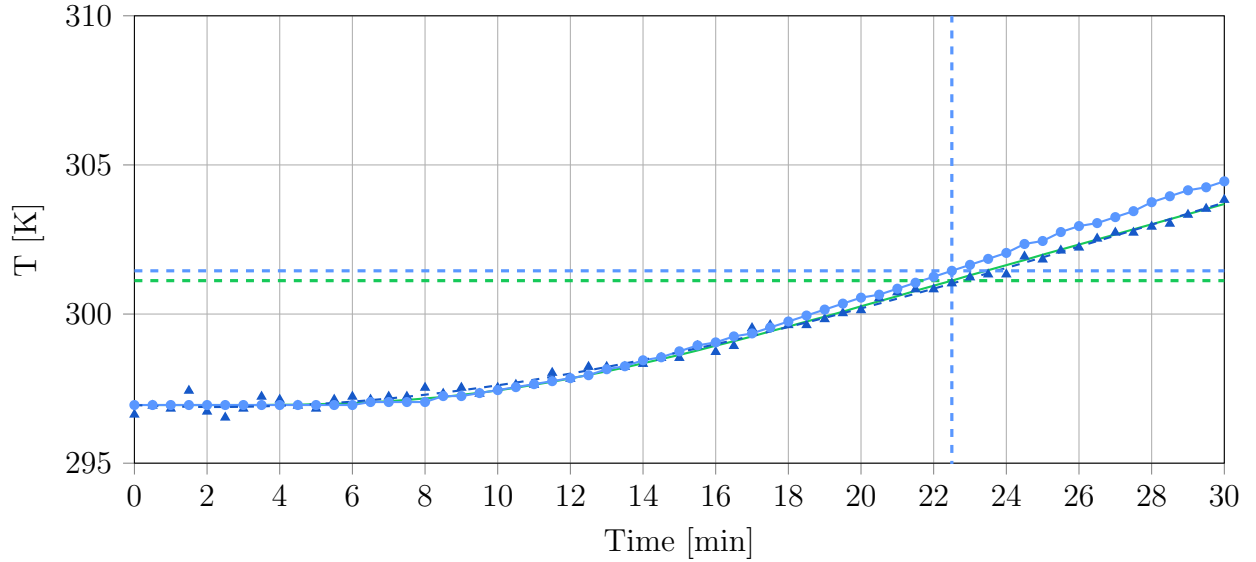


Figure D.12: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 glass spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

6: Glass, 6 mm , $d/D = 0.2$, $k = 1.5$

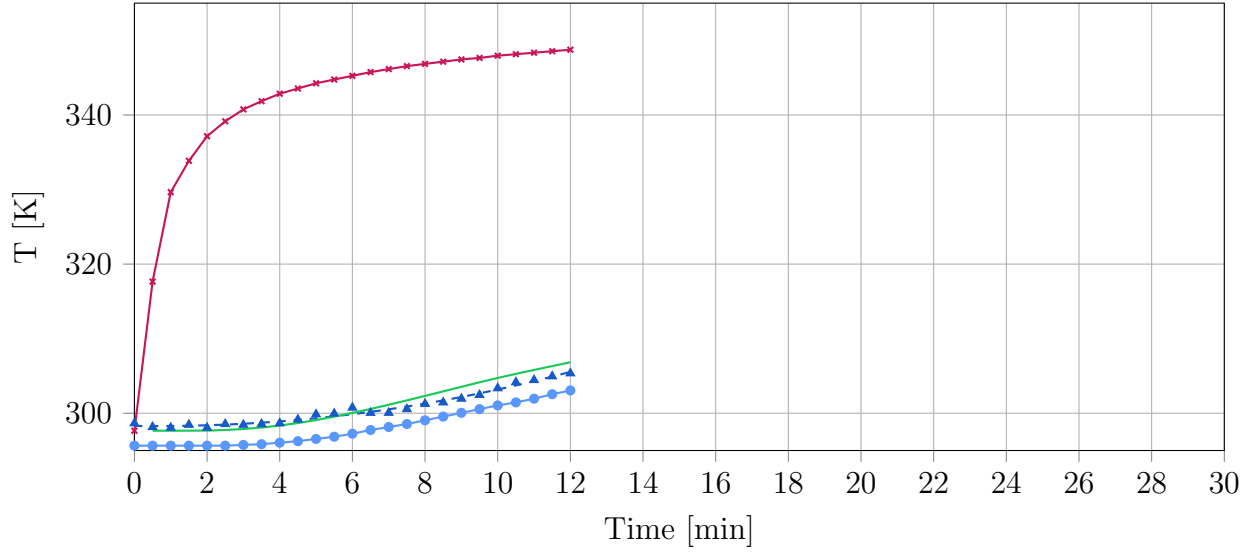


Figure D.13: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$).

6: Glass, 6 mm, $d/D = 0.2$, $k = 1.5$

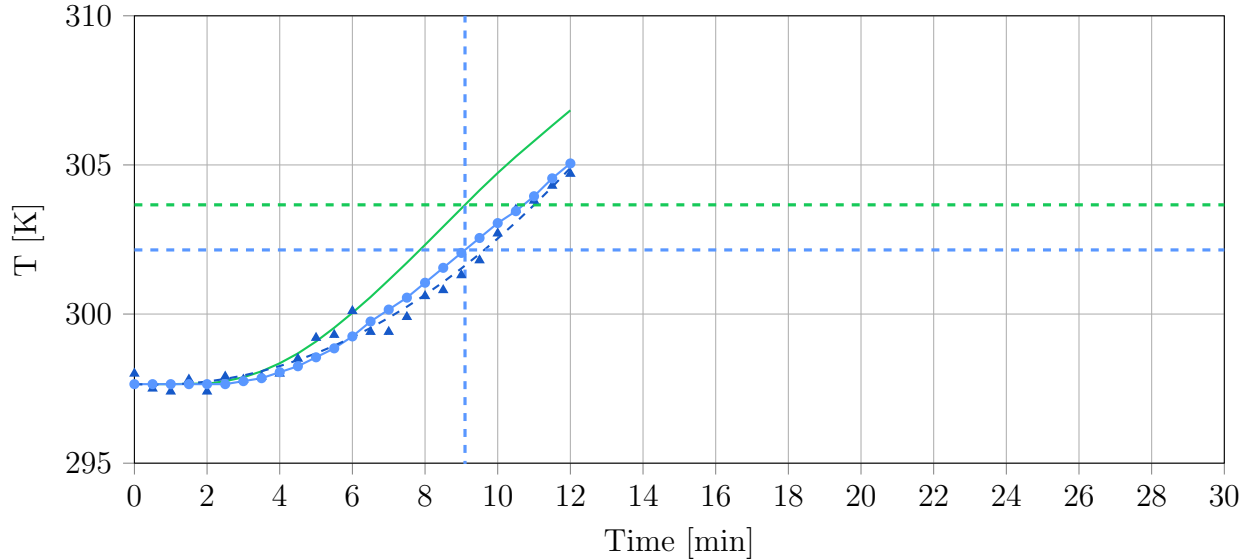


Figure D.14: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 glass spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

7: Alu, 10 mm, $d/D = 0.2$, $k = 1.5$

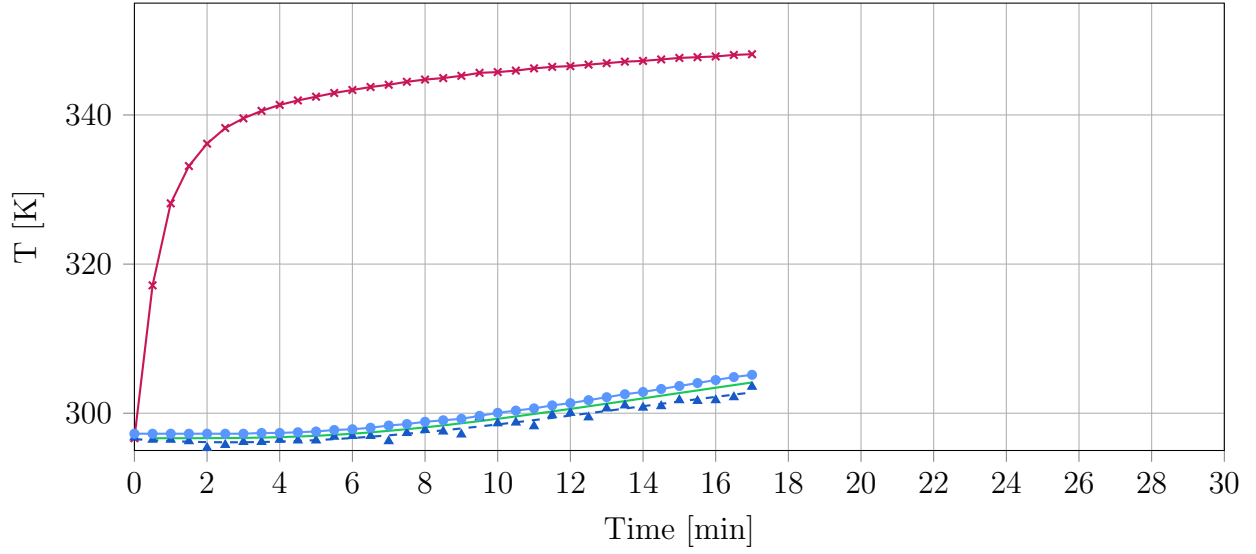


Figure D.15: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$).

