AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis dissertation.

_____
Date

_____
Signature

Kurzfassung

Katalytische Festbettreaktoren sind die am häufigsten industriell eingesetzte Reaktorbauform. Sie sind unentbehrlich in Prozessen bei der Rohölverarbeitung, der Chemikalien- und Pharmazeutikaproduktion, oder bei der Abluftreinigung. Unter den Festbettreaktoren stellen Schüttbett-Reaktoren, bestehend aus einem Bett kugelförmiger Katalysatorpartikel, eine Standardbauform dar. Die chemischen und physikalischen Vorgänge in diesen Reaktoren sind hoch komplex und werden durch das Zusammenspiel von örtlichen Strömungsbedingungen, Stofftransportvorgängen, der Wärmeentwicklung durch die chemische Reaktion und der Reaktionskinetik bestimmt.

Eine Analytische Beschreibung der Transportprozesse bzw. der chemischen Reaktion ist deshalb nur unter wesentlichen Vereinfachungen möglich. Eine Direkte Numerische Simulation (DNS) erlaubt hingegen eine lokal aufgelöste Berechnung von (i) Strömungsgeschwindigkeit, (ii) Druck, (iii) Temperatur, (iv) Edukt-Konzentration, sowie der (v) Reaktionsrate.

Konkret wurde in dieser Arbeit eine exotherme Flüssigphasenreaktion zur Herstellung von Acetylsalicylsäure untersucht, die an organischen Katalysatorpartikeln (Amberlite IR120) stattfindet. Durch die spezifischen Katalysatoreigenschaften kommt es im Feststoff zu einer Aufnahme an organischem Lösungsmittel, wodurch die Partikel ein Schwellverhalten zeigen. Dies hat eine Verdichtung des Partikelbetts zur Folge, was die Transportvorgänge wesentlich beeinflusst und zusätzlich die Erfassung charakteristischer geometrischer Parameter (Partikel-Volumenanteil und Oberfläche) erschwert.

Die Bettgeometrie wurde durch eine Simulation auf Basis der Discrete Element Method (DEM) ermittelt. Durch Anwendung einer Monte-Carlo Integrationsmethode konnten der Partikel-Volumenanteil sowie die der Reaktion zur Verfügung stehende Partikeloberfläche bestimmt werden. Untersuchungen am Einzelpartikel gaben Aufschluss über Wärme-Transportvorgänge im Inneren der Katalysatorpartikel und deren Einfluss auf die Reaktionsrate. Durch die anschließenden Simulationen eines Bettabschnittes konnten prozessrelevante Größen (Druckverlust, Geschwindigkeitsfelder, chemischer Umsatz,…) für verschiedene Betriebsbedingungen gewonnen werden. Ein analytisches Modell wurde mit diesen DNS-Daten kalibriert um das Verhalten derartiger Reaktoren präziser vorhersagen zu können.

Abstract

Fixed bed catalytic reactors are the most commonly used reactors in industrial practice. They are indispensable in the petrochemical industry, flue gas cleaning, or the production of bulk chemicals, fine chemicals and pharmaceuticals. A commonly used type of fixed bed reactors is a tubular reactor packed with spherical catalyst particles. The chemical and physical phenomena inside this reactors are governed by the interaction of fluid dynamics, heat production due to chemical reaction, mass transport processes as well as chemical reaction kinetics.

An analytical description of this complex problem is only possible under substantial simplifications of the system, and in general cannot predict local phenomena in the particle bed. Fortunately, Direct Numerical Simulations (DNS) nowadays allow a locally resolved calculation of (i) fluid flow velocity, (ii) pressure, (iii) temperature, (iv) species concentration and (v) reaction rate inside the particle bed.

As a model reaction, an exothermic esterification to produce acetylsalicylic acid was investigated. This reaction takes place on the surface of an organic catalyst (Amberlite IR120). It was found that due to the organic nature of the catalyst particles, solvent can diffuse into the solid phase which causes particle swelling. The swelling leads to a significant bed compaction which (i) on the one hand influences the transport processes, and (ii) on the other hand complicates the computing of characteristic geometric bed properties such as the particle volume fraction or the surface area.

The particle bed geometry was simulated applying a Discrete Elements Method (DEM). By Introducing a Monte-Carlo integration method it was possible to determine the particle volume fraction and the (for chemical reaction available) particle surface area. Investigations on a single sphere were used to quantify heat transfer outside and inside the catalyst particles, and its influence on the reaction rate. In subsequent DNS studies the reactive flow inside a short section of the particle bed was investigated and process relevant quantities (basically pressure drop and conversion) were computed for different reactor operation conditions. An analytical model was calibrated by use of the DNS data which makes it possible to precisely predict the overall reactor performance of packed bed reactors.

Acknowledgement

I owe my sincerest gratitude to all who supported me while writing this thesis.

First of all, I would like to thank my supervisor Ass.Prof. Dipl-Ing. Dr.techn. Stefan Radl for his professional guidance and support.

Further, I want to thank the leader of the project within I wrote this thesis, Ass.Prof. Dipl-Ing. Dr.techn. Heidrun Gruber-Wölfler, for providing important input.

Next, I want to thank the laboratory team and research assistants of the IPPT as well as the RCPE for performing auxiliary measurements and simulations.

Last but not least, I want to thank my family and friends who encouraged me throughout my years of study.

Table of Content

List of Figures

List of Tables

List of Symbols

A	[m²]	cross-sectional area
a	[m²/s]	thermal diffusivity
Bi	[-]	Biot number
c_i	[kg$_i$/m³]	mass concentration of species i
C_i	[kg$_i$/kg]	mass fraction of species i
c_p	[J/kg K]	specific heat capacity
d_{32}	[m]	Sauter mean diameter of a particle population
d_p	[m]	particle diameter
d_r	[m]	tube diameter
D_A	[m²/s]	diffusion coefficient of species A in fluid
Da	[-]	Damköhler number adapted for reactive flow in a particle bed
Da_I	[-]	Damköhler number (conventional formulation)
E_a	[J/mol]	activation energy for chemical reaction
E_{therm}	[-]	ratio of thermal energy over kinetic energy
Eu	[-]	Euler number, adapted for low *Re*-number flows
F	[1/s]	flow rate (index specifies the unit of the flow rate)
Fo	[-]	Fourier number
g	[m/s²]	gravity
h	[J/kg]	mass specific enthalpy
$\Delta H_{r,i}$	[J/kg$_i$]	mass specific reaction enthalpy of species i
k_0	[1/s]	fluid volume specific pre-exponential factor

k	[1/s]	fluid volume specific reaction velocity constant
L	[m]	reactor bed length
m	[kg]	mass
n	[-]	reaction order
N	[-]	tube to particle diameter ratio
Nu	[-]	Nusselt number
p	[Pa]	pressure
Δp	[Pa]	pressure drop
Pe	[-]	Peclet number
Pr	[-]	Prandtl number
$\dot{Q}$	[W]	heat removed by external heat transfer
R	[J/mol K]	universal gas constant
r_A	[kg/m³ s]	fluid volume specific reaction rate of species A
Δr	[m]	thickness of surface layers of the computational mesh
Re	[-]	Reynolds number
S	[m²]	surface area
Sc	[-]	Schmidt number
Sh	[-]	Sherwood number
S_V	[1/m]	volume specific particle surface
t	[s]	time
T	[K]	temperature
ΔT_{ad}	[K]	adiabatic temperature rise at complete conversion
Δt	[s]	time step

u	[m/s]	velocity vector
U_s	[m/s]	superficial velocity
V	[m³]	volume
X_A	[-]	relative conversion of species A
Δx	[m]	edge length of a cell
Y_p	[-]	relative catalyst volume per fluid volume
z	[m]	axial distance from reactor inlet
Z	[-]	dimensionless reactor length

Greek letters

α	[W/m²K]	heat transfer coefficient
β	[-]	relative temperature rise (relative to adiabatic temperature rise)
γ	[-]	dimensionless activation energy, defined at inlet temperature
δ	[-]	boundary layer thickness
ε	[-]	bed porosity
λ	[W/m K]	heat conductivity
η_{Surf}	[-]	Surface accessibility factor
η_{eff}	[-]	effectivity factor
ρ	[kg/m³]	density
φ_p	[-]	particle volume fraction
μ	[Pa s]	dynamic viscosity of the fluid
ν_i	[-]	stoichiometric coefficient of species i
τ	[s]	hydraulic residence time of the fluid inside reactor

Subscripts and Superscripts

0		at inlet conditions
A		species A
ad		adiabatic
available		accessible by chemical species so that chemical reaction is enabled
bed		related to the particle bed
C		cross-section
c		cooling medium
Cat		catalyst
f		flow averaged quantity
l		liquid
m		mass transport
M		Quantities related to the geometry of the lab-scale reactor
p		particle
r		entire reactor
S		sphere
T		temperature field
T_0		defined at inlet temperature
V		volume-specific
´	$[1/kg_{Cat}]$	catalyst mass specific
´´	$[1/m^2_{Cat}]$	surface area specific
*		dimensionless quantity

1 Introduction

Fixed bed catalytic reactors are widely used in a variety of processes, such as production of bulk chemicals, petrochemical products, pharmaceutical products and fine chemicals. A commonly used reactor design is a tube packed with catalyst particles (referred to as packed bed reactor). The size of the used reaction vessels range from several cm³ to several m³. Tubular fixed bed reactors are especially applied in case the reaction taking place is strongly exothermic (e.g. oxidation reactors) or endothermic (e.g. steam reforming reactors), i.e., when cooling or heating of the reactor is important [1]. In that concern, tubes with a small diameter facilitate heat exchange. To keep the pressure drop in a moderate range often requires the use of relatively large catalyst particles, leading to a low tube-to-particle diameter ratio N. The range of $2 < N < 10$ is of particular interest [2], which is also often encountered in lab-scale fixed bed reactors.

In such reactors, the confining walls significantly affect the bed structure (called wall effect), and differences between global and local bed porosity arise [3], [4]. The result is a uneven fluid flow which gives rise to local differences in species transport towards the catalytically active particle surface, heat transport, and consequently, reaction rate [1]. Thus, commonly used simplified reactor models which assume plug flow and uniform bed porosity are not valid for packed bed reactors where wall effects dominate. This points out the necessity of reliable predictions describing the transport processes in this type of reactors.

Computational fluid dynamics (CFD) is often used to obtain detailed insight in the behavior of chemical reactors. This approach relies on partial differential equations describing the fluid flow (Navier Stokes equations), as well as equations for energy and mass transport, which are in discretized time and space. This work focuses on a Direct Numerical Simulation (DNS), meaning that a particle-resolved simulation is applied to capture all transport processes (hydrodynamics, heat transport and chemical species transport), as well as chemical reaction rates.

For DNS, a sufficiently high resolution in temporal and spatial range is essential. Regarding the hydrodynamics, characterized by $Re_p = \rho_l U_s d_p / \mu_l$ (with Re_p, ρ_l, U_s, d_p, μ_l the Reynolds number, fluid density, superficial velocity, particle diameter and fluid kinematic viscosity, respectively), the requirement on spatial resolution increases with increasing Re.

Considering species transport inside a liquid system, characterized by $Sc = \mu / (\rho_l D)$ (with Sc the Schmidt number and D the mass diffusivity), a full resolution of all transport processes gets even more challenging. This is a result of the generally small diffusion coefficients of chemical species in liquids. Length scales over which mass transport occurs are shorter by a factor $\sqrt{Sc}$ compared to hydrodynamic length scales. Unfortunately, in liquids $Sc = O(10^3)$, meaning that thin concentration boundary layers exist. Thus, a computational grid that is appropriate to resolve details of the flow is not necessarily fine enough to resolve the concentration field, making DNS of mass transport in high Sc-number systems a key challenge [5]. Especially in the proximity of the particle surface, steep concentration gradients are expected, which asks for high grid resolution in this so called concentration boundary layer.

Introducing heat transfer, the same considerations as for mass transport can be made due to analogy of these phenomena. However, heat is also transferred into the catalyst material (denoted internal heat transfer). Depending on the thermo-physical properties of solid and fluid, internal heat transfer could lead to a formation of temperature profiles inside the catalyst particle, which alters the reaction rate (compared to the case where internal heat transfer is ignored). Hence, exact results for reactive systems with a strong temperature effect (i.e., a high activation energy and reaction enthalpy) require the simulation of conjugate heat transfer. This is connected to additional computational effort as the heat transport equation also needs to be solved in the (often unreactive) solid phase.

In summary, the numerical size of the problem can be defined by the Peclet number of mass transfer ($Pe_m = Re_p \cdot Sc$), the geometric dimensions of the reactor, and whether the conjugate heat transfer has to be taken into account or not. For high Peclet numbers, DNS of the whole reactor domain might not be feasible due to limited computational resources and long computation time. In case the simulations are run for a bed section that is long enough to gain representative results, the obtained DNS results can be used to calibrate a simple reactor model. This strategy allows a prediction of overall reactor performance for packed bed reactors taking the above discussed local phenomena into account.