7: Alu, 10 mm, $d/D = 0.2$, $k = 1.5$

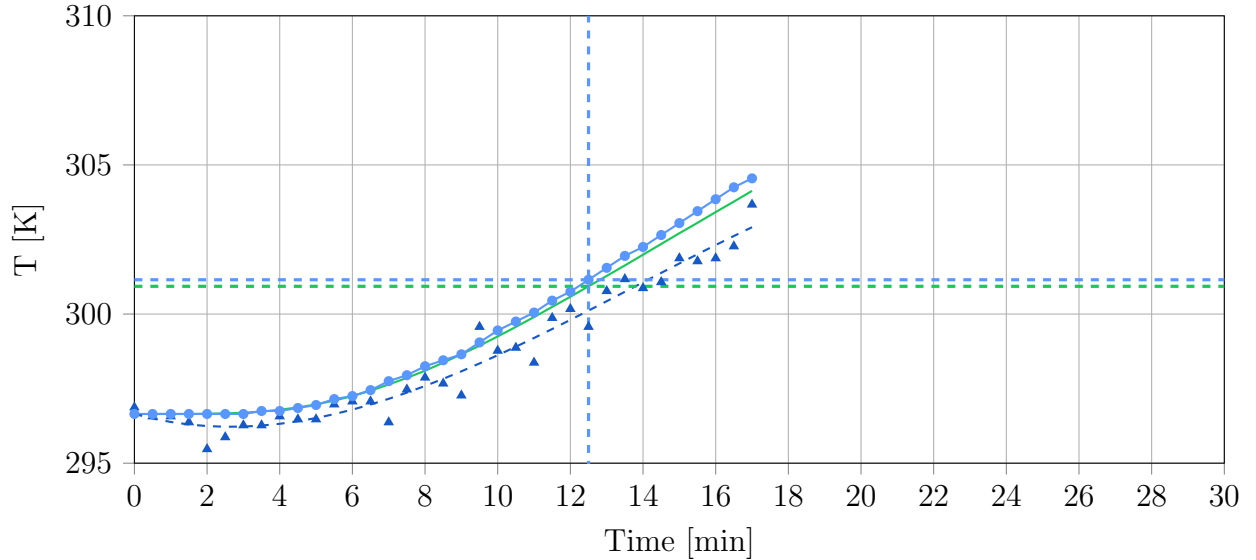


Figure D.16: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

1.1: Alu, 10 mm, $d/D = 0.15$, $k = 1.5$

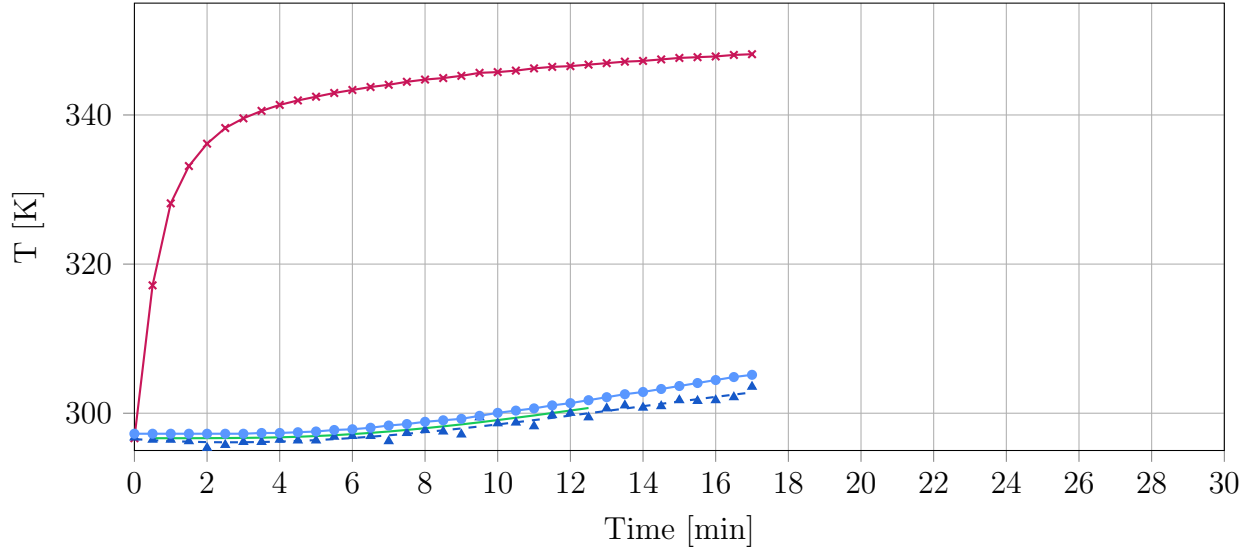


Figure D.17: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$).

1.1: Alu, 10 mm, $d/D = 0.15$, $k = 1.5$

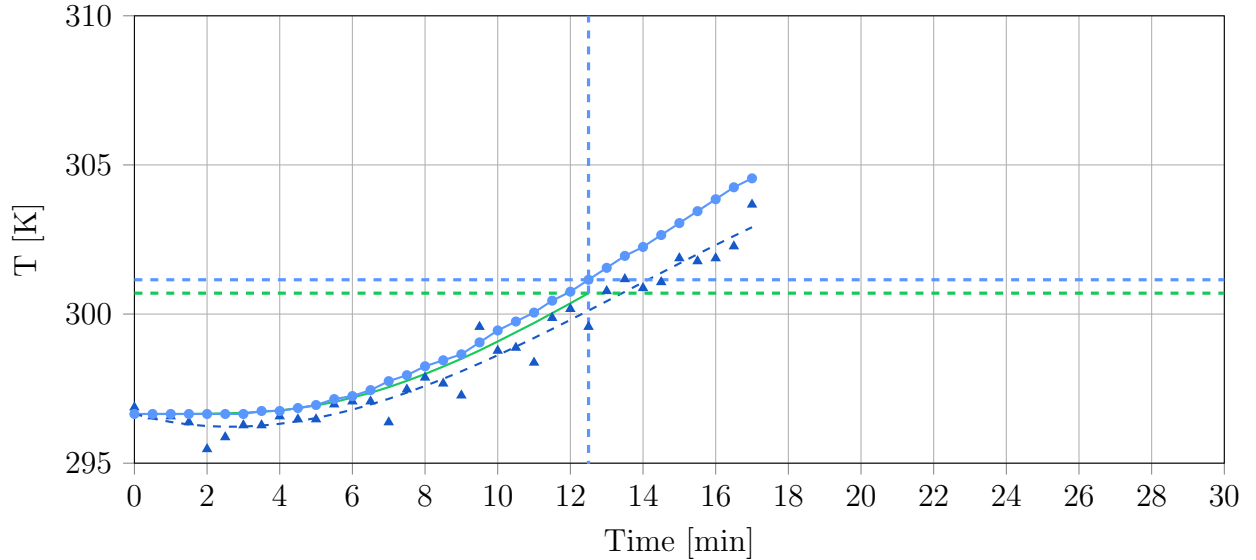


Figure D.18: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.15$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

2.1: Alu, 6 mm, $d/D = 0.2$, $k = 1.5$

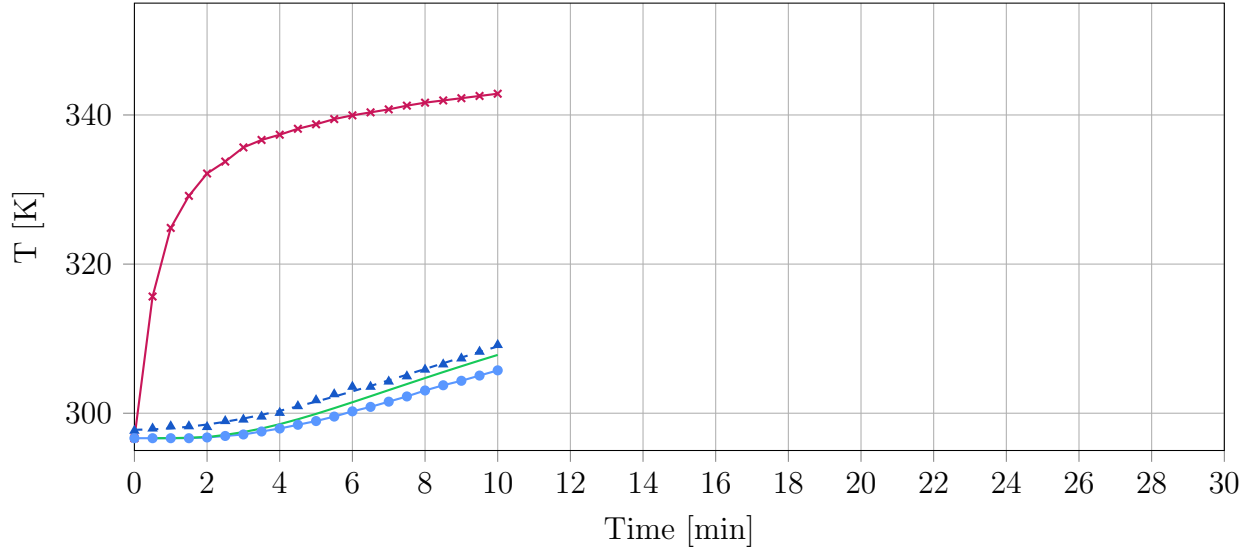


Figure D.19: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$).

2.1: Alu, 6 mm, $d/D = 0.2$, $k = 1.5$

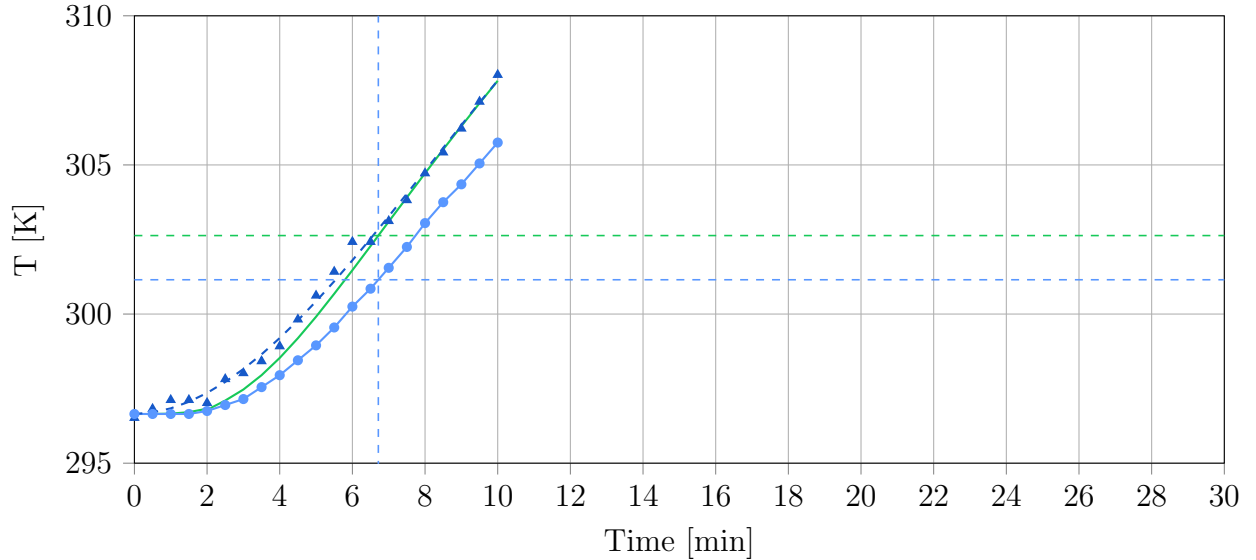


Figure D.20: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

4.1: Alu, 6 mm, $d/D = 0.15$, $k = 0.5$

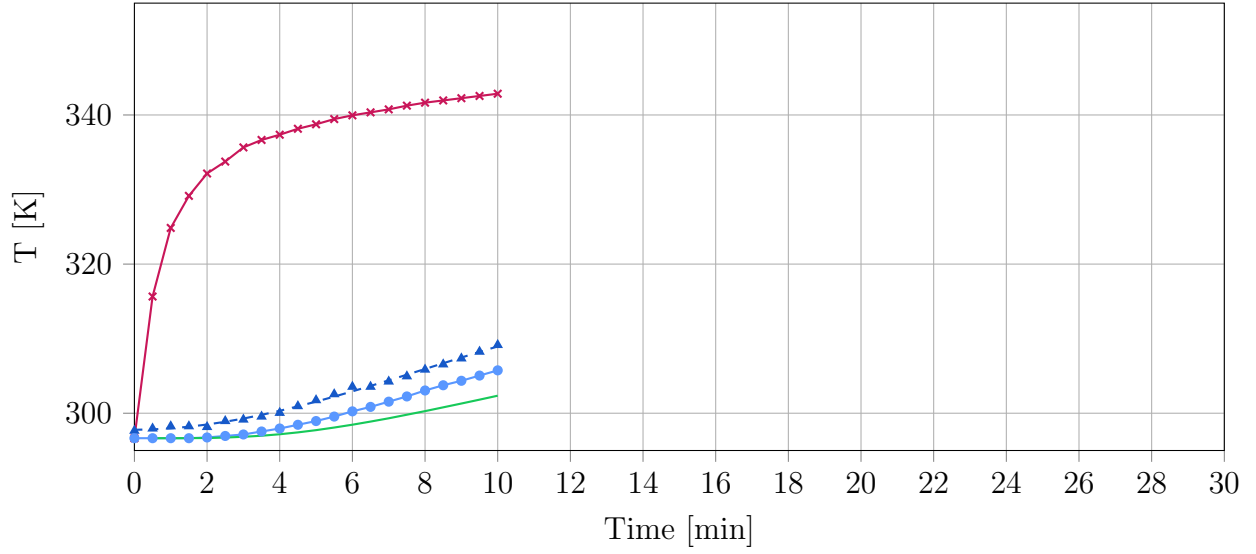


Figure D.21: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$).

4.1: Alu, 6 mm, $d/D = 0.15$, $k = 0.5$

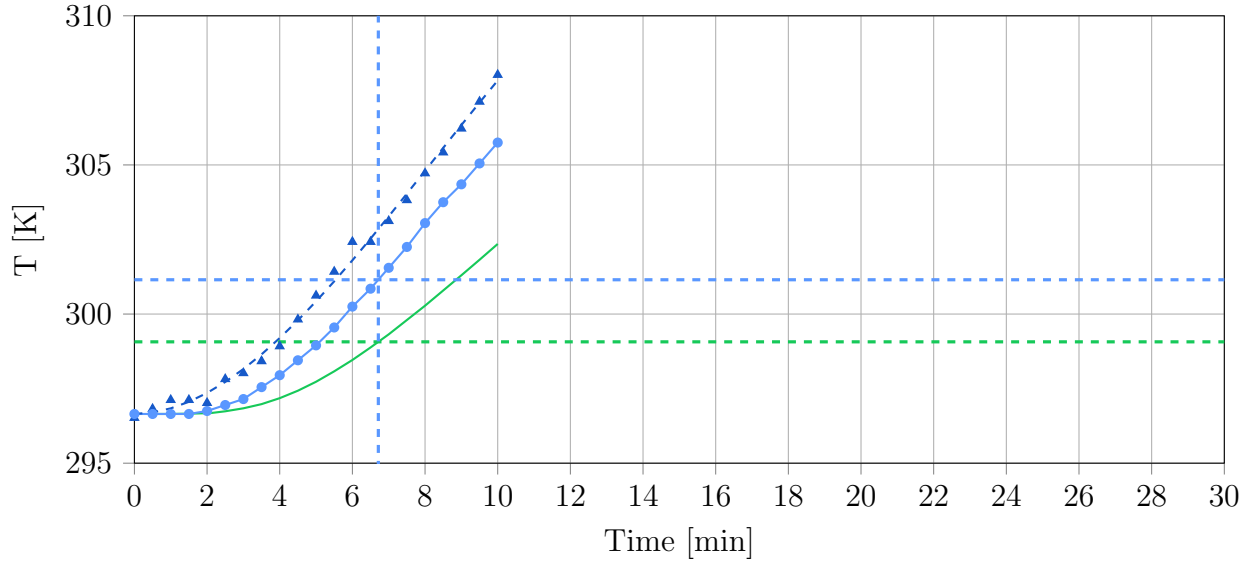


Figure D.22: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 619 aluminium spheres ($d = 6$ mm) using the bridges method ($d/D = 0.15$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

7.1: Alu, 10 mm, $d/D = 0.2$, $k = 0.5$

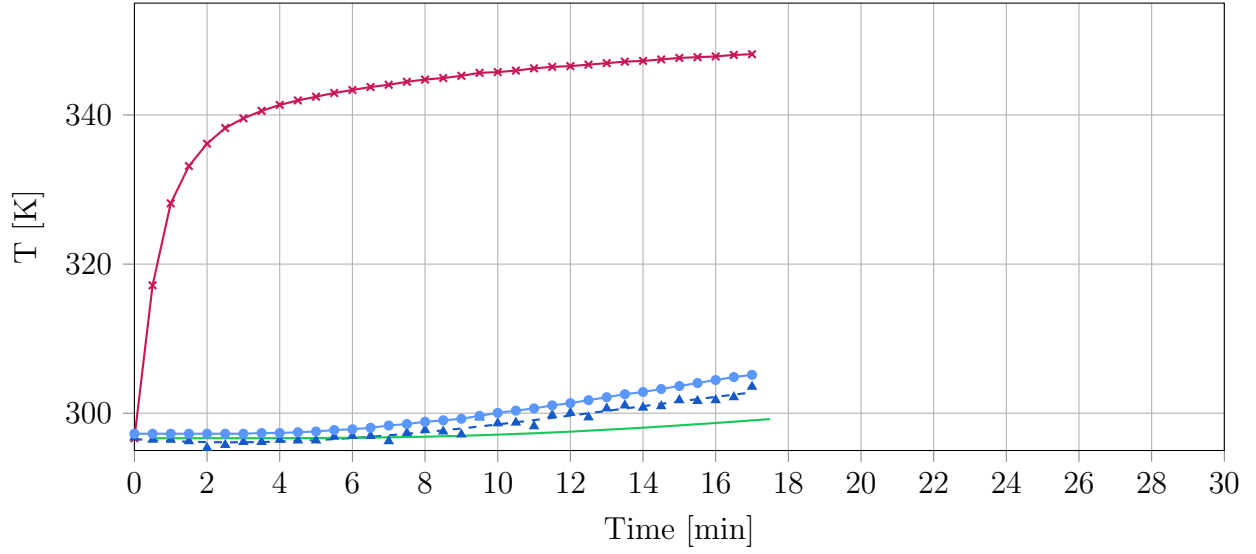


Figure D.23: Heaters temperature, surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$).

7.1: Alu, 10 mm , $d/D = 0.2$, $k = 0.5$

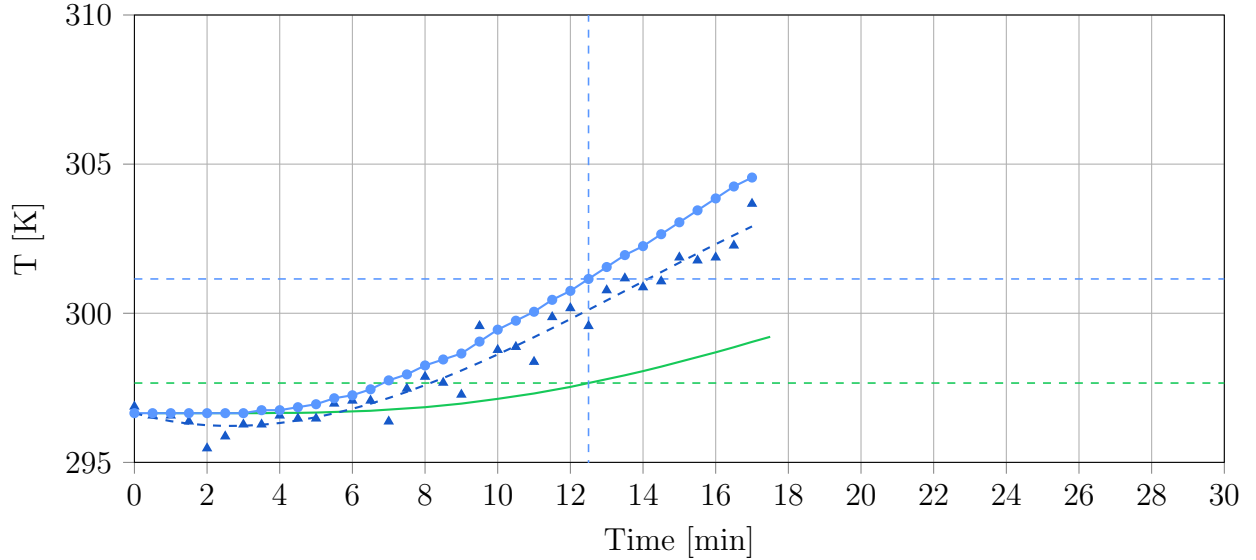


Figure D.24: Comparison of experimental surface temperature (measured using an IR-camera and a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 aluminium spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 0.5$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

E Validation Results



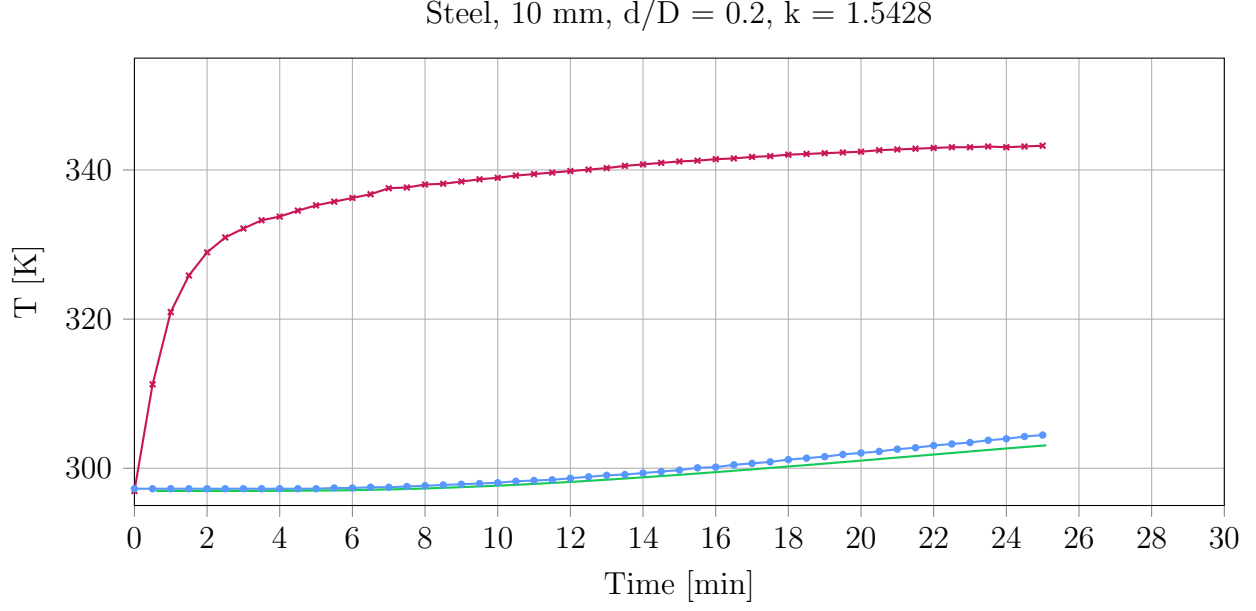


Figure E.1: Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5428$).