1.1 Goals

A Direct Numerical Simulation (DNS) is applied to obtain locally resolved information on the flow field, mass- and heat transfer processes, as well as on the reaction rate inside a tubular reactor. The tubular reactor is filled with a heterogeneous catalyst with a spherical shape. The fluid should represent a liquid with chemical species dissolved in it. As a model reaction, the acid-catalyzed exothermic esterification of dissolved salicylic acid and acetic anhydride is investigated, which results in the product acetylsalicylic acid and the byproduct acetic acid. Information on reaction kinetics and its temperature dependency are already available from preliminary batch reactor experiments (not part of this work). This model reaction is also carried out in a lab-scale tubular reactor, so the simulation results can be compared to experimental results.

The individual sub-goals of the present work can be summarized as follows:

- The catalyst bed is created applying the Discrete Elements Method (DEM). Therefore the open source software LIGGGHTS, version 3.0 is used. The organic catalyst used in the experiments shows the effect of swelling due to solvent diffusion into the solid phase. The changes of the geometric bed properties due to this effect should be investigated by introducing a numeric integration method (i.e., Monte Carlo Integration).

- Preliminary data collection and calculation of key parameters as well as deriving dimensionless numbers should clarify the reactor behavior on a qualitative level.

- A 3D simulation of an exothermic reaction inside an adiabatic reactor is run with the open source CFD software OpenFOAM, version 2.2.1. Key results from simulations are the pressure drop, the outlet temperature of the fluid, and the conversion. By varying Re and the bed porosity, different reactor operation conditions are investigated.

- Benchmark simulations (single sphere) are run for validation and verification of the used methods and solvers. Also, conjugated heat transfer (internal heat transport inside the catalyst material) and its effect on the chemical reaction rate is investigated.

- A reactor effectivity factor should be derived from the simulation results. Incorporating this factor in an adiabatic plug flow model should finally provide a reactor model that allows a more precise prediction of the overall performance of reactors with a low tube-to-particle diameter ratio.

1.2 Outline

In the first part of this work, approaches for modelling heterogeneous catalytic reactors and assessing non-ideal reactor behavior are briefly reviewed. Also, numerical simulation methods are presented. In Chapter 3, key parameters describing heterogeneous particle beds and non-isothermal reactive flow are provided and different reactor models are discussed in detail. In Chapter 4, preliminary calculations and considerations should clarify the challenges associated with the simulation setup. Chapter 5 treats the DEM-modelling of the particle bed and the calculation of key geometric parameters. Chapter 6 contains benchmark simulations focusing on stationary heat transfer, conjugate heat transfer, and reactive flow past a single sphere, as well as inside an empty duct with laminar flow. In chapter 7, the final simulations treating non-isothermal reactive flow inside a particle bed section are presented, together with DNS-calibrated predictions of the overall reactor performance. Conclusions and an outlook is provided in chapter 8.

2   State of the Art

A number of modeling concepts and correlations is available in the literature that is relevant for the current work. In the following, we give an overview over classical reaction engineering for packed bed reactors, then we introduce recent work in the field of detailed simulations and briefly summarize the most important correlations for the pressure drop in relevant packed beds.

2.1   Modelling of Ideal Reactors

In reaction engineering, the quickest way of designing heterogeneous catalytic fixed bed reactors is to model it like an ideal single phase tubular reactor, denoted as Plug Flow Reactor (PFR) model here. As the only difference, instead of determining the reactor volume at given throughput and target conversion, the required amount of catalyst is determined [6]. If temperature changes due to exothermic or endothermic reactions, the PFR model can be extended accordingly. After some simplifications (i.e., assuming isothermal or adiabatic reactor operation) a solution for such a PFR model can be obtained easily [7]. This PFR model assumes infinitely fast radial mixing (i.e., a uniform velocity, species concentration and temperature profile in the radial coordinate) and no axial mixing. This also implies that mass transport resistance towards the active surface is neglected. Clearly, the real behavior of packed bed reactors differs from this assumptions due to the complex bed geometry and the resulting complex hydrodynamics. This results in reaction rates that are governed by local effects (i.e., local temperature and species concentration) rather than attaining constant values over a reactor cross-section.

Therefore, intense research has been performed over many decades to understand how the particle bed structure, the hydrodynamic mixing effects and the transfer processes between solid and liquid phase influence the reactor performance.

2.2   Classical Approaches for Predicting Packed Bed Reactor Performance

Most investigations regarding hydrodynamic mixing effects and transport processes focus on infinitely wide beds of monodisperse spherical particles, since this is the most important design of packed bed reactors. Thus, for this prototype of packed beds, well established correlations are available [8,9].

The above mentioned hydrodynamic mixing effects are known as axial and radial dispersion effects. The axial dispersion can be understood as a back-mixing of the fluid with fluid regions upstream or downstream, caused by uneven fluid flow, which generally impairs reactor

performance [9]. Axial dispersion coefficients can be determined experimentally or by numerical simulation where the residence time distribution inside a reactor is evaluated (e.g. analysis of the response to a signal produced by a pulsed species injection into the system [8,10].

The radial dispersion describes the gradients of species concentration or temperature due to deflected fluid flow (caused by the bed geometry) in radial direction. Correlations for radial and axial diffusion coefficients are provided e.g. by Edwards [10] and Gunn [8].

Axial and radial dispersion coefficients can now be fed into a model which allows the prediction of local temperature and species concentration [9]. However, this *homogeneous* model is based on simplifications (homogeneous porosity, angular independency), so uncertainties for low N reactors still remain.

As another deviation from real-life reactor performance, the PFR model ignores the existence of mass transport resistance. Mass transport resistance arises if the chemical reaction rate is relatively fast compared to diffusive mass transport, i.e., in case the chemical species is consumed faster than it is transported towards the active surface. A widely used correlation to predict the overall heat or mass transfer (analog description of temperature and concentration field possible) is provided by Gunn [8]. This correlation was also established for infinitely wide beds of monodisperse spheres with moderately high particle volume fractions. In this work, the particle bed undergoes considerable compaction due to particle swelling, so the Gunn correlation is not expected to hold for these conditions.

Addressing the prediction of heat transfer inside a particle bed, most correlations just treat the so called *external problem* where the temperature distribution inside the particle is nearly isothermal and the heat transfer is limited by the liquid phase around the particle [11], [12]. This assumption is only legitimate in case the conductivity of the solid is much higher than the fluid conductivity, which is generally not the case when treating liquid systems with organic catalyst materials [13]. Thus, correlations which address conjugate heat transfer (i.e., a coupled solution of the *internal* and *external* heat transfer problem) are challenging.

Besides predicting the conversion and heat evolution inside chemical reactors, an accurate prediction of the pressure drop is also of basic importance, since the pressure drop defines the energy requirements of the pumping system. A widely used correlation to predict the pressure drop of an infinitely wide bed over a wide range of Re_p is the Ergun equation [14]. However, it was found that for reactors with a small tube-to-particle ratio ($N \leq 15$, [15]) the confining walls affect the pressure drop significantly.

It was even found that the qualitative change (i.e., increase or decrease) in the pressure drop depends on the flow regime, indicating the key importance of walls. Specifically, for laminar flow additional tube wall friction generally increases the pressure drop. In the turbulent regime (i.e., Re_p >100) the effect of a locally higher void fraction close to the (impenetrable) walls can reduce the pressure drop. Thus, for packed bed reactors with low N it is essential to incorporate the changes in hydraulic radius and local porosity due to confining walls [15].

2.3 Numerical Simulation of Packed Bed Reactors

In the early stage of CFD simulations, a conventional approach was to simplify the bed geometry, such as assuming a uniform bed porosity. Under these assumptions it is possible to treat the particle bed as a pseudo-continuum, i.e., there is no need to resolve the shape of each particle in the bed, reducing the numerical size of the problem and making the solution rather quick and cheap [1]. However, this method cannot capture flow and transport effects on a particle scale and hence is not suitable for predictions of low N reactors. With the fast growing computational power, particle resolved simulations of laminar and turbulent flow involving heat transfer for several hundreds of spheres became state of the art.

Dixon et al. [2] performed simulations involving a packed bed with 700 spheres and $N = 7$, with Re_p up to 5600. The turbulent flow was not directly simulated, however, modelled using a realizable k-ε model and enhanced wall treatment. A contact point modification was performed to reduce the cell number and to improve mesh quality. Addressing non-isothermal reactive flow, Dixon et al. [1] presented a *solid particle model* which allowed the simulation of reactions inside porous catalyst particles. Simulations were performed for a single sphere as well as for a small 120° bed section for $N = 4$ (containing around 8 spheres) for Re_p up to $2 \cdot 10^4$. In this work it was pointed out that the commonly used *porous media* concept for enabling species transport inside solid particles gives rise to a flow artefact inside the particle which means that the no slip boundary condition on the particle surface is violated and in this respect the *porous media* concept gives wrong results.

In the last decade, intense research has been performed on DNS of heat-and mass transfer in fluid-particle flows (e.g., fluidized beds). Specifically, alternative numeric methods to the (in this work used) finite volume method have been proposed (e.g., the Lattice Boltzmann Method, or the Immersed Boundary Method) [12]. Typically, the case of stationary arrays of spheres is frequently used for solver validation, so these previous studies also provide results for fixed bed applications [11].

For example, Derksen [5] investigated a bed of 165 spheres with Re_p ranging up to 10 and Sc up to 1000. Throughout his analysis the particle volume fraction was always kept in a low range, and contact points were generally avoided.

To the best of our knowledge, and at the time of writing this thesis, no work has considered DNS of non-isothermal reactive flow inside a packed bed with low N, high Schmidt numbers, and extremely dense packings.

2.4 Pressure Drop Correlations

Due to its practical importance there exist several correlations for predicting the pressure drop of packed bed reactors over a wide range of Re_p and ε. For infinitely wide beds packed with spherical particles, a commonly used correlation is the Ergun equation [14]:

$$\frac{\Delta p}{L} = 150 \frac{\mu U_S}{d_p{}^2} \frac{(1-\varepsilon)^2}{\varepsilon^3} + 1.75 \frac{\rho_l U_S{}^2}{d_p} \frac{(1-\varepsilon)}{\varepsilon^3} \tag{2.1}$$

With L, μ, U_S, ε, d_p, ρ_l being the length of the packed bed, dynamic viscosity of the fluid, superficial velocity, overall bed porosity, characteristic particle diameter and fluid density, respectively.

However, it was found that for reactors with a small tube-to-particle ratio ($N \leq 15$, [15]) the confining walls can affect the pressure drop significantly and even in a counteracting way, depending on the flow regime. In case of laminar flow, friction forces on the tube wall generally increase the pressure drop. In the turbulent regime ($Re_p > 100$) the effect of a locally higher void fraction, caused by the fact that the catalyst particles do not penetrate the tube walls, can reduce the pressure drop. In the correlation provided by Eisfeld [15], the wall effects are incorporated by introducing two additional parameters :

$$\frac{\Delta p}{L} = \frac{\rho_l U_S{}^2}{d_p} \left(154 \frac{A_w{}^2}{Re_p} \frac{(1-\varepsilon)^2}{\varepsilon^3} + \frac{A_w}{B_w} \frac{(1-\varepsilon)}{\varepsilon^3} \right) \tag{2.2}$$

with the coefficient A_w accounting for the contribution of the confining walls to the hydraulic radius and B_w describing the local porosity effects due to the confining walls:

$$A_w = 1 + \frac{2}{3} \frac{d_p}{d} (1-\varepsilon) \tag{2.3}$$

$$B_w = \left[1.15 \left(\frac{d_p}{d} \right)^2 + 0.87 \right]^2 \qquad (2.4)$$

with d being the tube diameter.

One can see that the first term in Eqn. (2.1) (representing viscous forces) scales linearly with flow velocity U_S, whereas the second term (representing the inertia forces) scales to the square of U_S. Thus, for very low flow velocity, viscous forces are dominating. This fact finds application when a dimensionless formulation of the pressure drop (Euler number) is used. So it should be noted that, since in this work we deal with low Reynolds number flows, pressure drop is scaled linearly with U_S as indicated in Table 3-2:

$$Eu = \frac{\Delta p \, d_p}{U_S \mu} \qquad (2.5)$$

3 Theory and Model

The physical processes of the exothermic surface reaction inside a packed bed reactor considered in this work can be summarized as follows:

The chemical species A and B are dissolved in a liquid flowing through a tube loaded with particles which have a catalytically active surface. It is assumed that the chemical reaction only takes place if species A and B undergo surface contact (that means that non-catalyzed reactions and reactions inside the solid particles are neglected). The relevant fields affecting the overall reactor performance are the flow field, temperature and species concentration. Pressure drop is also relevant for estimating the necessary power supply of the pumping system.