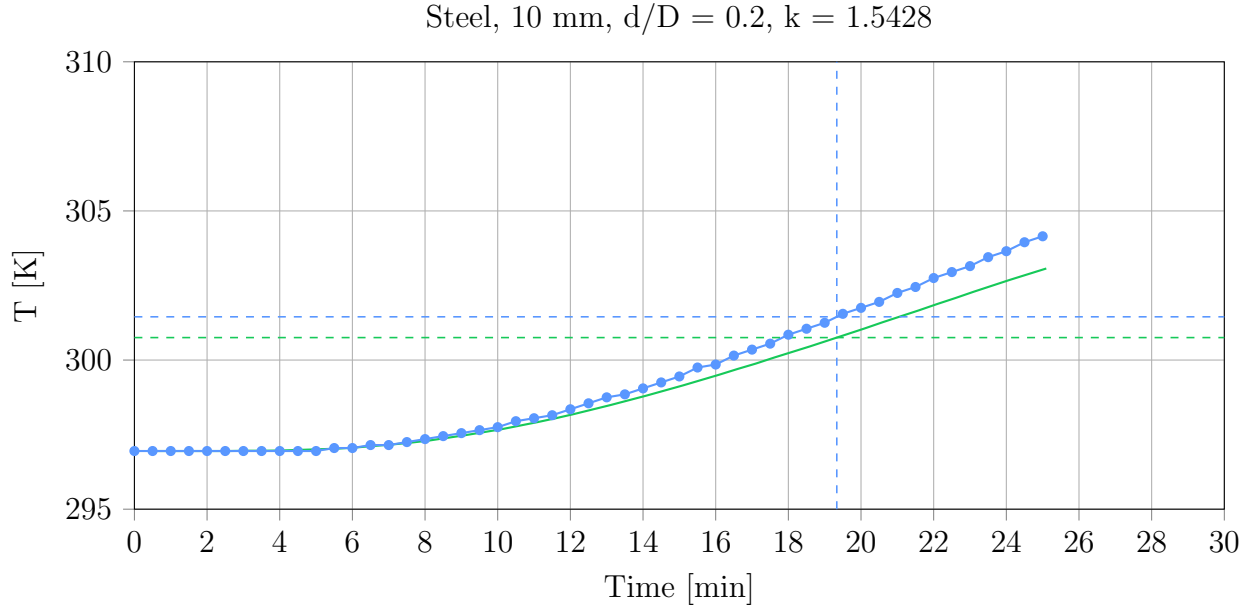


Figure E.2: Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5428$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

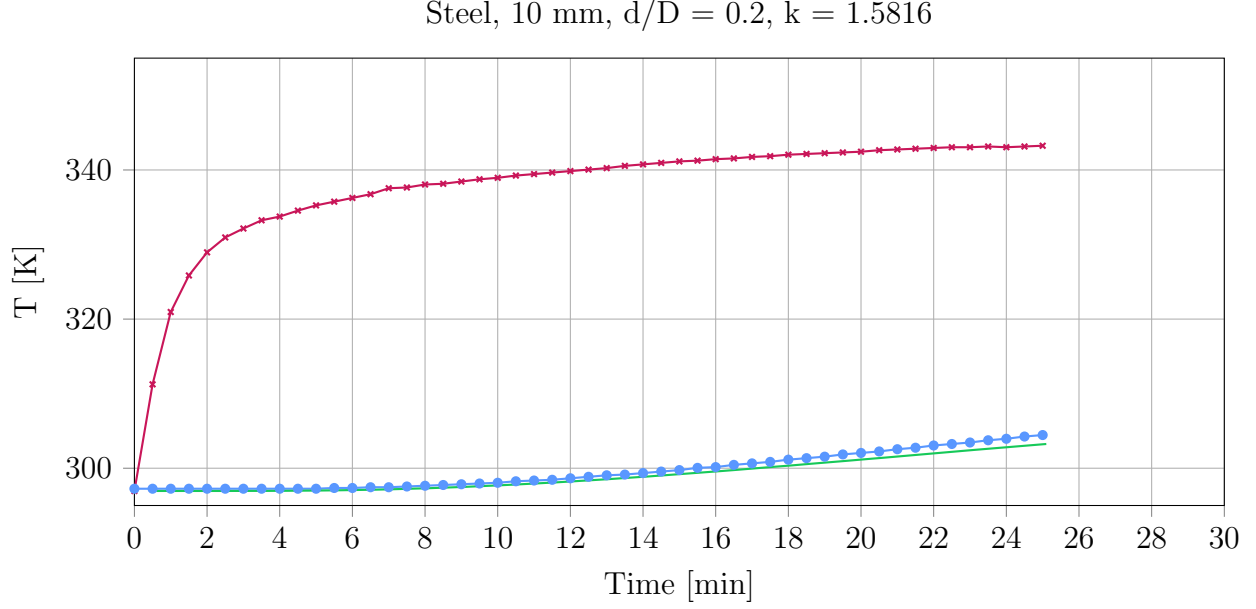


Figure E.3: Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5816$).

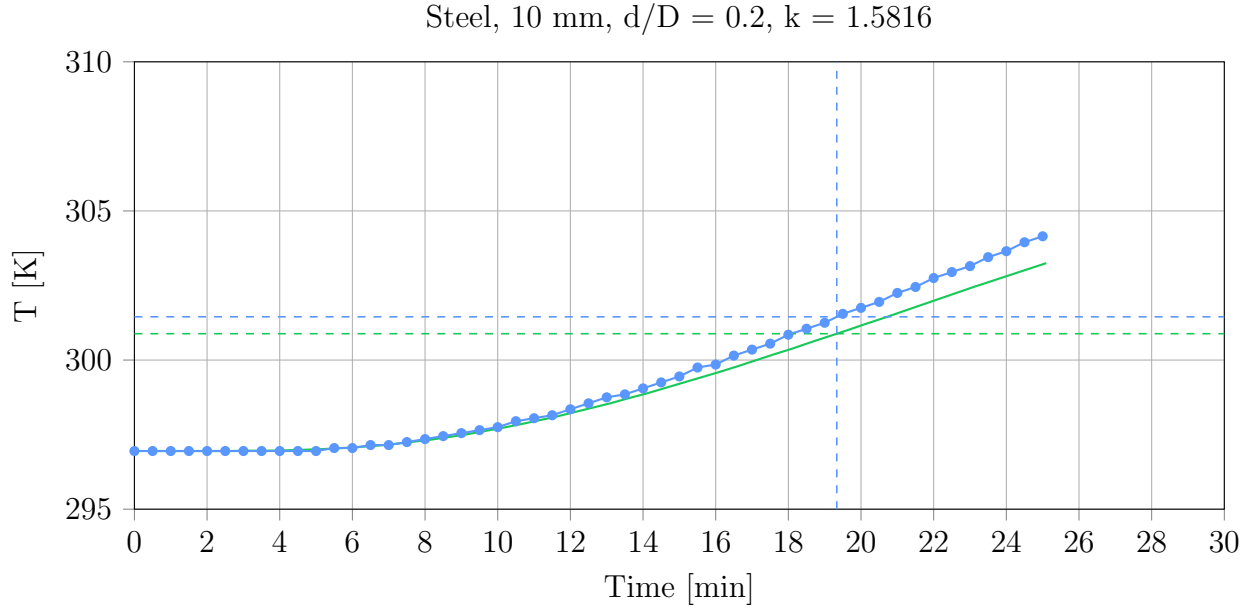


Figure E.4: Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.5816$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

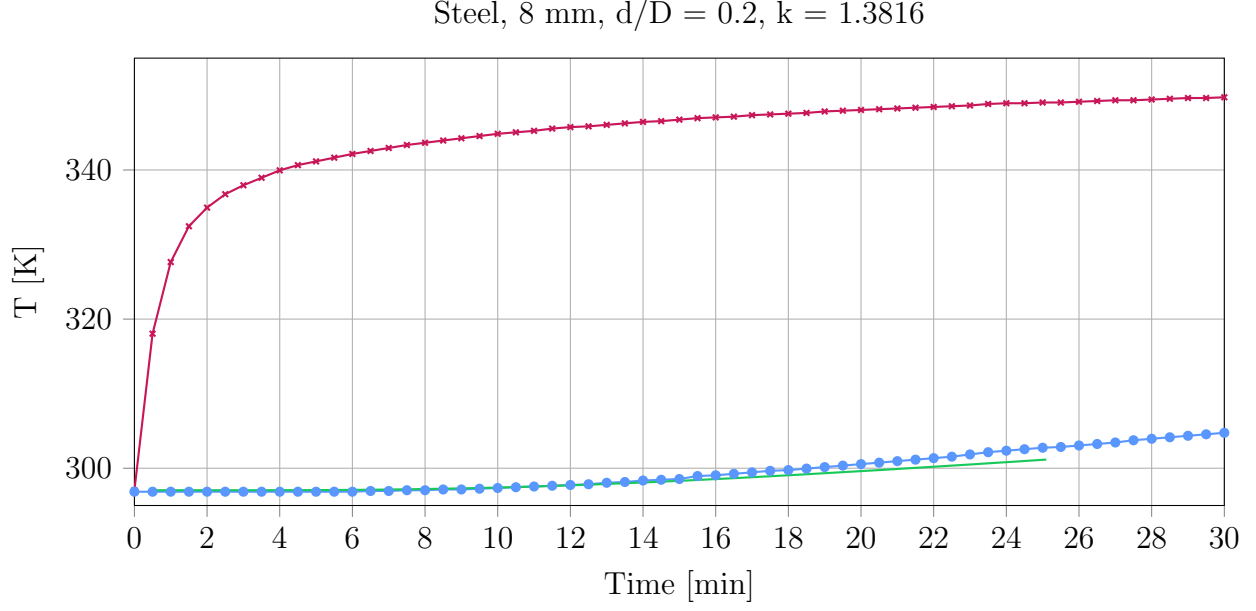


Figure E.5: Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 496 steel spheres ($d = 8$ mm) using the bridges method ($d/D = 0.2$, $k = 1.3816$).