The dominating transport mechanisms of species inside the fluid are convective and diffusive mass transport. Due to wall friction, the flow velocity in proximity to the particle surface (and also in proximity to the tube wall) is very low, approaching 0 on the surface (i.e., the no slip condition). Thus, in the proximity of the particle surface, the mass transport is dominated by diffusion, which in flowing liquids is orders of magnitudes slower than convective mass transport (or in other words, diffusive length scales are orders of magnitude shorter than convective ones). This results in narrow zones in the proximity of walls with steep gradients of the species concentration, which are denoted here as concentration boundary layers. The same occurs for the temperature field due to the analogy of these two scalar fields, however, with different time and length scales. Thus, the reaction rate on the catalyst surface (a function of actual species concentration and temperature on the active surface, with both fields dominated by the flow field) can deviate significantly from the reaction rate estimated with the bulk concentration.

In the subsequent part of this chapter, after presenting general theory on reaction kinetics, key parameters describing a heterogeneous particle bed and non-isothermal reactive flow (sections 3.1-3.3) are presented. These governing equations are the basis for the computation of local phenomena using of CFD software (sections 3.4-3.5). In section 3.6, standard methods for modelling a heterogeneous fixed bed reactor (assuming ideal reactor behavior) are presented. Finally, a method is presented to calibrate such a simple reactor model with DNS data to take into account local phenomena and non-ideal reactor behavior.

3.1 Chemical Reaction and Reaction kinetics

In this work we want to look at a heterogeneous chemical reaction where two reactants A and B, dissolved in one liquid solvent, are reacting in the presence of a solid catalyst to form a product P and a byproduct R:

$$A + B \xrightarrow{\quad Catalyst \quad} P + R \tag{3.1}$$

For an equimolar reaction the stoichiometric coefficients are $v_A, v_B = -1$ and $v_P, v_R = +1$.

A general expression for the reaction rate is given by:

$$-r = k c_A{}^n c_B{}^m \tag{3.2}$$

k is a temperature dependent reaction velocity constant and c_A, c_B are the mass concentrations of species A and B, respectively.

In equation (3.2), the sum of the exponents states the overall reaction order. Defining species A as the reaction rate limiting component and assuming a first order reaction of species A, the reaction rate of species A can be written as:

$$-r_A = k c_A \tag{3.3}$$

The reaction velocity constant, it´s temperature dependency and the reaction order usually have to be determined experimentally for the specific reactive system. As a heterogeneous catalyst is involved, knowing the physical properties of the catalyst as well as knowing where the reaction takes place is essential for a correct description of the reaction kinetics.

In this work we consider a non-porous, spherical catalyst, so the reaction can only take place on the catalyst surface. Generally, the reaction rate (and analogously the reaction velocity constant) can be related to the volume, mass or surface of catalyst involved in the reaction:

$$r_A V_{Cat} = r_A' m_{Cat} = r_A'' S_{Cat} \tag{3.4}$$

with r_A, $r_A{'}$, $r_A{''}$, being the catalyst volume, mass, and surface area specific reaction rate, respectively. The physical catalyst properties which relate catalyst volume, mass and surface are summarized in section 3.2.

Given the overall bed porosity of the reactor, one can also relate the reaction velocity constant to the fluid volume, which allows to treat the heterogeneous catalytic reactor like a single phase reactor.

Starting from a mass specific reaction velocity constant (which is readily available in experimental procedures because the mass of catalyst can be easily measured gravimetrically), the transformation to the remaining quantities can be done as follows:

Surface-area specific reaction velocity constant

$$k'' \left[\frac{m^3}{m^2_{Cat} \, s} \right] = k' \frac{\rho_s}{S_V} \tag{3.5}$$

with k', ρ_S and S_V being the mass specific reaction velocity constant, density of the solid catalyst, and the specific particle surface, respectively.

Fluid volume specific reaction velocity constant

$$k \left[\frac{1}{s} \right] = k'' \frac{V_{Cat}}{V_l} S_V = k'' Y_P S_V \tag{3.6}$$

with Y_P being the relative catalyst volume per fluid volume. In the subsequent part of the work, catalyst volume specific quantities are not used at all, so k as well as r_A are related to the fluid volume and not to the catalyst volume.

Fluid-volume specific reaction rate

$$r_A \left[\frac{kg_A}{m^3_l \, s} \right] = k c_A = k_0 e^{-\frac{Ea}{RT}} c_A \tag{3.7}$$

Surface-area specific reaction rate

Following the relation given in (3.4), the surface-area specific reaction rate is defined as

$$r_A'' \left[\frac{kg_A}{m^2_{Cat} \, s} \right] = k'' c_A = k_0'' e^{-\frac{Ea}{RT}} c_A \tag{3.8}$$

Note that equation (3.6) is only valid if the entire particle surface is available for reaction. If, however, the accessible particle surface diminishes because of particle swelling, this loss of surface can be taken into account by defining a surface accessibility factor (see equation (3.15) below).

Thus, the final relation between the fluid-volume specific and the surface-area specific reaction velocity constant inside a bed of swelling particles can be written as

$$k = k'' Y_P S_V \eta_{Surf} \tag{3.9}$$

3.2 Key Parameters Describing a Dense Bed of Particles

Characteristic particle diameter

Given an ensemble of differently sized and shaped particles, it is convenient to define a characteristic particle diameter which can be used to calculate adequate properties for the whole particle regime. In this work, the Sauter mean diameter d_{32} is used as characteristic diameter for particle regime with a known particle size distribution. d_{32} is defined as the diameter of equally sized spheres which have the same specific particle surface area as the whole particle ensemble:

$$d_{32} = \frac{6}{S_V} = 6\frac{V_{bed}}{A_{bed}} \tag{3.10}$$

Specific surface area (surface per catalyst volume, for spherical particles):

$$S_V = \frac{A_{Cat}}{V_{Cat}} = \frac{6}{d_p} \tag{3.11}$$

Particle volume fraction

$$\varphi_p = \frac{V_{Cat}}{V_r} = \frac{V_{Cat}}{V_l + V_{Cat}} \tag{3.12}$$

with V_l being the fluid volume.

Bed porosity (as an alternative expression for the particle volume fraction)

$$\varepsilon = \frac{V_l}{V_r} = 1 - \varphi_p \tag{3.13}$$

Relative catalyst volume per fluid volume

$$Y_p = \frac{\varphi_p}{1 - \varphi_p} \tag{3.14}$$

Surface accessibility factor

To take into account the swelling behavior of the catalyst used in this work, an accessibility factor can be introduced which takes into account the loss of surface area due to swelling (the method to compute the accessibility factor is explained in more detail in section 6):

$$\eta_{Surf} = \frac{S_{available}}{V_{Cat}S_V} \tag{3.15}$$

3.3 Key Parameters describing Non-Isothermal Reactive Flow

An approach to express the relevant process parameters of a system is to derive dimensionless transport equations. After collecting all relevant (dimensional) process parameters describing a non-isothermal reactive flow in a fixed bed reactor, the dimensionless numbers were derived from a dimension analysis (see Table 3-1, details are provided in Appendix A, Section 10.1).

Table 3-1: Physical significance of dimensionless quantities for reactive non-isothermal flow in a fixed bed catalytic reactor; input quantities

Quantity	*Equation*	*Significance*
N	$\dfrac{d_r}{d_p}$	tube to particle diameter ratio
Z	$\dfrac{z}{d_p}$	dimensionless length
Y_p	$\dfrac{\varphi_p}{1-\varphi_p}$	relative particle volume per fluid volume
Re_p	$\dfrac{d_p U_S \rho_l}{\mu}$	importance of inertial forces over viscous forces
E_{therm}	$\dfrac{T_0\, c_{p,l}}{U_s^2}$	importance of thermal energy over kinetic energy (not important for low-Reynolds number flow, since $>>1$)
Pr	$\dfrac{c_{p,l}\mu}{\lambda_l}$	fluid property: momentum transport vs. conductive heat transport
Sc	$\dfrac{\mu}{D_A \rho_l}$	momentum transport vs. species transport
γ	$\dfrac{E_a}{RT_0}$	dimensionless activation energy, related to the inlet temperature
Da_I	$\dfrac{r_{A,0}d_p}{c_{A,0}U_S}$	reaction rate over convective transport rate
$C_{A,0}$	$\dfrac{c_{A,0}}{\rho_l}$	mass fraction of species A at the inlet
β	$\dfrac{\Delta H_{r,A}c_{A,0}}{\rho_l c_{p,l}T_0}$	relative adiabatic temperature rise at complete conversion

Applying a dimension analysis also for the output parameters of interest (which are basically the pressure drop, temperature change and species conversion in the reactor), one can derive the following dimensionless output quantities:

Table 3-2: Dimensionless output quantities for reactive non-isothermal flow

Quantity	Equation	Significance
Eu	$\dfrac{\Delta p\, d_p}{U_S \mu}$	dimensionless pressure drop for low-Reynolds number flows
X_A	$1 - \dfrac{c_A}{c_{A,0}}$	conversion of species A

Regarding the Damköhler number shown in Table 3-1, $r_{A,0}$ is defined as the reaction rate at the inlet temperature and the inlet concentration of species A. Hence, a reference reaction velocity constant can be defined based on Arrhenius law:

$$k_{T0} = k_0 e^{\frac{-Ea}{RT_0}} = k_0 e^{-\gamma} \tag{3.16}$$

Note that for a first order reaction Da_I becomes independent of the inlet species concentration:

$$Da_I = \frac{c_{A,0} k_0 e^{\frac{-Ea}{RT_0}} d_p}{c_{A,0} U_S} = \frac{k_{T0} d_p}{U_S} \tag{3.17}$$

Describing a reaction inside a particle bed, a more adequate definition for *Da* can be found, based on the surface-area specific reaction rate and the interstitial velocity (i.e. taking into account the average velocity inside the particle bed which is done by incorporating the overall bed porosity ε):

$$Da = k_{T0}^{''} \frac{\varphi_p}{1-\varphi_p} \frac{\varepsilon d_p}{U_S} \frac{6}{d_p} \eta_{Surf} = \frac{k_{T0}^{''} \varphi_p 6 \eta_{Surf}}{U_S} \tag{3.18}$$

3.4   Governing Equations describing Non-Isothermal Reactive Flow

The transport processes inside a fixed bed reactor can be described with the conservation equations for mass, momentum and energy. For expressing the transport equations, the following assumptions are made [11]:

- Fluid is treated as an incompressible medium (no divergence in flow field)
- Constant thermo-physical properties (density, viscosity, heat capacity, thermal diffusivity, diffusion coefficient)
- Viscous heating effect is neglected
- Effects due to buoyancy are neglected

Continuity equation

$$\nabla \cdot \mathbf{u} = 0 \tag{3.19}$$

with ∇ being the Nabla operator and $\mathbf{u}$ being the velocity vector

Momentum equation

$$\rho_l \frac{\partial \mathbf{u}}{\partial t} + \rho_l \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u} \tag{3.20}$$

Mass transport equation

$$\rho_l \frac{\partial c_A}{\partial t} + \rho_l (\mathbf{u} \cdot \nabla) c_A = \rho_l D_A \nabla^2 c_A + v_A r_A \tag{3.21}$$

with c_A being the mass concentration of species A, t the time, D_A the diffusion coefficient of A in the fluid, v_A the stoichiometric coefficient of A and r_A the reaction rate of species A.

Energy equation

$$\rho_l c_{p,l} \frac{\partial T}{\partial t} + \rho_l c_{p,l} (\mathbf{u} \cdot \nabla) T = \lambda \nabla^2 T + \Delta H_{r,A} r_A \tag{3.22}$$

with $c_{p,l}$ being the heat capacity, T the temperature, λ the thermal conductivity of the fluid and $\Delta H_{r,A}$ the mass specific reaction enthalpy of species A.

Chemical reaction

Describing a first order reaction, the fluid-volume specific reaction rate (see Eqn. (3.7)) is used.

3.5 Dimensionless Transport Equations

Introducing specific reference values for the above variables, one can derive dimensionless transport equations. These equations can be readily applied for upscaling problems (e.g. when experimental results from a lab-scale reactor are used to predict de behavior of an industrial reactor), since only dimensionless quantities occur in these equations. Here, the dimensionless values are set in a way to allow a meaningful description of fluid flow through a tube filled with particles. Specifically, the dimensionless values are defined as:

$$
t^* \equiv \frac{t U_S}{d_p} \qquad Z \equiv \frac{z}{d_p} \qquad p^* \equiv \frac{p d_p}{U_S \mu} \qquad u^* \equiv \frac{u}{U_S} \qquad C_i \equiv \frac{c_i}{c_{A,0}} \tag{3.23}
$$

Introducing the fluid inlet temperature T_0 as reference temperature and expanding the exponential term of equation (3.40) leads to the expression

$$
e^{\frac{-Ea}{R \cdot T}} = e^{\frac{-Ea}{R \cdot T_0}} \, e^{\left\{ \frac{-Ea}{R}\left(\frac{1}{T} - \frac{1}{T_0} \right) \right\}} \tag{3.24}
$$

By doing so, the reference reaction velocity constant can be rewritten in compact form as already indicated in Eqn. (3.16).