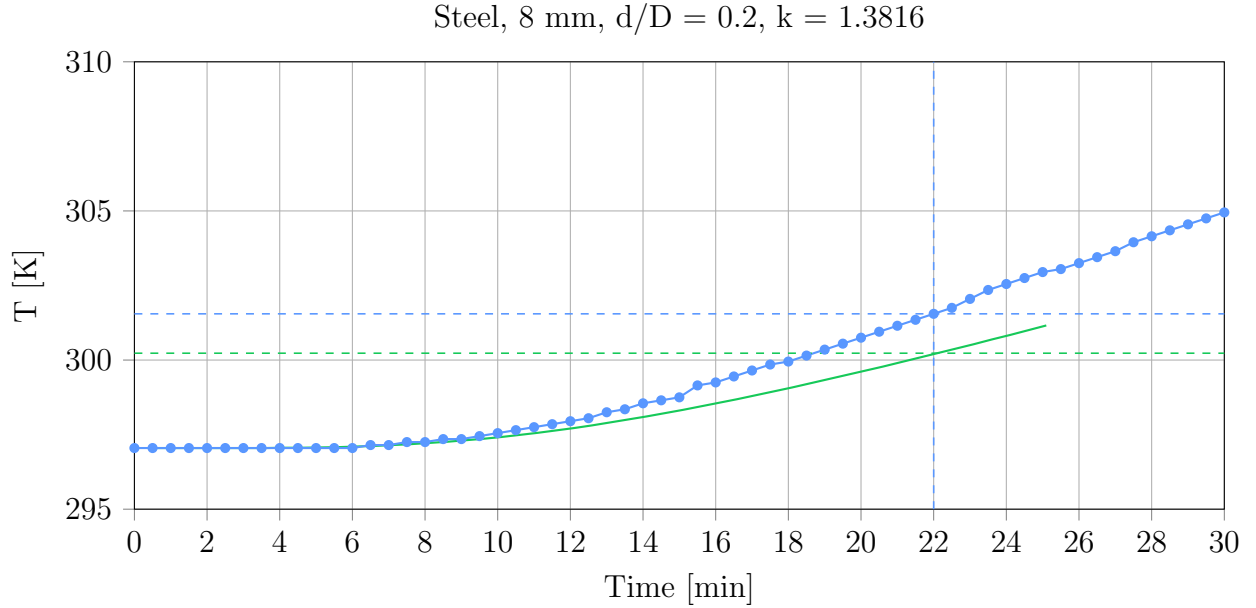


Figure E.6: Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 496 steel spheres ($d = 8$ mm) using the bridges method ($d/D = 0.2$, $k = 1.3816$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

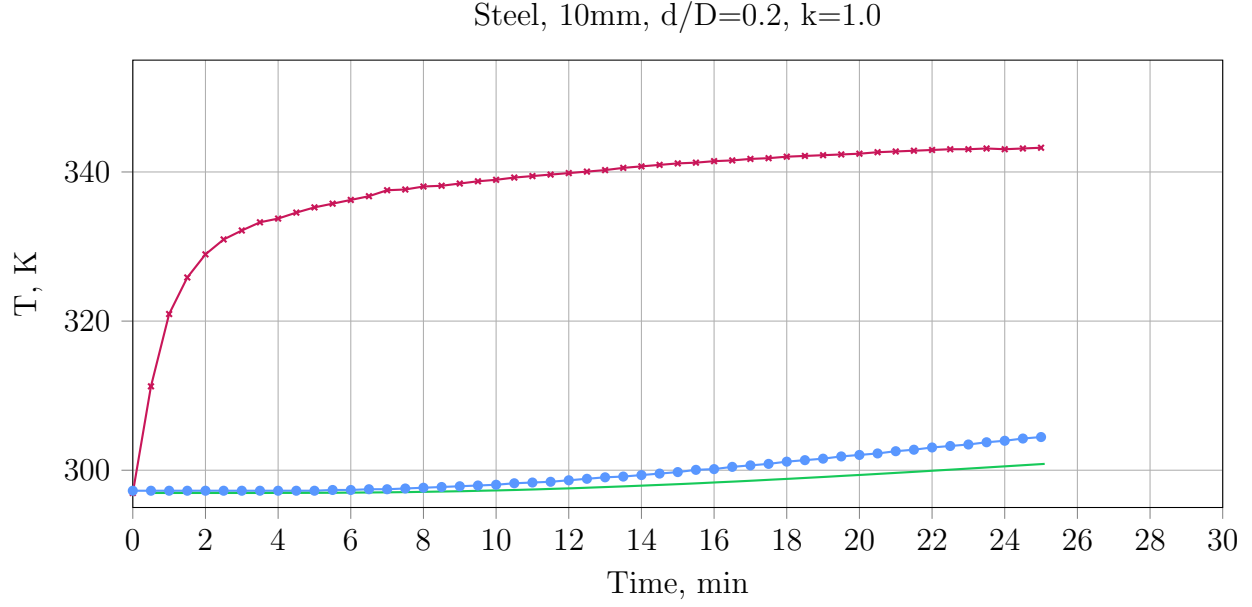


Figure E.7: Heaters temperature, surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.0$).

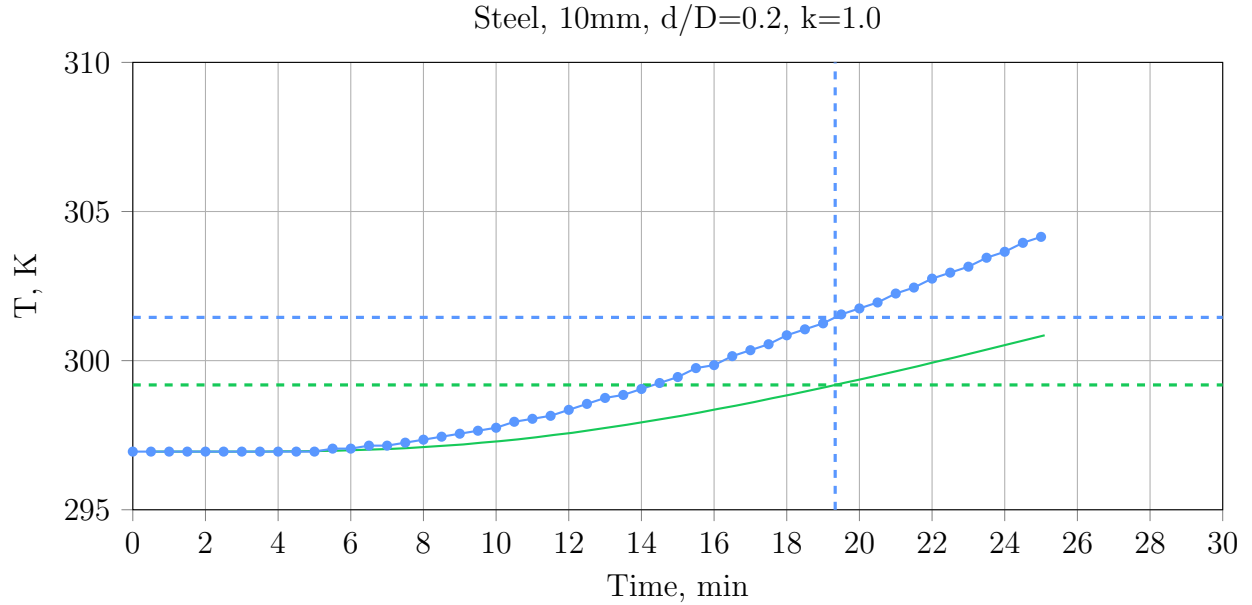


Figure E.8: Comparison of experimental surface temperature (measured using a PT1000 temperature probe) and CFD-simulated surface temperature of a packing consisting of 200 steel spheres ($d = 10$ mm) using the bridges method ($d/D = 0.2$, $k = 1.0$). The temperatures of CFD-simulation and experiment at $T_{exp}(t) - T_{exp}(t = 0) = 4.5$ K are marked with dashed lines.

F SALOME-code for bridged sphere packing

```
1 import salome
2 salome.salome_init()
3 theStudy = salome.myStudy
4
5 import GEOM
6 from salome.geom import geomBuilder
7 geompy = geomBuilder.New(theStudy)
8
9 import SMESH, SALOMEDS
10 from salome.smesh import smeshBuilder
11 smesh = smeshBuilder.New(salome.myStudy)
12
13 import os
14 import math
15
16 gg = salome.ImportComponentGUI("GEOM")
17
18
19 #-----#
20 # Input
21
22 inputFileLocation = "inputFileLocation"
23 outputLocation = "outputFileLocation"
24
25 # Scaling factor (1000 for m in mm)
26 f = 1000
27
28 # Define 2 Points to create a box for wall/sphere contacts
29 p1_box = geompy.MakeVertex(-0.0425*f, -0.0257*f, -0.200*f)
30 p2_box = geompy.MakeVertex(0.0425*f, 0.0257*f, 0.0*f)
31
32 # p1_box = geompy.MakeVertex(-0.01*f, -0.01*f, -0.08*f)
33 # p2_box = geompy.MakeVertex(0.01*f, 0.01*f, 0.0*f)
34
35 # Define 2 points, radius and height to create a cylinder for wall/sphere
    contacts
36 p1_cyl = geompy.MakeVertex(0.0*f, 0.0*f, 0.001*f) # base of cylinder
37 p2_cyl = geompy.MakeVertex(0.0*f, 0.0*f, 0.300*f) # 2nd point for vector
38 r_cyl = 0.0158 * f
39 h_cyl = 0.3 * f
40
41 # Defines BridgeRadius/ParticleRadius ratio for sphere-sphere (SS) and wall-
    sphere (WS) contact points
42 rbSS_rp = 0.2
43 rbWS_rp = 0.2
44 rbWS_rp_heater = 0.2
45
46 # Defines the minimum distance at contact-points where bridges should be
    inserted
47 contactDistance = 1e-4
48 contactDistance = contactDistance * f
49
50 # Do you want to create a boundary box?
```

```

51 createBox = "yes"          # "yes" for box, else: cylinder
52
53 # Do you want to save bridges and capped spheres as STEP or STL?
54 createSTEP_B = "no"       # "yes" to activate, standard: "yes"
55 createSTL_B = "yes"       # "yes" to activate, standard: "yes"
56
57 # Do you want to save linked spheres as STEP or STL?
58 createSTEP = "no"         # "yes" to activate, standard: "yes"
59 createSTL = "no"          # "yes" to activate, standard: "yes"
60
61 # do you want to get more info about what the code is doing right now?
62 moreInfo = "yes"          # "yes" to activate, standard: "no"
63
64 #-----#
65 createBox = createBox.lower()
66 createSTL_B = createSTL_B.lower()
67 createBoundary = createBoundary.lower()
68 createSTEP = createSTEP.lower()
69 createSTL = createSTL.lower()
70 logFile = os.path.join(outputLocation, "logFile")
71 myfile = open(logFile, 'w')
72
73 # Import spheres
74
75 print "\nStep 1: Importing spheres..."
76 myfile.write("Step 1: Importing spheres...")
77
78 vertexes = []
79 spheres = []
80 centres = []
81
82 infile = open(inputFileName, 'r')
83 i = 0
84 for line in infile:
85     if i != 0 :
86         s = line.split()
87         [x, y, z] = [ float(s[2]) * f, float(s[3]) * f, float(s[4]) * f ]
88         radius = float(s[5]) * 0.9995 * f
89
90         vertex = geompy.MakeVertex(x, y, z)
91         vertexes.append(vertex)
92
93         sphere = geompy.MakeSpherePntR(vertex, radius)
94
95         spheres.append(sphere)
96         centres.append(vertex)
97     i = i + 1
98
99 spheresC = geompy.MakeCompound(spheres)
100 geompy.addToStudy(spheresC, "spheres")
101
102 print "Number of imported spheres = ", len(spheres)
103 myfile.write("\nNumber of imported spheres = " + str(len(spheres)))
104
105 #-----#