By use of the expressions (3.23) and, the dimensionless transport equations can be formulated (shown here for the energy equation (3.22)):

$$
\frac{\partial T}{\partial t^*} + \mathbf{u}^* \cdot \nabla^* T = \frac{1}{Pe} \nabla^{*2} T + \Delta T_{ad} \, Da_I \, C_A \, e^{\left\{ \frac{-Ea}{R}\left(\frac{1}{T} - \frac{1}{T_0} \right) \right\}} \tag{3.25}
$$

with $Pe = Re \cdot Pr$ being the Peclet number for heat transfer.

Applying the same procedure, the remaining transport equations can be written as:

$$
\frac{\partial \mathbf{u}^*}{\partial t^*} + \mathbf{u}^* \cdot \nabla^* \mathbf{u}^* = -\frac{1}{Re} \nabla^* p^* + \frac{1}{Re} \nabla^{*2} \mathbf{u}^* \tag{3.26}
$$

$$
\frac{\partial C_A}{\partial t^*} + \mathbf{u}^* \cdot \nabla^* C_A = \frac{1}{Pe_m} \nabla^{*2} C_A + v_A \, Da_I \, C_A \, e^{\left\{ \frac{-Ea}{R}\left(\frac{1}{T} - \frac{1}{T_0} \right) \right\}} \tag{3.27}
$$

with $Pe_m = Re \cdot Sc$ being the mass transfer Peclet number.

Solving the partial differential equations (3.25) to (3.27) in time and space requires the use of dedicated CFD software. The CFD software used in this work is OpenFOAM, version 2.2.1, which is based on the finite volume method. This method discretizes the system´s geometry into volume elements (cells) and solves the (discretized) transport equations in each cell for predefined number of time steps. Hence, the size of the problem is defined by the extent of the temporal and spatial range, which is:

- the shortest and longest characteristic time scale of transport processes in the system
- the smallest and largest characteristic length scale of the system

The dimensionless numbers shown in the equations (3.25) to (3.27) assess the temporal and spatial extents of each particular field, and thus allow further considerations regarding discretization. As in this work the attempt is made to realize a DNS (which means no modelling of field gradients close to surfaces, as well as no time averaging implicit to the method), a sufficiently fine discretization of time and space is essential. Concerning spatial discretization, the so called meshing process has to take care of especially high grid resolution in regions where certain boundary conditions have to be fulfilled or where high gradients of the fields (velocity, pressure, species concentration and temperature) are expected. A more detailed discussion on spatial as well as temporal discretization is provided in Chapter 4.

Inevitably when dealing with several hundreds of fully resolved particles and high Peclet numbers, the meshing process becomes a challenge, leading to a high cell count. Thus, in order to reduce the numerical size, the attempt is made to only investigate a short bed section to obtain representative numerical results. Afterwards we will use this results for calibrating a simple reactor model which allows precise predictions for the entire particle bed.

Therefore, the used reactor models are presented in the next section.

3.6 Classical Modelling of Fixed Bed Reactors

3.6.1 Modelling of a Stationary Isothermal Plug Flow Reactor (PFR)

The simplest way to model a fixed bed catalytic reactor can be realized analog to the modelling of an ideal single phase isothermal plug flow reactor. This means the mass transfer processes between solid particle phase and fluid phase are not incorporated at all. The assumptions are stationary and isothermal operation conditions, uniform fluid velocity and uniform composition in radial direction. Furthermore, no back mixing of the fluid in axial direction occurs, providing a plug flow regime in the duct. Hence, the composition of a mixture can be described only as a function of the axial distance from the entrance (see Figure 3-1):
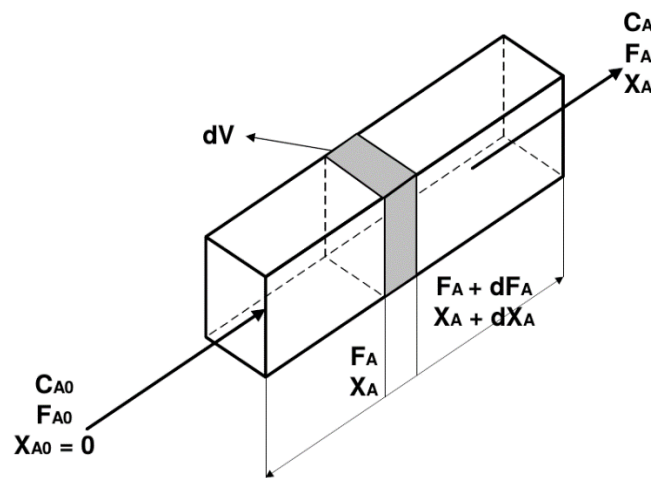
Figure 3-1: Model of an ideal plug flow reactor (PFR) **[6]**.

For calculating the concentration of species A, the material balance has to be applied to a differential section of the tubular reactor, giving the material balance equation:

$$F_A = F_A + dF_A - r_A dV \qquad (3.28)$$

with r_A being the fluid volume specific reaction rate, dV being a infinitesimal volume element (built-up by cross-sectional area A of reactor and length element dz) and F_A being the mass flow rate of species A.

For reactions where volume change due to reaction can be neglected (valid for fluid phase reactions), F_A can be written as

$$F_A = F_{V,0} c_A \qquad (3.29)$$

$F_{V,0}$ is the constant volumetric flow rate passing through the reactor.

The amount of converted species A can be expressed by the conversion X_A:

$$X_A = \frac{F_{A,0} - F_A}{F_{A,0}} = 1 - \frac{c_A}{c_{A,0}}$$

(3.30)

with equation (3.30), the mass flow of species A exiting the volume element can be written as:

$$F_A = F_{A,0}(1 - X_A)$$

(3.31)

Differentiation of equation (3.31) leads to

$$\frac{dF_A}{dX_A} = -F_{A,0}$$

(3.32)

So differential equation (3.28) can be rearranged to

$$F_{A,0}dX_A = -r_A dV$$

(3.33)

By substituting equation (3.29) for $F_{A,0}$ into (3.33), splitting of variables and integration leads to the general relation between reactor volume and conversion for a single phase PFR:

$$\int \frac{dV}{F_{V,0}c_{A,0}} = \int \frac{dX_A}{-r_A}$$

(3.34)

Keeping in mind that in this work we consider a heterogeneous reactor, and the reaction rate r_A is expressed fluid volume specific, a further adaption of equation (3.34) is made.

Therefore, the hydraulic residence time of a fluid passing through the tubular reactor (which is partially filled with solid particles, see Figure 3-2), is introduced:

$$\tau = \frac{V_l}{F_{V,0}} = \frac{zA\varepsilon}{U_S A} = \frac{z\varepsilon}{U_S}$$

(3.35)

U_S is the superficial velocity (fluid velocity for empty duct), V_l the fluid volume, ε the overall bed porosity, A the cross-sectional area of the reactor tube and z the bed length of the reactor.
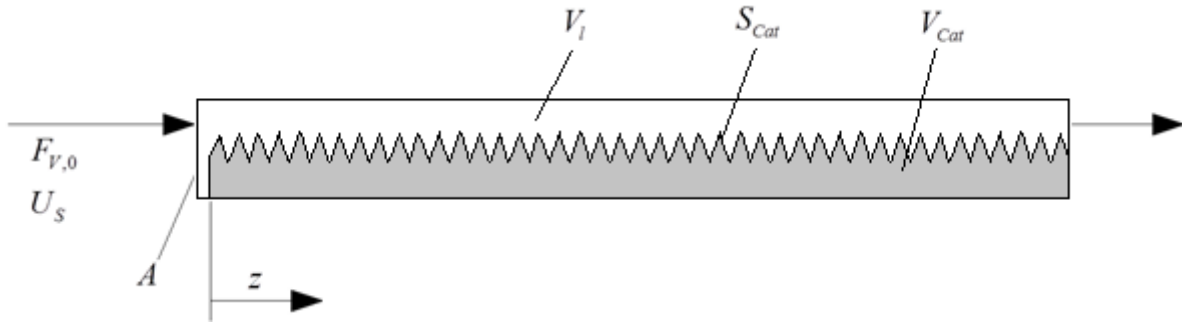
Figure 3-2: Sketch of a heterogeneous catalytic reactor model.

Substituting c_A by $c_A = c_{A,0}(1 - X_A)$ (analog to expression (3.31)) and inserting the reaction rate for a first order reaction yields

$$\frac{\tau}{c_{A,0}} = \frac{1}{k c_{A,0}} \int \frac{dX_A}{(1 - X_A)} \qquad (3.36)$$

Solving the integral gives

$$k\tau = \ln\left(\frac{1}{1 - X_A}\right) \qquad (3.37)$$

Finally, the conversion inside a heterogeneous reactor, modelled as a PFR, is given by:

$$X_{A,PFR} = 1 - e^{-k\tau} = 1 - e^{\left(\frac{-kA\varepsilon z}{F_{V,0}}\right)} \qquad (3.38)$$

Equation (3.38) can also be expressed in dimensionless form by using equation (3.9) and (3.18) :

$$X_{A,PFR} = 1 - e^{-Da \cdot Z} \qquad (3.39)$$

where $Z = z / d_p$ is a dimensionless bed length of the particle bed.

3.6.2   Thermal Modelling of Plug Flow Reactors [7]

Dealing with exothermic reactions, heat is released due to the chemical reaction. In this work, the released heat is related to the conversion of species A. This influences the reaction kinetics as the reaction velocity constant is a function of temperature, which can be expressed by Arrhenius´ law:

$$k = k_0 e^{\frac{-Ea}{RT}} \tag{3.40}$$

with k_0 being the pre-exponential factor, Ea the activation energy, R the Universal gas constant and T the temperature.

This implies that, if no adequate measures of external cooling are taken, isothermal reaction conditions (as assumed in section 3.6.1) can no longer be met and the fluid temperature rises in downstream direction of the reactor. Hence, it is important to take into account the temperature evolution along the chemical reactor. The steady state energy balance over a differential length element states that (if no work is introduced into the system) the enthalpy change of the fluid equals the heat removed by external heat transfer:

$$F_m dh = d\dot{Q} \tag{3.41}$$

with $F_m = F_{V,0} \cdot \rho_l$ being the stationary mass flow (in the case of constant fluid density) and $d\dot{Q}$ being the heat removed by external heat transfer.

If the pressure changes relatively little over the reactor, the enthalpy change of the fluid is characterized by the fluid´s temperature change and by the heat release due to conversion of species A, which can be expressed as:

$$dh = c_{p,l} dT + \frac{\Delta H_{r,A} c_{A,0}}{\rho_l} dX_A \tag{3.42}$$

The reactors capacity to remove the heat released by the chemical reaction can be calculated by

$$d\dot{Q} = -\alpha_c (T - T_c) d_r \pi dz \tag{3.43}$$

with α_c being the overall heat transfer coefficient between the outer tube wall and the cooling medium, T_c being the temperature of the cooling medium and d_r being the tube diameter.

This leads to the expression:

$$F_{V,0}\rho_l c_{p,l}dT + \Delta H_{r,A}F_{V,0}c_{A,0}dX_A = -\alpha_c(T-T_c)d_r\pi dz \qquad (3.44)$$

So the differential temperature change due to reaction and external heat transfer can be expressed as (division by $F_{V,0}\cdot\rho_l\cdot c_{p,l}$):

$$dT = \frac{-\Delta H_{r,A}c_{A,0}}{\rho_l c_{p,l}}dX_A - \frac{\alpha_c d_r\pi}{F_{v,0}\rho_l c_{p,l}}(T_c-T)dz \qquad (3.45)$$

with the material balance equation for dX_A given by equation (3.34).

To compute the evolution of temperature and conversion for a PFR along the reactor, the differential equation (3.34) needs to be solved in conjunction with (3.45), which is only straightforward for two ideal reactor models:

- Adiabatic reactor operation
- Isothermal reactor operation

Adiabatic reactor operation

Under adiabatic conditions, the last term in equation (3.45) vanishes, so a direct relation between temperature and the conversion is obtained:

$$dT = \Delta T_{ad}dX_A \qquad (3.46)$$

ΔT_{ad} is called adiabatic temperature rise and is defined as:

$$\Delta T_{ad} = \frac{-\Delta H_{r,A}c_{A,0}}{\rho_l c_{p,l}} \qquad (3.47)$$

Equation (3.47) allows to estimate the theoretical outlet temperature of the fluid at complete conversion and adiabatic reactor operation by the simple relation:

$$T_{ad} = T_0 + \Delta T_{ad} \qquad (3.48)$$

Integrating equation (3.46) with initial conditions $T=T_0$ for $X_A=0$ leads to the relation

$$X_A = \frac{T-T_0}{\Delta T_{ad}} \qquad (3.49)$$

Expression (3.46) can now be substituted in equation (3.34), and after inserting the reaction rate expression for a first order reaction, a single differential equation is left [7]:

$$\frac{dT}{d\tau} = (T_{ad} - T)k_0 e^{\frac{-Ea}{RT}}$$

(3.50)

After substituting equation (3.35) for τ, an ordinary differential equation is obtained, which can be integrated numerically to compute the temperature profile along an adiabatic PFR:

$$\frac{dT}{dz} = (T_{ad} - T)\frac{k_0 \varepsilon A}{F_{V0}} e^{\frac{-Ea}{RT}}$$

(3.51)

By inserting equation (3.24) and applying the same procedure as for the PFR model, the final equation describing the adiabatic PFR model (denoted as APFR model) is:

$$\frac{dT}{dZ} = (T_{ad} - T) Da \; e^{-\frac{Ea}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

(3.52)

Again, here we have used the dimensionless axial position $Z = z/d_p$.

3.6.3 Calibrating the Reactor Model with DNS Data

In order to incorporate experimental or numerical findings concerning the species conversion X_A (which can be seen as the measurable result of overall reactor performance), one can define a reactor effectivity factor:

$$\eta_{eff} \equiv \frac{X_{A,Sim}}{X_{A,APFR}}$$

(3.53)

In this work, the effectivity factor relates the conversion obtained in simulations with the conversion predicted by the APFR model, so η_{eff} accounts for the non-ideal reactor behavior like back-mixing, species transport resistance and local deviations from bulk concentration and temperature.

η_{eff} can be introduced as a calibration parameter into equation (3.52), so the adapted model equation to predict conversion and temperature rise inside a non-ideal, adiabatic PFR can be expressed as (denoted as CAPFR model):

$$\frac{dT}{dZ} = (T_{ad} - T) Da \; \eta_{eff} \; e^{-\frac{Ea}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

(3.54)

4 Theoretical Analysis

4.1 Collection of Process relevant Data

The chemical reaction taking place inside the fixed bed reactor can be written as:

$$Salicylic\ acid + Acetic\ anhydride \xrightarrow{\ catalyst\ } Acetylsalicylic\ acid + Acetic\ acid \quad (4.1)$$
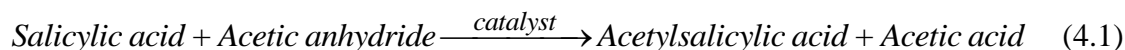
The involved chemicals (for their chemical structure see Table 4-1) are dissolved in the solvent ethyl acetate. The catalyst particles are made out of an organic resin, consisting of a styrene-divinylbenzene matrix with sulfonic acid groups acting as active centers [16]. In Table 4-2, the thermo-physical properties of fluid and solid are presented. Table 4-3 shows the reactor geometry and the operation conditions applied in the experiments. Experimental work was performed by co-workers within our group. The most important experimental results are summarized in the Appendix C (see Chapter 12), but are not discussed here in detail.

Table 4-1: Chemical structure of involved substances
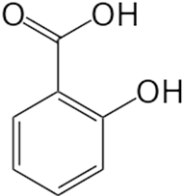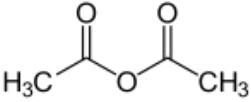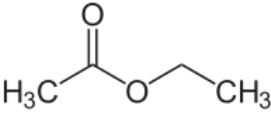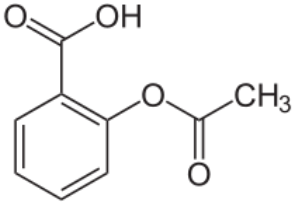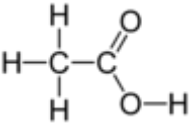
Educts		*Solvent*
Salicylic acid (A)	Acetic anhydride (AA)	Ethyl acetate (EtAc)

Products	
Acetylsalicylic acid (ASS)	Acetic acid (AAc)

Table 4-2: Thermo-physical properties of fluid and solid phase.

Property	Symbol	Unit	Liquid	Solid
Temperature	T	K	348	348
Density (of mixture)	ρ	kg/m³	908	1400 [1]
Thermal conductivity	λ	W/m K	0.125	0.17
Specific heat capacity	c_p	J/kg K	2100	2567 [2]
Thermal diffusivity	a	m²/s	$6.55 \cdot 10^{-8}$	$4.73 \cdot 10^{-8}$
Kinematic viscosity	μ	kg/m s	$2.55 \cdot 10^{-4}$	-
Diffusion coefficient	D_A	m²/s	$6.8 \cdot 10^{-10}$ [3]	-

(1)…measured with a helium pyknometer

(2)…measured with a Differential Scanning Calorimeter (DSC)

(3)…Diffusion coefficient of A in EtAc; (Molecular Dynamics simulation, performed by Feenstra [17])

Table 4-3: Geometric parameters and operation conditions of lab-scale reactor

Geometric parameters	Symbol	Unit	Value
particle diameter	d_p	mm	0.6 [1]
reactor diameter	d_r	mm	8
reactor length	l	mm	40
reactor volume	V_r	ml	2
cross-sectional area	A	mm²	50.3
initial bed porosity	ε	-	0.5
catalyst loading	m_{Cat}	g	1.41
number of catalyst particles	n_p	-	$7.6 \cdot 10^3$
Operation conditions			
inlet temperature	T_0	K	348
max. swelling of particle diameter	-	%	15 [1]
bed porosity 15% swollen	ε	-	0.27 [2]
volumetric flow rate	$F_{V,0}$	ml/min	1
mass fraction species A inlet	$c_{A,0}$	-	0.152 [3]
Reaction parameters			
activation energy	E_a/R	K	5318.1 [4]
pre-exponential factor (mass specific)	k_0'	m³/kg_{Cat} s	141.1 [4]
pre-exponential factor (fluid volume specific)	k_0	1/s	$1.82 \cdot 10^5$

reaction enthalpy species A	$\Delta H_{r,A}$	kJ/mol$_A$	-23.4 [5]
		J/kg$_A$	$-1.7 \cdot 10^5$

(1)…QICPIC measurements of sieved particle fraction of catalyst

(2)…estimated assuming constant particle volume (i.e., no loss of solid material due to particle overlap)

(3)…corresponds to a molar concentration of 1 mol/l

(4)…determined experimentally by Lichtenegger [18]

(5)…calculated with Hess law; values for enthalpy of formation and enthalpy of fusion have been obtained from literature [19,20].

4.2 Dimensionless Numbers

With the quantities provided in the tables 4.1 to 4.3 one can calculate the dimensionless numbers describing the process, which already allows some conclusion in the importance of various effects (see Table 4-4, the therefore needed equations are collected in Table 3-1).

Table 4-4: Range of dimensionless numbers describing the process

Dimensionless number	Calculated values	Typical range [21],[22]	Conclusion
N	10	1.5 … 1,000	wall effects are important
Z	60	50 … 1,000	full simulation of particle bed computationally challenging
ε	0.5 … 0.27	0.7 … 0.36	0-15% swollen particles, considerable bed compaction
Re_p	0.5-5	1 … 1,000	low-Re flow regime
Pr	4.3	1 … 10	thermal boundary layer thinner than fluid-dynamic boundary layer
Sc	410	50 … 700	very thin concentration boundary layer, high resolution needed
Da	0.027	0.001 … 100	slow reaction compared to flow
ΔT_{ad} (β)	12.3K (0.036)	(0 … 10)	moderate temperature rise: no update of fluid properties considered, however $r_{A,max} \sim 1.68 \cdot r_{A,T0}$

4.3 Grid Resolution

Considerations regarding grid resolution should address the suitability for a quantitatively correct description of the flow field. The flow regime, represented by the Reynolds number Re_p = 1 … 5, can be specified as laminar. This means that no flow effects with very short time- or length scale (like stochastic velocity fluctuations and stochastic creation or extinction of eddies) have to be expected. Clearly, the required grid resolution to obtain a correct flow field is relatively low. In what follows, the required grid resolution is estimated based on considerations for the heat and mass transfer. Our analysis follows that presented by Derksen [5].

Since the reactive zone is the particle surface and the most interesting transport processes take place in the proximity to the surface, the main focus regarding grid resolution lies in this particular zone. In the meshing process, the resolution in this zone can be controlled by introducing additional surface layers. Therefore, a common method to estimate the grid resolution needed for resolving the transport processes is to compare the thickness of the so called boundary layers. Mathematically the boundary layer is defined as zone in which field values are different from bulk values, caused by the general necessity of the fields to fulfill boundary conditions at walls. Qualitatively, the boundary layer thickness can be expressed as a function of the dimensionless numbers describing each particular field (i.e. Re_p, Pr, Sc number for describing the flow field, temperature field and concentration field, respectively) using the relation:

$$\frac{\delta}{r_p} = O\left(\frac{1}{\sqrt{Re_p}}\right); \quad \frac{\delta_T}{r_p} = O\left(\frac{1}{\sqrt{Pe}}\right); \quad \frac{\delta_c}{r_p} = O\left(\frac{1}{\sqrt{Pe_m}}\right) \tag{4.2}$$

with $\delta, \delta_T, \delta_c$ the flow-, temperature- and concentration boundary layer thickness and r_p the particle radius as a characteristic length scale.

Pe_m is larger than Pe and Re, indicating that the concentration boundary layer is the thinnest one, and mass transport phenomena set the resolution requirements. In this work, the attempt was made to resolve the smallest mass transport length scale by at least three grid points, which means that three surface layers were introduced with an overall layer thickness of less than the concentration boundary layer thickness. Specifically, the following estimate was used:

During a characteristic time scale for the flow past a sphere (i.e., r_p / U_S), chemical species A can diffuse over a (normalized) distance of typically $\sqrt{\pi D_A r_p / U_S} / r_p = \sqrt{2\pi / Pe_m}$.

This leads to the following estimate for the layer thickness that sets the spatial resolution Δr (with $Re_p = 1$ and $Sc = 410$, so $Pe_m = 410$):

$$\frac{\Delta r}{r_p} < \frac{1}{3}\sqrt{\frac{2\pi}{Pe_m}} < 0.041 \tag{4.3}$$

Also, a grid convergence study was performed on a benchmark case (Benchmark 1: stationary heat transfer past a single sphere, discussed in more detail in Chapter 6.1) to estimate the appropriate grid resolution for this case. This allows a rough estimate of the number of cells needed for simulations of the entire particle bed. The coarsest grid under investigation already consisted of approx. $3.75 \cdot 10^4$ cells for a single sphere. Resolving the entire particle bed (with a dimensionless bed length of $Z = 60$; $n_p = 7.6 \cdot 10^3$ particles) in the same manner would lead to a cell number of $O(10^8)$, which exceeds the limit of a currently feasible computation with our resources. In the present work, a cell number of $O(10^7)$ was considered as a feasible limit. Hence, it was decided to run the simulations for a shorter bed section, but still long enough to obtain results representative for the entire particle bed. In the work of Calis et al. [22] it is pointed out that a bed length of about 5 sphere diameters is already representative for a much longer bed (at least in terms of pressure drop). In this work we followed this consideration, and considering the restriction implied by Eqn. (4.3), a computational domain consisting of 374 particles (corresponding to $Z = 5$) and a cell count of approx. 10^7 was obtained.

4.4 Temporal Resolution

When running transient simulations with the CFD software OpenFOAM®, temporal resolution is automatically controlled by the solver. Specifically, an automatic time step adaption can be realized which is based on the Courant number:

$$Co = \frac{u\Delta t}{\Delta x} \leq Co_{max} \tag{4.4}$$

with Δt, Δx and u being and the time step, the edge length of a cell, and a typical flow speed calculated at the cell center, respectively.

Generally, $Co_{max} < 1$ is required to guarantee the stability of an explicit solver, which is the case for the current implementation. In this work, the maximum Courant number was set to $Co_{max} = 0.3$, ensuring a stable simulation.

4.5 Scaled System

In order to treat the catalytic reactor as a dimensionless system (except for temperature; there it is seen as convenient to keep the quantity T [K]), the simulation was set up in a way to match the dimensionless numbers presented in Table 4-4. Consequently, also the reactor geometry was made dimensionless with the geometric key parameter $d_p=1$. This means that the reactors geometry was scaled up by a scale factor K, which makes it necessary to also rescale other relevant process parameters accordingly.