```

```

106 # Define Bridges
107
108 d_p = 2 * radius # m
109 rbSS = rbSS_rp * radius
110 rbWS = rbWS_rp * radius
111 rbWS_heater = rbWS_rp_heater * radius
112
113 print "sphere/sphere bridges: bridge to particle radius ratio = ", rbSS_rp, "
    => bridge radius = ", rbSS
114 print "wall/sphere bridges: bridge to particle radius ratio = ", rbWS_rp, "=>
    bridge radius = ", rbWS
115 print "heater/sphere bridges: bridge to particle radius ratio = ",
    rbWS_rp_heater, "=> bridge radius = ", rbWS_heater
116
117 myfile.write("\nsphere/sphere bridges: bridge to particle radius ratio = " +
    str(rbSS_rp) + "=> bridge radius = " + str(rbSS))
118 myfile.write("\nwall/sphere bridges: bridge to particle radius ratio = " + str(
    rbWS_rp) + "=> bridge radius = " + str(rbWS))
119 myfile.write("\nheater/sphere bridges: bridge to particle radius ratio = " +
    str(rbWS_rp_heater) + "=> bridge radius = " + str(rbWS_heater))
120
121 #-----#
122 # Create box
123 if createBox == "yes":
124     print "\nStep 2: Creating boundary box ..."
125     myfile.write("\n\nStep 2: Creating boundary box ...")
126
127     box = geompy.MakeBoxTwoPnt(p1_box, p2_box)
128     geompy.addToStudy(box, "boundary box")
129     boxFaces1 = geompy.SubShapeAll(box, geompy.ShapeType["FACE"])[0]
130     boxFaces2 = geompy.SubShapeAll(box, geompy.ShapeType["FACE"])[1]
131     boxFaces3 = geompy.SubShapeAll(box, geompy.ShapeType["FACE"])[2]
132     boxFaces4 = geompy.SubShapeAll(box, geompy.ShapeType["FACE"])[3]
133     boxFaces5 = geompy.SubShapeAll(box, geompy.ShapeType["FACE"])[4]
134     boxFaces6 = geompy.SubShapeAll(box, geompy.ShapeType["FACE"])[5]
135     # geompy.addToStudy(boxFaces1, "boxFaces1")
136     # geompy.addToStudy(boxFaces2, "boxFaces2")
137     # geompy.addToStudy(boxFaces3, "boxFaces3")
138     # geompy.addToStudy(boxFaces4, "boxFaces4")
139     # geompy.addToStudy(boxFaces5, "boxFaces5")
140     # geompy.addToStudy(boxFaces6, "boxFaces6")
141
142     Faces = [boxFaces1, boxFaces2, boxFaces3, boxFaces4, boxFaces5, boxFaces6]
143
144     box_shell2 = geompy.SubShapeAll(box, geompy.ShapeType["SHELL"])
145     box_shell = geompy.MakeCompound(box_shell2)
146
147     nrHeater = 4
148
149 # Create Cylinder
150 else:
151     print "\nStep 2: Creating boundary cylinder ..."
152     myfile.write("\n\nStep 2: Creating boundary cylinder ...")
153
154     vector_cyl = geompy.MakeVector(p1_cyl, p2_cyl)

```

```

155     cyl = geompy.MakeCylinder(p1_cyl, vector_cyl, r_cyl, h_cyl)
156     geompy.addToStudy(cyl, "boundary cylinder")
157     cylFaces1 = geompy.SubShapeAll(cyl, geompy.ShapeType["FACE"])[0]
158     cylFaces2 = geompy.SubShapeAll(cyl, geompy.ShapeType["FACE"])[1]
159     cylFaces3 = geompy.SubShapeAll(cyl, geompy.ShapeType["FACE"])[2]
160
161     Faces = [cylFaces1, cylFaces2, cylFaces3]
162     geompy.addToStudy(cylFaces1, "cylFaces1")
163     geompy.addToStudy(cylFaces2, "cylFaces2")
164     geompy.addToStudy(cylFaces3, "cylFaces3")
165
166     cyl2 = geompy.MakeCylinder(p1_cyl, vector_cyl, r_cyl*1.5, h_cyl)
167     cyl_hollow = geompy.MakeCut(cyl2, cyl)
168     geompy.addToStudy(cyl_hollow, "hollow cylinder")
169
170     nrHeater = 2
171
172     #-----#
173     # Creating sphere/sphere bridges
174     print "\nStep 3: Creating bridges..."
175     print "Creating sphere/sphere bridges..."
176
177     myfile.write("\n\nStep 3: Creating bridges...")
178     myfile.write("\n\nCreating sphere/sphere bridges...")
179
180     nbSpheres = len(spheres)
181     bridges_SS_List = []
182     bridgesForCap_SS_List = []
183     bridgesToFuse = []
184     contactPoints = []
185
186     for i in range(nbSpheres):
187         sphere1 = spheres[i]
188         sphere1Face = geompy.SubShapeAll(sphere1, geompy.ShapeType["FACE"])[0]
189
190         print i + 1, "/", len(spheres)
191
192         for j in range(i, nbSpheres):
193             if i != j:
194                 sphere2 = spheres[j]
195                 distance = geompy.MinDistance(sphere1, sphere2)
196
197                 if distance <= contactDistance:
198
199                     [x1, y1, z1, x2, y2, z2] = geompy.ClosestPoints(sphere1,
200 sphere2)[1]
201                     [x, y, z] = [(x1 + x2)/2, (y1 + y2)/2, (z1 + z2)/2]
202
203                     contactPoint = geompy.MakeVertex(x, y, z)
204                     contactPoints.append(contactPoint)
205                     #normalVector = geompy.GetNormal(sphere1Face, contactPoint)
206                     normalVector = geompy.MakeVector(contactPoint, vertexes[i])
207
208                     capSS = distance/2 + radius*(1 - math.sqrt(1 - rbSS_rp**2))

```

```

209         h = 2 * capSS
210         bridge_SS = geompy.MakeCylinder(contactPoint, normalVector,
rbSS, h)
211         bridge_SS = geompy.MakeTranslationVectorDistance(bridge_SS,
normalVector, -h/2)
212         bridge_SS = geompy.MakeRotation(bridge_SS, normalVector, 45)
213         bridges_SS_List.append(bridge_SS)
214
215         bridgeForCap_SS_List = geompy.MakeCylinder(contactPoint,
normalVector, rbSS*1.2, h)
216         bridgeForCap_SS_List = geompy.MakeTranslationVectorDistance(
bridgeForCap_SS_List, normalVector, -h/2)
217         bridgeForCap_SS_List = geompy.MakeRotation(
bridgeForCap_SS_List, normalVector, 45)
218         bridgesForCap_SS_List.append(bridgeForCap_SS_List)
219
220         if createSTL == "yes" or createSTEP == "yes":
221             h = 2 * rbSS
222             bridgeToFuse = geompy.MakeCylinder(contactPoint,
normalVector, rbSS, h)
223             bridgeToFuse = geompy.MakeTranslationVectorDistance(
bridgeToFuse, normalVector, -h/2)
224             bridgeToFuse = geompy.MakeRotation(bridgeToFuse,
normalVector, 45)
225             bridgesToFuse.append(bridgeToFuse)
226
227         if moreInfo == "yes":
228             print "Inserting bridge from particle ", i+1, " to
particle ", j+1, "; contact distance = ", distance
229             myfile.write("\nInserting bridge from particle " + str(i
+1)+ " to particle " + str(j+1) + "; contact distance = " + str(distance))
230
231     print "\nNumber of sphere/sphere contact points= ", len(bridgesForCap_SS_List)
232     myfile.write("\n\nNumber of sphere/sphere contact points= " + str(len(
bridgesForCap_SS_List)))
233
234     bridges_SS = geompy.MakeCompound(bridges_SS_List)
235     geompy.addToStudy(bridges_SS, "bridges sphere/sphere")
236
237     bridgesForCap_SS = geompy.MakeCompound(bridgesForCap_SS_List)
238     geompy.addToStudy(bridgesForCap_SS, "bridges sphere/sphere for cap")
239
240     if createSTL == "yes" or createSTEP == "yes":
241         bridgesToFuse_SS = geompy.MakeCompound(bridgesToFuse)
242         #geompy.addToStudy(bridgesToFuse_SS, "bridges sphere/sphere to fuse")
243
244     # creating wall/sphere bridges
245     print "\nCreating wall/sphere bridges..."
246     myfile.write("\n\nCreating wall/sphere bridges...")
247
248     nbSpheres2 = len(spheres)
249     nFaces = len(Faces)
250     bridges_WS_List = []
251     bridges_WSHeater_List = []
252     bridgesForCap_toNonHeater_List = []

```



```

253 bridgesForCap_toHeater_List= []
254 bridgesToFuse2 = []
255 contactPoints2 = []
256
257 for i in range(nbSpheres2):
258     sphere = spheres[i]
259     sphereFace = geompy.SubShapeAll(sphere, geompy.ShapeType["FACE"])[0]
260
261     print i + 1, "/", len(spheres)
262
263     for j in range(nFaces):
264         FaceBox = Faces[j]
265         distance = geompy.MinDistance(sphere, FaceBox)
266
267         if distance <= contactDistance:
268             [x, y, z] = geompy.ClosestPoints(FaceBox, sphere)[1][0:3]
269
270             contactPoint2 = geompy.MakeVertex(x, y, z)
271             contactPoints2.append(contactPoint2)
272
273             #normalVector = geompy.GetNormal(sphereFace, contactPoint2)
274             normalVector = geompy.MakeVector(contactPoint2, vertexes[i])
275             #geompy.addToStudy(normalVector, "normalVector")
276
277             # wall nrHeater = wall to heater
278             if j == nrHeater:
279                 capWS_heater = distance + radius * (1 - math.sqrt(1 -
280 rbWS_rp_heater ** 2))
281
282                 # bridges to export STEP/STL
283                 bridge_WSHeater = geompy.MakeCylinder(contactPoint2,
284 normalVector, rbWS_heater, capWS_heater)
285                 bridge_WSHeater = geompy.MakeTranslationVectorDistance(
286 bridge_WSHeater, normalVector, 0)
287                 bridge_WSHeater = geompy.MakeRotation(bridge_WSHeater,
288 normalVector, 45)
289                 bridges_WSHeater_List.append(bridge_WSHeater)
290
291                 # bridges to cap spheres (bigger radius)
292                 bridgeForCap_toHeater = geompy.MakeCylinder(contactPoint2,
293 normalVector, rbWS_heater * 1.2, capWS_heater + distance * 2)
294                 bridgeForCap_toHeater = geompy.MakeTranslationVectorDistance(
295 bridgeForCap_toHeater, normalVector, -distance * 2)
296                 bridgeForCap_toHeater = geompy.MakeRotation(
297 bridgeForCap_toHeater, normalVector, 45)
298                 bridgesForCap_toHeater_List.append(bridgeForCap_toHeater)
299
300                 if createSTL == "yes" or createSTEP == "yes":
301                     # bridges to fuse with spheres (longer)
302                     bridgesToFuse = geompy.MakeCylinder(contactPoint2,
303 normalVector, rbWS_heater, rbWS_heater)
304                     bridgesToFuse = geompy.MakeTranslationVectorDistance(
305 bridgesToFuse, normalVector, 0)
306                     bridgesToFuse = geompy.MakeRotation(bridgesToFuse,
307 normalVector, 45)