The thermo-physical properties of the fluid and solid were not altered at all, so the dimensionless numbers Pr and Sc (which are independent of any length scale) are not affected by geometric upscaling. As temperature and species concentration is not affected by upscaling, also the quantities γ, ΔT_{ad}, $\Delta H_{r,A}$ stayed unchanged. So the affected quantities are the fluid inlet velocity U_S and the pre-exponential factor k_0 of the chemical reaction (consequently also k_0''). The therefore applied scaling restrictions are presented in the Appendix A in Table 10-2. In the subsequent part of this work, all thermo-physical properties, as well as parameters concerning the chemical reaction are referred to the "up-scaled" particle diameter $d_p=1$. Table 4-5 should be seen as the basic simulation setup (if not described differently in the specific case) for all reactive flow simulations inside a reactor.

Table 4-5: Basic simulation setup fixed bed simulations

Geometric parameters	Symbol	Unit	Value
particle diameter	d_p	m	1
reactor diameter	d_r	m	10
reactor length	Z	-	5
number of catalyst particles	n_p	-	374
Operation conditions			
inlet temperature	T_0	K	348
dimensionless concentration A inlet	$C_{A,0}$	-	1
fluid inlet velocity	U_s	m/s	$2.81 \cdot 10^{-7}$
Reaction parameters			
diffusion coefficient A	D_A	[m²/s]	$6.8 \cdot 10^{-10}$
activation energy	Ea/R	K	5318.1
stoichiometric coefficient A	v_A	-	-1
pre-exponential factor (surface area specific)	k_0''	m/s	0.0119
heat of reaction (species A)	$\Delta H_{r,A}$	J/kg$_A$	$-1.7 \cdot 10^5$

4.6 The Influence of the Reaction Enthalpy

Using the simplified model provided in Chapter 3.6.2, one can compute the maximum temperature rise due to exothermic reaction and adiabatic reactor operation (see Eqn. (3.47)), which results in a temperature rise of approximately 12K. Therefrom two major considerations regarding temperature influence on the reactors behavior can be derived:

- The thermo-physical properties of fluid and solid change relatively little over this temperature range, so they can be set to a constant value (i.e., temperature dependency of thermo-physical properties is neglected)

- The temperature dependency of the reaction rate, expressed by Arrhenius´ law, indicates that a temperature rise of ΔT_{ad}=12.3K accelerates the reaction rate by 68%. Thus, thermal modelling of the reactor is essential for a precise prediction of reactor performance.

4.7 A Comment on Non-Ideal Reactor Behavior

Due to the complex bed geometry with locally differing porosities and numerous contact zones, non-ideal hydrodynamic reactor behavior can be expected. It is known from literature [4] that reactors with a tube to particle diameter ratio $N < 15$ result in axial velocity profiles which are oscillating over the radial distance. Additionally, zones of stagnant flow and even backflow are expected at particle contact points. Thus, using a plug flow model as presented in chapter 3.6.1 might not represent the real flow field very well.

Another source of non-ideal reactor behavior is mass transport resistance, which in general is neglected in PFR models. Especially in zones of stagnant flow, mass transport is hindered because of the absence of convective mass transport and consequently the reaction rate is lowered in these zones. This clearly illustrates the need of a locally resolved flow, temperature and concentration fields, which allow a precise prediction of the real reactor performance.

5   A Virtual Particle Bed

The particle bed investigated in the simulations consists of monodisperse, non-porous spherical particles. However, experiments showed that the solvent used in the reactive system can diffuse into the particle, which leads to a significant swelling of the particles. The maximum swelling was determined to be approx. 13-15% of the particle diameter.

This effect causes considerable bed compaction, leading to oblated contact regions between particles, as well as between particles and the confining wall. These contact regions lead to a loss of available surface area, since the reacting liquid can no longer access these zones (note, that we assume that there is no diffusion into the catalyst particles on a time scale relevant for reactions). As a consequence, the overall conversion inside the reactor will decrease. Therefore, we attempted to capture changes in bed structure caused by this effect as accurate as possible.

5.1   Preparation of the Particle Bed using LIGGGHTS®

To generate the particle bed, the Discrete Elements (DEM) software LIGGGHTS®, version 3.0 is used. This software is fully parallelized, and hence can be used to study large particle systems (i.e., several millions of particles), excluding the surrounding fluid, though. LIGGGHTS® relies on a soft sphere collision modelling of spherical particles. This means that each particle is represented by its center, as well as a spherical shell that is used to simulate interaction forces with the surrounding particles (or walls). Most important, the DEM method cannot be used to compute any particle deformation due to stress. This would require an intra-particle spatial discretization, as well as the calculation of internal stresses and deformation for each single particle.

The simulation method is as follows: after inserting the particles at random points inside a large box, gravity is activated and the particle movement is calculated by solving Newton's equation of motion (both translational and rotational) for each particle. In case of collision, an appropriate model is employed, in which the interaction forces between the particles are related to the overlap. Specifically, a spring-dashpot model is used to calculate normal force and normal damping, and a tangential model is used to calculate shear force and tangential damping. For this approach a high temporal resolution is crucial, so that the force-overlap relation can be updated several (i.e., approximately 50) times during the particle contact time. For basic literature of DEM modelling the reader is referred to Luding [23].

Detailed documentation of the software, as well as the used models can be found online (https://github.com/CFDEMproject). Table 5-1 summarizes the most important model specifications and material properties. The complete simulation setup is provided in Appendix D (see chapter 13).

Table 5-1: Summary of the simulation setup for preparing the particle bed.

	model settings
domain boundary	fixed wall (cylinder) in x and y, periodic in z
force model	hertz tangential history
particle insertion mode	insert/pack with overlap check
compressing of the bed	deform z
particle growing	adapt atom diameter + relaxation phase
	model parameters
time step size	$2.5 \cdot 10^{-5}$
particle fraction start	0.2
	material properties
youngs modulus	$5 \cdot 10^{6}$
friction coefficient	0.05
poison ratio	0.45
restitution coefficient	0.95
	geometric properties
particle fraction compact	0.5
cylinder diameter	10
initial particle diameter	1
bed height z_{final}	5

The impact of values for the material properties on the bed structure was quantified using a sensitivity analysis. Only little effect on the overall bed porosity and the accessible surface area was detected, with a maximum deviation of 0.4% for the overall bed porosity when lowering the Youngs modulus to $5 \cdot 10^{4}$. The results of this sensitivity analysis can be found in Table 10-3 in Appendix A (section 10.4).

Altering the seed value (which can be seen as the start point for the random particle insertion method) had more influence on the bed structure than altering the material properties. Therefore, more detailed investigation on the actual material properties was not performed within this work.

5.2 Analysis of the Bed Structure

As described in section 3.1, the key parameters to describe a particle bed is the bed porosity and the available surface area of the particles. For reactors with a low tube to particle ratio, the overall porosity, as well as local porosity differs compared to the porosity of an infinitely wide bed. This is because in the former case confining walls impose a geometrical constraints of particles near the wall, i.e., particles cannot pack randomly in the direction normal to the wall. An empirical correlation to compute the overall bed porosity as a function of the tube to particle ratio N is given by Dixon [3]:

$$\varepsilon = 0.4 + \frac{0.05}{N} + \frac{0.412}{N^2} \tag{5.1}$$

$\varepsilon = 0.4$ refers to the overall bed porosity of an infinitely wide bed of monodisperse spheres in random configuration.

A common method to describe the bed structure of reactors with a low tube to particle ratio is expressing the local porosity as a function of radial distance from the confining wall [4]. In the present work the particles are allowed to overlap due to the effect of swelling, but cannot deform. This causes regions in which two (or more) particle overlap, which is unphysical since it conflicts mass conservation. Consequently, the determination of the available surface area, as well as the determination of overall or local bed porosity cannot be based on purely geometric considerations. Therefore, a Monte Carlo (MC) Integration method was implemented to avoid double counting of particle volumina, and hence yielding physically meaningful porosities and surface areas. Detailed information to this method can be found in Appendix A (section 10.3).

The swelling is modelled as a linear increase of the particle diameter after the particle insertion step is finished. During a certain number of time steps, the particles grow and have the possibility to rearrange their position inside the bounding domain. After a certain swelling diameter is reached, there is no space left for particle rearrangement, and the particles are forced to overlap at the contact zones (see Figure 5-1). Note that this is a simplified approach where no particle deformation occurs. Using the MC integration algorithm, the overlapping volume is only counted once, and hence some particle volume is subtracted. This loss of solid volume was quantified to be less than 3% for the maximum swelling under investigation, so the approach was considered to capture reasonably well the effect of particle swelling. Note that one could simply increase the particle size by a small amount in order to compensate for this loss of solid volume. However, this was not done in the present work.

Once the particle bed is generated using LIGGGHTS®, its geometry can be transformed to obtain the fluid domain on which the CFD simulations are applied. This is carried out by use of Matlab interface scripts (provided in Appendix B).
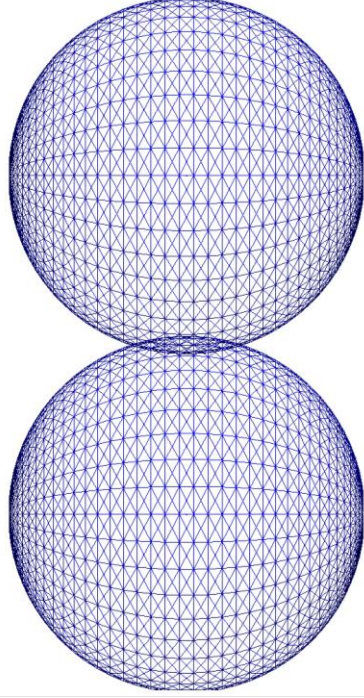
Figure 5-1: Particle overlap due to swelling of particles.

To investigate how the particle swelling affects the bed structure, three different swelling scenarios (i.e., 5, 10 and 15% increase of particle diameter) were investigated numerically. Results are summarized in Table 5-2. The base case (non-swollen) refers to the overall bed porosity ε obtained by experimental filling of the reactor with dry catalyst particles.

Table 5-2: Geometric bed properties for simulated particle beds at different swelling rates

swelling	0%	5%	10%	15%
overall bed porosity ε	0.521	0.443	0.365	0.286
particle fraction φ_p	0.479	0.557	0.635	0.714
accessibility factor η_{Surf}	0.95	0.95	0.85	0.72

The trend of η_{Surf} indicated in Table 5-2 leads to the reasoning that up to a swelling rate of 5% the particles still have enough space inside the reaction vessel to rearrange themselves rather than undergo a forced particle overlap. In other words, the original (i.e., 0% swollen) packing does not represent a packing "saturated" with particles. However, swelling rates ≥ 10% already imply considerable loss of active surface, amounting to 28% at a swelling rate of 15%.

$$Z = \frac{d/2 - r}{d_p} \ [-]$$
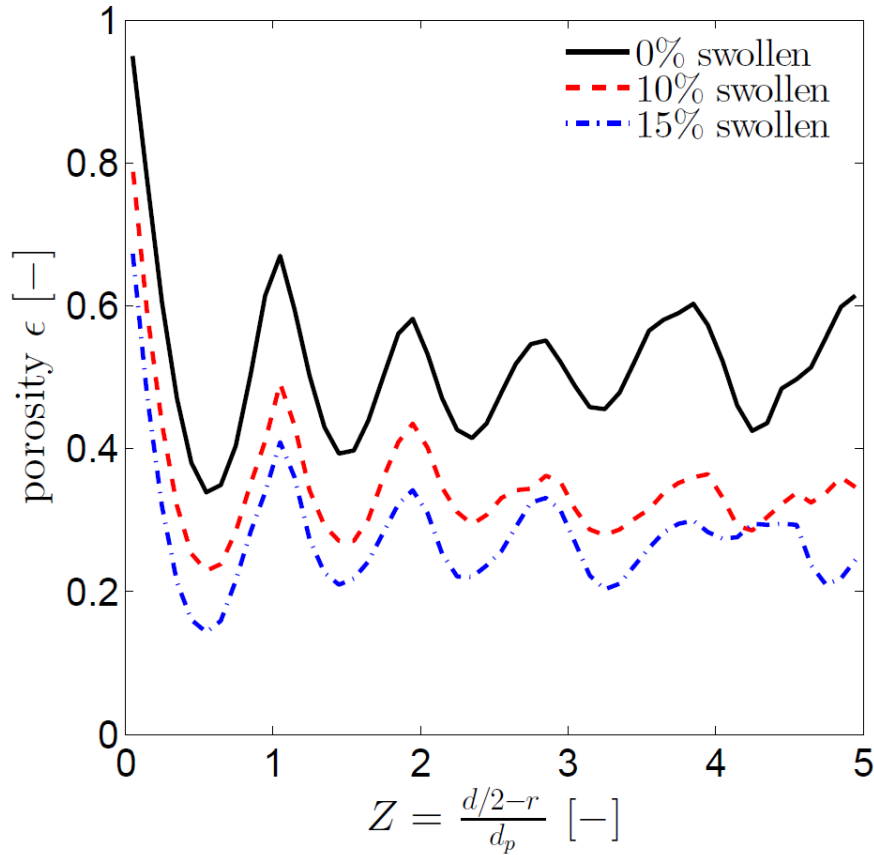
Figure 5-2: Local porosity as function of radial distance from the confining wall at different swelling rates; initial bed porosity ε = 0.52; N = 10;

In Figure 5-2, the local porosity distribution is shown for different swelling rates. The typical oscillation of local porosity encountered in reactors with low tube to particle ratio arises from the effect that the particles next to the confining wall are forced into a nearly structured position by the other particles. In the close proximity of the wall the local porosity is the highest, further peaks of high local porosity appear at distances of being integer multiples of the particle diameter.

Even at the highest investigated swelling rates, the oscillation of the local porosity are visible, although less pronounced, and propagate to the center of the reactor. Especially in the proximity of the confining wall, the local porosity affects the fluid flow since in regions of high porosity the pressure drop diminishes. Consequently, the flow rate increases in these zones, known as wall effect. Due to this, the local hydraulic residence time is lower than the overall one, leading to difficulties when trying to predict the reactor performance by a simple plug flow model [4].

6   Benchmark Simulations

Most simulations were run using a transient solver for compressible fluids, including buoyancy effects (i.e., the *buoyant-PIMPLE* solver available in OpenFOAM®). In the case of conjugate heat transfer simulations, a multi-region solver was used (i.e., *chtMultiRegion*) which couples the energy equation of the fluid region with the heat conduction equation in the solid region. For the cases where chemical reaction is involved, the *buoyant-PIMPLE* solver was adapted to also solve the species transport equation (3.21) together with the energy equation (3.22). In this solver also a model for the chemical reaction was included, and the solver is denoted as *PIMPLEreact* in what follows. With respect to the latter, an implicit algorithm was used to treat the reaction term, and hence to improve solver stability. To model the chemical reaction, a sink for chemical species A is introduced in the cell centers of the first cell layer next to reactive surfaces. This implementation routine hands over a surface reaction to the fluid region, so that the concentration field can be solved within the fluid region. This is done by modifying the expression for the reaction rate, which takes the following form:

$$r_A \left[\frac{kg_A}{m^3_l \, s} \right] = k_0'' \, \frac{A_{cell}}{V_{cell}} \, e^{-\frac{Ea}{RT_{cell}}} \, C_A \qquad (6.1)$$

Thus, in OpenFOAM® the surface-area specific pre-exponential factor has to be specified, but the result for the reaction rate is converted to the fluid-volume specific form and applied to the cells next to reacting surfaces. The most important solver settings are summarized in Table 6-1.

Table 6-1: Used solvers and settings in OpenFOAM

Temporal discretization	backward
Gradient	Gauss linear
Divergence	Gauss limited linear 1.0
Laplacian	Gauss linear limited corrected 0.333
Surface Normal Gradient	Limited corrected 0.333
Solver	*Buoyant-PIMPLE/ chtMultiRegion/ PIMPLEreact*
PIMPLE correctors /outer correctors	2
PIMPLE non orthogonal correctors	1
Pressure solver	GAMG
Solver for $\mathbf{u}$, C, h	PBiCG
Under-relaxation coefficients	$\mathbf{u} = 0.4$; $h = 0.5$;

In what follows, a number of benchmark cases are discussed that should illustrate the correct implementation of the models.

6.1   Benchmark 1: Stationary Heat Transfer from a Single Sphere

To validate solver and solver settings, the case of conjugate heat transfer of a fluid passing past a heated single sphere was studied, which requires spatial discretization of solid and fluid region (referred to as multi-region setup) and the use of the *chtMultiRegion* solver. A sphere was placed in the axial and radial center of a cylindrical flow channel (with dimensionless diameter of 3 and length of 6, dimensions are normalized with the sphere diameter). In particular, the simulation was set up to mimic stationary heat transfer, so the fluid inlet temperature and the temperature of the solid sphere were set to a constant value. As boundary conditions for the flow field, uniform inlet velocity and free outlet (relative pressure = 0) were set. On the wall of the flow channel, a *zeroGradient* condition was used so that the wall doesn´t affect the internal flow field. Thus, the cylinder approximates an infinitely wide channel (see Table 6-2). The thermo-physical properties were set according to Table 4-2.

Table 6-2: Benchmark 1: Simulation setup

field	internal	inlet	outlet	wall	solid
U_z	$-2.81 \cdot 10^{-7}$	uniform/$-2.81 \cdot 10^{-7}$	inletOutlet/ 0	slip	fixedValue/ 0
p	0	zeroGradient	fixedValue/0	slip	calculated
p_rgh	0	zeroGradient	fixedValue/0	slip	zeroGradient
T	300	fixedValue/300	inletOutlet/300	slip	fixedValue/400

In the simulations, the result of interest is the Nusselt number which represents a dimensionless heat transfer coefficient between solid and fluid. The heat transfer coefficients are defined as:

$$Nu = \frac{\alpha}{\lambda_l d_p} \tag{6.2}$$

$$\alpha = \frac{Q}{(T_S - T_0)A_S} \tag{6.3}$$

where Q is the total heat flux, T_S is the constant surface temperature of the sphere and A_S is the surface area of the sphere.

In OpenFOAM®, a post processing utility is implemented which allows a direct calculation of Q (denoted as *wallHeatFlux*), so in this case the calculation of Nu from simulation results is straightforward.

For the problem of stationary heat transfer there exist well established correlations predicting the heat transfer coefficient, which are valid for spherical particles. The Nusselt number obtained from simulation results was compared to the well-known empirical correlation provided by Ranz-Marshall [24]:

$$Nu = 2 + 0.6\, Re_p^{1/2} Pr^{1/3} \tag{6.4}$$

This case was also used for a grid convergence study, for which a detailed description of the applied procedure can be found in [25]. This concept presumes that if simulation results are in an asymptotic range, a finer grid consequently leads to a more exact solution. As a results, a *Grid Convergence Index* (GCI) is calculated, indicating how close the numeric results obtained with a "coarse" grid are to the results obtained with an "infinitely fine" grid. In other words, the GCI represents the error band of the obtained numeric result.

Table 6-3: Benchmark1: Grid convergence study

$Re_p = 1$	$Pr = 4.28$	$d_p = 1$m		
$T_S = 400$K (fixed value)	$T_0 = 300$K			
Case name	**Grid resolution**	**n Cells**	**Nu**	**GCI**
048_Re1_1	10_10_20	$3.75 \cdot 10^4$	2.78	
048_Re1_2	15_15_30	$9.42 \cdot 10^4$	2.79	2.5%
048_Re1_3	20_20_45	$2.04 \cdot 10^5$	2.79	2.1%
Theoretic result for *Nu* at infinite grid resolution			**2.84**	
Nu predicted by Ranz-Marshall			**2.86**	

As can be seen from Table 6-3, the numeric value for *Nu* is in good agreement with the predicted value, even at the coarsest studied grid resolution (for velocity and temperature profiles also see Figure 6-1 panel a and c). To verify if the used solver settings are also applicable for the more challenging case of heat transfer at a higher flow velocity, simulations were also performed for $Re_p = 10$ and $Re_p = 100$.

In Figure 6-1 one can observe how an increasing Reynolds number narrows the velocity, as well as the temperature boundary layer thickness, leading to steeper gradients of the mentioned

fields. This states the need of a higher spatial resolution for exact results. Accordingly, for computing the fields shown in Figure 6-1 (panel b and d), a grid resolution of 20_20_45 was used and compared to 10_10_20 for the base case. Whereas at $Re_p = 1$ a nearly symmetric flow profile (indicating *creeping flow* behavior) can be observed, at $Re_p = 100$ flow detachment and backflow in the rear section of the sphere is observed. At $Re_p = 1$ the temperature profile differs from the shape of the velocity profile caused by the significant contribution of heat conduction to overall heat transfer. Contrary to that, it follows the shape of the temperature profile at $Re_p = 100$, indicating that inside the fluid region, convective heat transport is predominating rather than heat conduction.
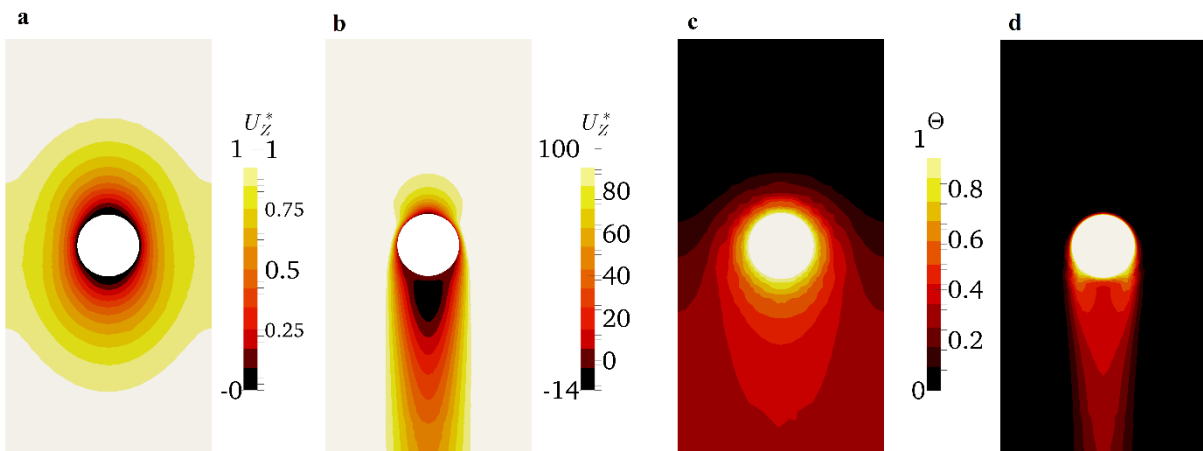
Figure 6-1: a) velocity profile $Re_p = 1$; b) velocity profile $Re_p = 100$; c) temperature profile $Re_p = 1$; d) temperature profile $Re_p = 100$; flow from above, Pr = 4.3; velocity and temperature are normalized with inlet values and the maximum temperature difference in the system, respectively.

6.2   Benchmark 2: Transient Heat Transfer from a Single Sphere

In this chapter, the performed simulations should address transient investigations of conjugate heat transfer during the cooling process of a single particles. The simulation setup is identical to that shown in Table 6-2, but contrary to the former case, the sphere temperature is now varying and a boundary condition is set to couple internal and external heat transport. This leads to the formation of a non-uniform temperature profile inside the sphere. The main objective of these simulations is to investigate if the analytical solution provided by Crank [26] can be used to predict a heating/cooling process (i.e. cooling time and internal temperature distribution inside sphere) under the present process conditions. This analytical solution is limited to radial heat conduction only, and hence treats the cooling process as a 1-D problem. Thus, the heat transport equation inside the sphere reduces to

$$\frac{\partial T}{\partial t} = a_s \left(\frac{\partial^2 T}{\partial r^2} + \frac{2}{r} \frac{\partial T}{\partial r} \right) . \tag{6.5}$$

with a_s being the thermal diffusivity of the sphere and r the radial position within the sphere.

As boundary condition (BC), a continuous heat flux across the interface (also known as Cauchy BC) is applied:

$$-a_s \frac{\partial T}{\partial r} = \alpha (T_S - T_0) \tag{6.6}$$

here, T_S is the actual surface temperature of the sphere at any time.

For a dimensionless formulation of the dependencies it is convenient to introduce the following quantities:

Dimensionless temperature (starting at 1 and approaching 0 after infinite cooling time)

$$\theta = \frac{T - T_0}{T_{S,0} - T_0} \tag{6.7}$$

Biot number (ratio of transferred heat and heat conduction inside the sphere)

$$Bi = \frac{\alpha d_p}{\lambda_s} \tag{6.8}$$

Fourier number (conduction rate over stored enthalpy; represents a dimensionless time)

$$Fo = \frac{4 a_s t}{d_p^2} \tag{6.9}$$

By applying Eqn. (6.7)-(6.9), and after some substitutions, one can derive an analytical solution for the temperature distribution within the sphere as a function of Bi and Fo:

$$\theta = Bi \frac{d_p}{r} \sum_{n=1}^{\infty} \frac{\exp(-Fo\beta_n^2)}{\left[\beta_n^2 + Bi(Bi-1)\right]} \frac{\sin\left(\beta_n 2r / d_p\right)}{\sin(\beta_n)} \tag{6.10}$$

with β_n being the roots of the function:

$$\beta_n \cot \beta_n + Bi - 1 = 0 \tag{6.11}$$

The problem when using this model is that the heat transfer coefficient is not known a priori, and hence an estimate must be made for α (or accordingly, for the Biot number). This can be done by using the Ranz-Marshall correlation presented in Eqn. (6.4). However, this correlation was developed for stationary heat transfer which assumes a fully developed temperature boundary layer. We will show that this assumption is problematic for low Reynolds numbers. In addition, the Ranz-Marshall correlation ignores the existence of internal heat transfer resistance, and hence the fluctuation of the surface temperature over the sphere's surface.