```

```

298         bridgesToFuse2.append(bridgesToFuse)
299
300     # all the other walls
301     else:
302         capWS = distance + radius * (1 - math.sqrt(1 - rbWS_rp ** 2))
303
304         # bridges to export STEP/STL
305         # -capWS instead of 0 if usend GetNormal for NormalVector
306         bridge_WS = geompy.MakeCylinder(contactPoint2, normalVector,
307         rbWS, capWS)
308         bridge_WS = geompy.MakeTranslationVectorDistance(bridge_WS,
309         normalVector, 0)
310         bridge_WS = geompy.MakeRotation(bridge_WS, normalVector, 45)
311         bridges_WS_List.append(bridge_WS)
312
313         # bridges to cap spheres (bigger radius)
314         bridgeForCap = geompy.MakeCylinder(contactPoint2, normalVector
315         , rbWS * 1.2, capWS + distance * 2)
316         bridgeForCap = geompy.MakeTranslationVectorDistance(
317         bridgeForCap, normalVector, -distance * 2)
318         bridgeForCap = geompy.MakeRotation(bridgeForCap, normalVector,
319         45)
320         bridgesForCap_toNonHeater_List.append(bridgeForCap)
321
322         if createSTL == "yes" or createSTEP == "yes":
323             # bridges to fuse with spheres (longer)
324             bridgesToFuse = geompy.MakeCylinder(contactPoint2,
325             normalVector, rbWS, rbWS)
326             bridgesToFuse = geompy.MakeTranslationVectorDistance(
327             bridgesToFuse, normalVector, 0)
328             bridgesToFuse = geompy.MakeRotation(bridgesToFuse,
329             normalVector, 45)
330             bridgesToFuse2.append(bridgesToFuse)
331
332         if moreInfo == "yes":
333             print "Inserting bridge from particle ", i+1, " to wall " , j,
334             "; contact distance = ", distance
335             myfile.write("\nInserting bridge from particle " + str(i+1) +
336             " to wall " + str(j) + "; contact distance = " + str(distance))
337
338     print "\nNumber of non-heater wall/sphere contact points = ", len(
339     bridges_WS_List)
340     print "Number of heater wall/sphere contact points = ", len(
341     bridges_WSHeater_List)
342
343     myfile.write("\n\nNumber of non-heater wall/sphere contact points = " + str(
344     len(bridges_WS_List)))
345     myfile.write("\nNumber of heater wall/sphere contact points = " + str(len(
346     bridges_WSHeater_List)))
347
348     # Merging wall/sphere bridges
349     bridges_WS = geompy.MakeCompound(bridges_WS_List)
350     geompy.addToStudy(bridges_WS, "bridges wall/sphere to non-heater-wall")
351     bridges_WSHeater = geompy.MakeCompound(bridges_WSHeater_List)
352     geompy.addToStudy(bridges_WSHeater, "bridges wall/sphere to heater")

```

```

339 |
340 | # Merging wall/sphere bridges for cap
341 | bridgesForCap_toNonHeater = geompy.MakeCompound(bridgesForCap_toNonHeater_List
    | )
342 | geompy.addToStudy(bridgesForCap_toNonHeater, "bridges wall/sphere for cap")
343 | bridgesForCap_toHeater = geompy.MakeCompound(bridgesForCap_toHeater_List)
344 | geompy.addToStudy(bridgesForCap_toHeater, "bridges wall/sphere to heater for
    | cap")
345 | bridgesForCap_WS = geompy.MakeFuse(bridgesForCap_toNonHeater,
    | bridgesForCap_toHeater)
346 |
347 | if createSTL == "yes" or createSTEP == "yes":
348 |     bridgesToFuse_WS = geompy.MakeCompound(bridgesToFuse2)
349 |     #geompy.addToStudy(bridgesToFuse_WS, "bridges wall/sphere to fuse")
350 |
351 | #-----#
352 | # Cut
353 | print "\nStep 4: Creating capped and linked spheres..."
354 | myfile.write("\n\nStep 4: Creating capped and linked spheres...")
355 |
356 | print "Cutting spheres with wall/sphere bridges..."
357 | myfile.write("\nCutting spheres with wall/sphere bridges...")
358 | capSpheres_WS = geompy.MakeCut(spheresC, bridgesForCap_WS)
359 | #geompy.addToStudy(capSpheres_WS, "capSpheres_WS")
360 |
361 | print "Cutting spheres with sphere/sphere bridges..."
362 | myfile.write("\nCutting spheres with sphere/sphere bridges...")
363 | capSpheres = geompy.MakeCut(capSpheres_WS, bridgesForCap_SS)
364 | geompy.addToStudy(capSpheres, "capSpheres")
365 |
366 | if createBox != "yes":
367 |     print "Cutting wall/sphere bridges with hollow cylinder..."
368 |     myfile.write("\nCutting wall/sphere bridges with hollow cylinder...")
369 |     bridges_WS = geompy.MakeCut(bridges_WS, cyl.hollow)
370 |     geompy.addToStudy(bridges_WS, "bridges sphere/sphere cut")
371 |
372 | if createSTL == "yes" or createSTEP == "yes":
373 |     print "Fuse spheres with sphere/sphere bridges..."
374 |     myfile.write("\nFuse spheres with sphere/sphere bridges...")
375 |     bridgedSphere = geompy.MakeFuse(bridgesToFuse_SS, capSpheres_WS)
376 |     #geompy.addToStudy(bridgedSphere, "bridgedSphere")
377 |
378 |     print "Fuse spheres with wall/sphere bridges..."
379 |     myfile.write("\nFuse spheres with wall/sphere bridges...")
380 |     bridgedSphere2 = geompy.MakeFuse(bridgesToFuse_WS, bridgedSphere)
381 |     geompy.addToStudy(bridgedSphere2, "bridgedSphere2")
382 |
383 | #-----#
384 | # Create STEP and STL files
385 |
386 | if createSTEP_B == "yes" or createSTL_B == "yes" or createSTL == "yes" or
    | createSTEP == "yes" or createBoundary == "yes":
387 |     print "\nStep 5: Creating selected files..."
388 |     myfile.write("\n\nStep 5: Creating selected files...")
389 | else:

```

```

390     print "\nNo files are exported."
391     myfile.write("\n\nNo files are exported.")
392
393 if createSTEP_B == "yes":
394     print "Creating capped sphere STEP-file"
395     myfile.write("\nCreating capped sphere STEP-file")
396     f_step = os.path.join(outputLocation, "spherePacking_capped.step")
397     geompy.ExportSTEP(capSpheres, f_step, GEOM.LU_MILLIMETER)
398
399     print "Creating sphere/sphere bridge STEP-file"
400     myfile.write("\nCreating sphere/sphere bridge STEP-file")
401     f_step = os.path.join(outputLocation, "bridges_SS.step")
402     geompy.ExportSTEP(bridges_SS, f_step, GEOM.LU_MILLIMETER)
403
404     print "Creating non-heater-wall/sphere bridge STEP-file"
405     myfile.write("\nCreating non-heater-wall/sphere bridge STEP-file")
406     f_step = os.path.join(outputLocation, "bridges_WS_nonHeater.step")
407     geompy.ExportSTEP(bridges_WS, f_step, GEOM.LU_MILLIMETER)
408
409     print "Creating heater-wall/sphere bridge STEP-file"
410     myfile.write("\nCreating heater-wall/sphere bridge STEP-file")
411     f_step = os.path.join(outputLocation, "bridges_WS_Heater.step")
412     geompy.ExportSTEP(bridges_WSHeater, f_step, GEOM.LU_MILLIMETER)
413
414 if createSTL_B == "yes":
415     print "Creating capped sphere mesh and STL-file"
416     myfile.write("\nCreating capped sphere mesh and STL-file")
417     capSpheres_stl = smesh.Mesh(capSpheres, "spherePacking_capped mesh by
NETGEN_1D2D")
418     # create a Netgen_1D2D algorithm for solids
419     algo2D = capSpheres_stl.Triangle(smeshBuilder.NETGEN_1D2D)
420     # define hypotheses
421     n12_params = algo2D.Parameters()
422     # define fineness
423     n12_params.SetFineness(4)
424     capSpheres_stl.Compute()
425     f_stl = os.path.join(outputLocation, "spherePacking_capped.stl")
426     capSpheres_stl.ExportSTL(f_stl, ascii=1, meshPart=None)
427
428     print "Creating sphere/sphere bridge mesh and STL-file"
429     myfile.write("\nCreating sphere/sphere bridge mesh and STL-file")
430     bridges_SS_stl = smesh.Mesh(bridges_SS, "bridges_SS mesh by NETGEN_1D2D")
431     # create a Netgen_1D2D algorithm for solids
432     algo2D = bridges_SS_stl.Triangle(smeshBuilder.NETGEN_1D2D)
433     # define hypotheses
434     n12_params = algo2D.Parameters()
435     # define fineness
436     n12_params.SetFineness(4)
437     bridges_SS_stl.Compute()
438     f_stl = os.path.join(outputLocation, "bridges_SS.stl")
439     bridges_SS_stl.ExportSTL(f_stl, ascii=1, meshPart=None)
440
441     print "Creating non-heater-wall/sphere bridge mesh and STL-file"
442     myfile.write("\nCreating non-heater-wall/sphere bridge mesh and STL-file")

```

```

443     bridges_WS_stl = smesh.Mesh(bridges_WS, "bridges_WS_toNonHeater mesh by
NETGEN_1D2D")
444     # create a Netgen_1D2D algorithm for solids
445     algo2D = bridges_WS_stl.Triangle(smeshBuilder.NETGEN_1D2D)
446     # define hypotheses
447     n12_params = algo2D.Parameters()
448     # define fineness
449     n12_params.SetFineness(4)
450     bridges_WS_stl.Compute()
451     f_stl = os.path.join(outputLocation, "bridges_WS_nonHeater.stl")
452     bridges_WS_stl.ExportSTL(f_stl, ascii=1, meshPart=None)
453
454     print "Creating heater-wall/sphere bridge mesh and STL-file"
455     myfile.write("\nCreating heater-wall/sphere bridge mesh and STL-file")
456     bridgesToHeater2_stl = smesh.Mesh(bridges_WSheater, "bridges_WS_toHeater
mesh by NETGEN_1D2D")
457     # create a Netgen_1D2D algorithm for solids
458     algo2D = bridgesToHeater2_stl.Triangle(smeshBuilder.NETGEN_1D2D)
459     # define hypotheses
460     n12_params = algo2D.Parameters()
461     # define fineness
462     n12_params.SetFineness(4)
463     bridgesToHeater2_stl.Compute()
464     f_stl = os.path.join(outputLocation, "bridges_WS_Heater.stl")
465     bridgesToHeater2_stl.ExportSTL(f_stl, ascii=1, meshPart=None)
466
467     if createSTEP == "yes":
468         print "Creating bridged sphere STEP-file"
469         myfile.write("\nCreating bridged sphere STEP-file")
470         f_step = os.path.join(outputLocation, "spherePacking_bridged.step")
471         geompy.ExportSTEP(bridgedSphere2, f_step, GEOM.LU_MILLIMETER)
472
473     if createSTL == "yes":
474         print "Creating bridged sphere mesh and STL-file"
475         myfile.write("\nCreating bridged sphere mesh and STL-file")
476         f_stl = os.path.join(outputLocation, "spherePacking_bridged.stl")
477         bridgedSphere2_stl = smesh.Mesh(bridgedSphere2, "bridgedSphere mesh by
NETGEN_1D2D")
478         # create a Netgen_1D2D algorithm for solids
479         algo2D = bridgedSphere2_stl.Triangle(smeshBuilder.NETGEN_1D2D)
480         # define hypotheses
481         n12_params = algo2D.Parameters()
482         # define fineness
483         n12_params.SetFineness(4)
484         bridgedSphere2_stl.Compute()
485         bridgedSphere2_stl.ExportSTL(f_stl, ascii=1, meshPart=None)
486
487     print "\nDone"
488     myfile.write("\n\nDone")
489
490     myfile.close()

```

data/SALOMECODE.txt

G R-code for DoE

```
1 library(rsm)
2 library(conf.design)
3 library(daewr)
4 library(DoE.base)
5 library(FrF2)
6 library(scatterplot3d)
7 library(ggplot2)
8 library(colorRamps)
9 library(grDevices)
10
11 #-----#
12 #Colours for colour plot
13 jet.colors <- colorRampPalette(c("black","grey","black"))
14 jet.colors <- colorRampPalette(matlab.like(9))
15
16 #-----#
17 #Screening
18 designscreening <- FrF2(8,4,factor.names = list(D=c(6,10),dD=c(0.15,0.2),
19                                     k=c(0.5,1.5),material=c('Glass',
20                                     'Aluminium'))), randomize=FALSE)
21 design_coded <- FrF2(8,4, randomize=FALSE)
22
23 y <- c(-0.27, #Run0
24       -3.21, #Run1
25       -2.39, #Run2
26       -1.3,  #Run3
27       0.69,  #Run4
28       -0.33, #Run5
29       1.51,  #Run6
30       -0.22) #Run7
31
32 screening <- add.response(designscreening,y)
33 screening_coded <- add.response(design_coded,y)
34
35 shapiroWilk_1 <- shapiro.test(y)
36 hist(screening$y,main = expression("Histogram of screening response ("*Delta*"
37                                     "T)"),xlab = expression("Response ("*Delta*"T)"))
38
39 model_1 <- lm(y~(. )^2,data = screening)
40 model_2 <- lm(y~(. )^2,data = screening_coded)
41 summary <- summary(model_1)
42
43 MEPlot(model_1,main = expression("Main effects plot for "*Delta*"T"))
44 IAPlot(model_1,select=c(1:4),main = expression("Interaction plot for "*Delta*"
45                                     "T"),show.alias=TRUE)
46 Gaptest(screening_coded)
47 LGB(coef(model_2)[2:8], rpt=FALSE)
48
49 runsModel <- c(0:7)
50
51 plot(runsModel[1:8], y[1:8],main = expression("Response Plot ("*Delta*"T)"),
52       xlab = "Run", ylab = expression("Response ("*Delta*"T)"))
```

```

51 #-----#
52 #Screening Full Factorial Design for Alu
53 designscreeningFFD <- FrF2(8,3, factor.names = list(D=c(6,10),dD=c(0.15,0.2),k=
    c(0.5,1.5)), randomize=FALSE)
54 design_codedFFD <-FrF2(8,3, randomize=FALSE)
55
56 #Delta T = 4.5
57 y <- c(-2.08, #Run4.1
58        -3.21, #Run1
59        -2.39, #Run2
60        -3.49, #Run7.1
61        0.69, #Run4
62        -0.45, #Run1.1
63        1.48, #Run2.1
64        -0.22) #Run7
65 #Gap corrected response
66 #y <- c(-2.08, #Run4.1
67 #      -3.21, #Run1
68 #      -2.39, #Run2
69 #      -3.49, #Run7.1
70 #      0.69, #Run4
71 #      -0.45, #Run1.1
72 #      0.94, #Run2.1
73 #      -0.22) #Run7
74
75 #Add respopnse to Design
76 screeningFFD <- add.response(designscreeningFFD,y)
77 screening_codedFFD <- add.response(design_codedFFD,y)
78
79 #Shapiro-Wilk test
80 shapiroWilkFFD <- shapiro.test(y)
81 hist(screeningFFD$y,main = expression("Histogram of screening response ("*
    Delta*"T)"),xlab = expression("Response ("*Delta*"T)"))
82
83 #Fit linear 2-factor interaction model
84 model_1FFD <- lm(y~(. )^2,data = screeningFFD)
85 model_2FFD <- lm(y~(. )^2,data = screening_codedFFD)
86 summaryFFD <- summary(model_1FFD)
87
88 #Main effect plot + interaction plot
89 MEPlot(model_1FFD,main = expression("Main effects plot for "*Delta*"T"),show.
    alias=TRUE)
90 IAPlot(model_1FFD,select=c(1:3),main = expression("Interaction plot for "*
    Delta*"T"),show.alias=TRUE)
91
92 #Gap test for outliers + Half normal plot
93 Gaptest(screening_codedFFD)
94
95 #Half normal plot
96 #LGB(coef(model_2FFD)[2:8], rpt=FALSE)
97 #Normal probability plot
98 #qqnorm(coef(model_2FFD)[2:7],plot.it = TRUE)
99 #qqline(coef(model_2FFD)[2:7])
100
101 #Daniel plot

```

```

102 DanielPlot(model_1FFD, autolab = FALSE, half = FALSE)
103
104 #Plot response over runs
105 runsModel <- c(0:7)
106 plot(runsModel[1:8], y[1:8], main = expression("Response Plot (*Delta*T)"),
107       xlab = "Run", ylab = expression("Response (*Delta*T)"))
108 #Residual Plot
109 plot(runsModel[1:8], residuals(model_1FFD), main = expression("Residual Plot (
110       *Delta*T)"),
111       xlab = "Run", ylab = "Residual")
112
113 #-----#
114 #Fit linear Model for ~A+B+C+BC
115 D=c(6,10,6,10,6,10,6,10)
116 d=c(0.15,0.15,0.2,0.2,0.15,0.15,0.2,0.2)
117 k=c(0.5,0.5,0.5,0.5,1.5,1.5,1.5,1.5)
118 model_iaFFD <- lm(y ~ D + d + k + d*k)
119 model_iaFFDCoded <- lm(y ~ D + d + k + d*k, data=screeningFFD)
120
121 #Main effect plot + interaction plot
122 MEPlot(model_iaFFDCoded)
123 IAPlot(model_iaFFDCoded, select=c(1:3))
124
125 predict(model_iaFFDCoded)
126 residuals(model_iaFFDCoded)
127
128 #Residual Plot
129 plot(runsModel[1:8], residuals(model_iaFFDCoded), main = expression("Residual
130       Plot (*Delta*T)"),
131       xlab = "Run", ylab = "Residual")
132
133 #-----#
134 #Fit linear Model for ~A+C
135 model_lmFFD <- lm(y ~ D + k, data = screeningFFD)
136 summarylmFFD <- summary(model_lmFFD)
137
138 #Main effect plot + interaction plot
139 MEPlot(model_lmFFD)
140
141 predict(model_lmFFD)
142 residuals(model_lmFFD)
143
144 #Residual Plot
145 plot(runsModel[1:8], residuals(model_lmFFD), main = expression("Residual Plot (
146       *Delta*T)"),
147       xlab = "Run", ylab = "Residual")
148
149 #Fit linear Model for 3D-plot (uncoded factors!)
150 D=c(6,10,6,10,6,10,6,10)
151 k=c(0.5,0.5,0.5,0.5,1.5,1.5,1.5,1.5)
152 model_lmFFD2 <- lm(y ~ D + k)
153
154 #Plot for fixed ~A+C
155 dp3 <- seq(6,10, length.out = 30)

```



```

154 lambdaEff3 <- seq(0.5,1.5, length.out = 30)
155 matrix3 <- expand.grid(D=dp3, k=lambdaEff3)
156 # prediction from the linear model
157 matrix3$y <- predict(model_lmFFD2, newdata=matrix3)
158
159 # 3D scatterplot
160 s3dPlot <- scatterplot3d(matrix3$D, matrix3$k, matrix3$y, main = "Response Plot
      Linear Model",
161                        xlab = "dp", ylab = "lambdaEff", zlab = expression("Response Plot
      (*Delta*T)"),
162                        xlim = c(6, 10), ylim = c(0.5, 1.5), zlim = c(-3.8, 1.5), pch =
      20)
163 s3dPlot$points3d(D,k,y, pch = 8, col = "blue")
164
165 ## add blue and red residuals
166 orig <- s3dPlot$xyz.convert(D, k, y)
167 plane <- s3dPlot$xyz.convert(D, k, fitted(model_lmFFD2))
168 i.negpos <- 1 + (resid(model_lmFFD2) > 0)
169 segments(orig$x, orig$y, plane$x, plane$y,
170          col = c("blue", "red")[i.negpos], lty = 1, lwd = 2)

```

data/RCode.txt

H α_w -Model implementation (python-code)

```
1 import numpy as np
2
3
4 def heatEQNum(x, dx, dt, a, Time, Tbottom):
5
6     r = 2 #### multiplier for simulated bed hight
7     Time = Time * 60
8     dt = dt * 60
9
10    j = int(r*x/dx)
11    n = int((Time[-1, 0])/dt)
12    dTime = int(len(Time[:, 0]))
13
14    s = a*dt/dx**2
15
16    if s > 0.5:
17        print('s > 0.5, s = ', s)
18
19    else:
20        print('s = ', s)
21        #### build Matrix, fill T0
22        A = np.zeros((j, j))
23        T = np.zeros((j,1))
24
25        for i in range(1, j-1, 1):
26            A[i, i-1] = s
27            A[i, i] = 1-2*s
28            A[i, i+1] = s
29
30            T[i, 0] = Tbottom[0, 0]
31
32        #### x = 0
33        A[0, 0] = 1
34        #### x = rx
35        A[j-1, j-2] = s
36        A[j-1, j-1] = 1 - s
37        T[0, 0] = Tbottom[0, 0]
38        T[j-1, 0] = Tbottom[0, 0]
39
40        Tx = np.zeros((n+1, 1))
41        TimeR = np.zeros((n+1, 1))
42        k = np.zeros((j, 1))
43
44        t = 0
45        m = 0
46        for i in range(0, dTime-1, 1):
47            k[0, 0] = (Tbottom[i+1, 0] - Tbottom[i, 0]) / (Time[i+1, 0] - Time
[i, 0])
48            while t <= Time[i+1, 0]:
49                Tx[m, 0] = T[int(j/r)-1, 0]
50                T = np.dot(A, T) + k*dt
51                TimeR[m, 0] = t
52                t = t + dt
```

```
53         m = m + 1
54
55     TimeR = TimeR/60
56
57     return Tx, TimeR
```

data/heatEqu.txt