For conjugate heat transfer between liquid and solid phase, the ratio of fluid conductivity over solid conductivity can be used to categorize the heat transfer process into two regimes [13]:

$$\zeta = \frac{Bi}{Nu} = \frac{\lambda_l}{\lambda_s}$$
(6.12)

- In the case $\zeta < 1$ (i.e., $\lambda_l << \lambda_s$), called the ***external*** problem, the temperature distribution inside the particle is nearly isothermal and the heat transfer is limited by the liquid phase around the particle

- In the case $\zeta > 1$ (i.e., $\lambda_l >> \lambda_s$), called the ***internal*** problem, the heat transfer is limited by conductivity of the solid.

In our case, $\zeta = 0.73$ which indicates that internal heat transfer might play a considerable role for the overall heat transfer resistance in the transient cooling processes.

Simulations were run for $Re_p = 1$ and $Re_p = 100$ to investigate the influence of the flow field on the cooling process. To compute the heat transfer coefficient, we monitored the stored enthalpy within the sphere at different time steps, which leads to a mean heat transfer coefficient $\overline{\alpha}$ calculated via the following equation:

$$\overline{\alpha}_S = \frac{\dot{q}}{\Delta T} = \frac{\rho_s c_{p,s} \left[\int TdV \big|_{t_2} - \int TdV \big|_{t_1} \right]}{A_S (T_0 - \overline{\overline{T}}_S)(t_2 - t_1)}$$
(6.13)

$\overline{T}_S$ is the *surface averaged temperature*, which is described in literature to suite best for describing the local driving force for heat transfer [12]. For the use in subgrid models describing large scale granular flows, however, it is more convenient to use the *volume averaged temperature* $\overline{T}_V$ [13].

This is simply because this temperature best represents the thermal energy conveyed with the particle. To probe the difference of the two methods, $\overline{\alpha}$ was also computed using $\overline{T_V}$, with its formula analogous to (6.13):

$$\overline{\alpha}_V = \frac{\dot{q}}{\Delta T} = \frac{\rho_s c_{p,s} \left[\int TdV \big|_{t_2} - \int TdV \big|_{t_1} \right]}{A_S (T_0 - \overline{T_V})(t_2 - t_1)} \tag{6.14}$$

The simulation results regarding heat transfer coefficient, average sphere temperature and cooling time are presented in dimensionless form in what follows. Finally, the obtained simulation results can be compared to the result of model Eqn. (6.10) using the Ranz-Marshall correlation for predicting the overall heat transfer coefficient (see Figure 6-2).

One can see that the higher Re_p, the higher Nu, and consequently the shorter the cooling time to achieve a sphere temperature close to the fluid inlet temperature. At $Re_p = 100$, Nu calculated with $\overline{T_V}$ is significantly lower than Nu predicted with $\overline{T_S}$, which is also clear from a physical point of view: due to internal heat transfer resistance, the core temperature of the sphere stays at its initial value over a longer time, giving the volume averaged temperature a higher value compared to $\overline{T_S}$. Hence, the Nusselt number based on such a temperature difference is lower (since a lower driving force $\overline{T_V}$ -T_0 is assumed). The very good accordance of predicted and simulated θ at $Re_p = 100$ is an interesting finding, since the model presented by Crank assumes a radially symmetric temperature distribution within the sphere. This is not the case in reality, since the fluid flow accelerates the heat transfer at the front section of the sphere (compared to the rear section), so the internal temperature profile always shows some asymmetry, as illustrated in Figure 6-3. Fortunately, this asymmetry doesn´t really affect the overall heat transfer coefficient in a significant way.
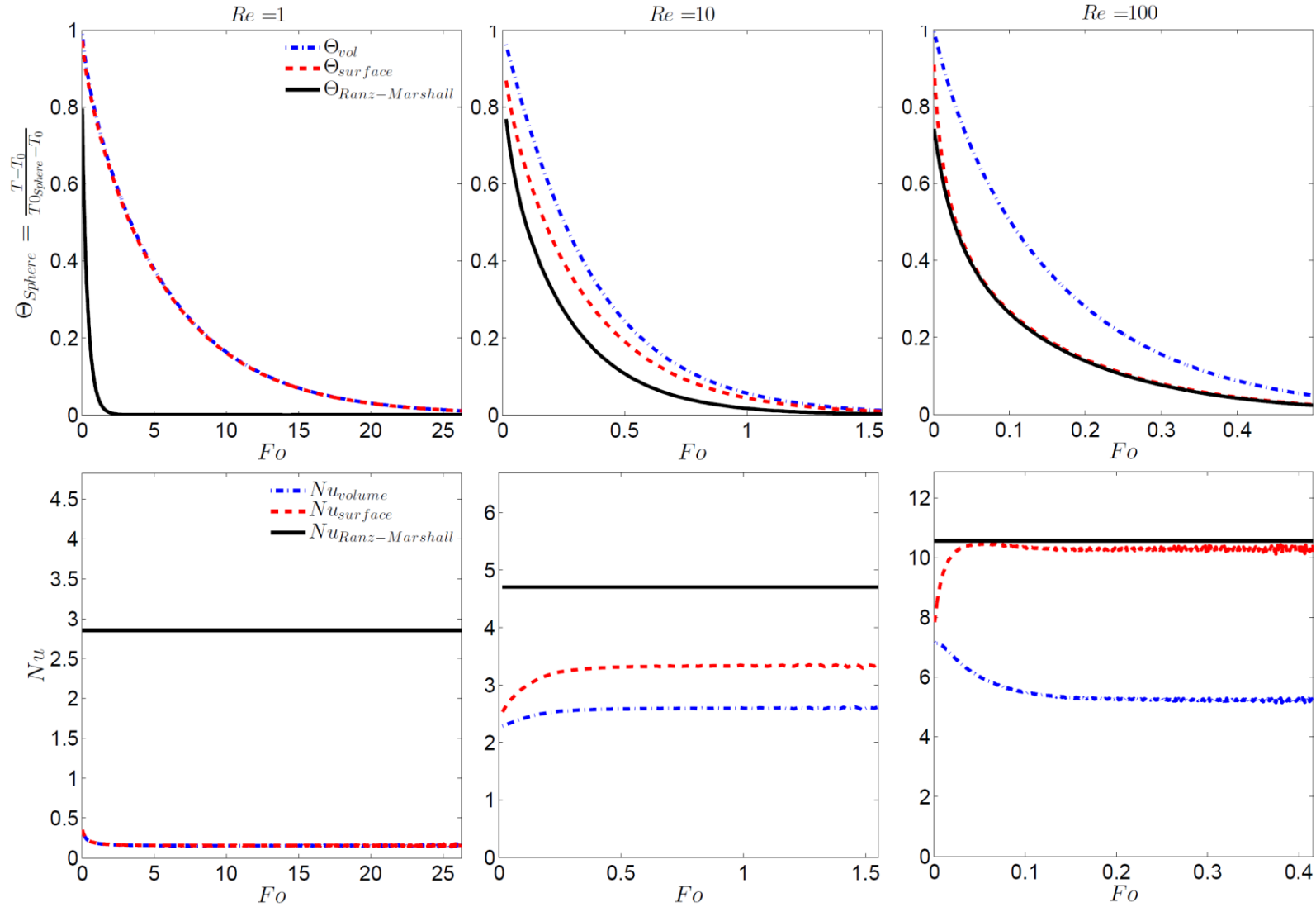
Figure 6-2: Evolution of sphere surface temperature over time for a cooling process at different Reynolds numbers (top panels); Nusselt number evolution and comparison with Ranz-Marshall correlation at different Reynolds numbers (bottom panels; the Reynolds number increases from left to right).

At lower Reynolds Numbers (i.e., Re_p=1, and Re_p=10), however, simulation data strongly differs from the analytical solution of Crank. An explanation is that the model assumes a fully developed temperature boundary layer at any time, which is never the case in a transient heat transfer processes. This inconsistency is especially problematic, since the (relative) convective transport rate (in other words the Peclet number) is small compared to the rate of heat conduction through the fluid. At low flow rates, the initially stored heat in the fluid region (caused by the initial condition of a fully developed temperature boundary layer), contributes significantly to the instantaneous heat balance, consequently strongly affecting heat transfer.
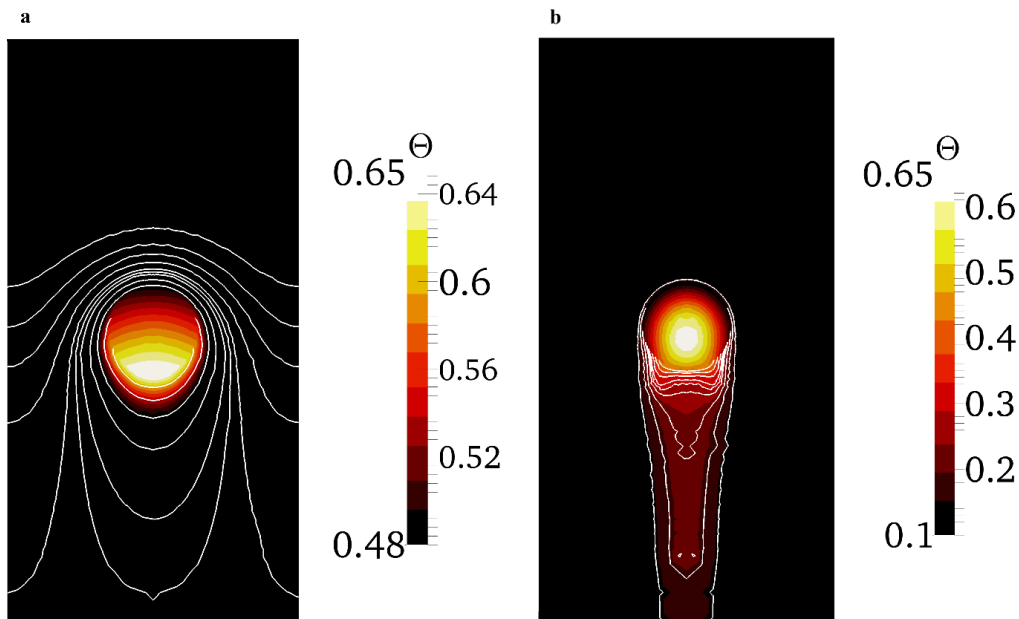
Figure 6-3: Internal and external temperature profiles for the cooling process of a single sphere with $\theta_{initial} = 1$, Pr = 4.3; flow from above; white lines are temperature iso-contours in the fluid region; a) $Re_p = 1$, Fo = 2.6; b) $Re_p = 100$, Fo = 0.17.

6.3 Benchmark 3: Reactive Laminar Pipe Flow

To verify the functional interaction between mass transport, heat transport and chemical reaction on a descriptive example, a reactive flow through a wedge of a pipe was studied. Results were compared to the predictions by the APFR model presented in chapter 3.6. The pipe geometry was set to length $L = 5$, pipe diameter $d_r=1$ and a wedge opening angle of $\varphi = 4°$. The reactive surface was placed on the upper wall, and the reaction kinetics were set according to Table 4-5. A parabolic velocity function (describing laminar pipe flow) was imposed at the inlet:

$$U_y = 2\overline{U}\left(1-\frac{y^2}{R^2}\right) \tag{6.15}$$

$\overline{U}$ is the volume flow averaged velocity, R is the pipe radius and y is the vertical position within the pipe. The simulation setup is summarized in Table 6-4.

To compute the achieved conversion at the reactor outlet from DNS results, the flow averaged concentration (also known as cup mixing concentration) is measured, which is defined as:

$$C_{A,f} = \frac{\int\limits_{A_C} u_x C_A dydz}{\int\limits_{A_C} u_x dydz} = \frac{\int\limits_{A_C} \rho_l u_x C_A dydz}{F_{V,0}} \tag{6.16}$$

This method is also used for computing the flow averaged outlet temperature T_f.

Table 6-4: Benchmark 3: Simulation setup

field	internal	inlet	outlet	sides	reactive wall
$\overline{U}$	$2.81 \cdot 10^{-7}$	groovyBC/$2.81 \cdot 10^{-7}$	inletOutlet/0	wedge	fixedValue/ 0
p	0	calculated	fixedValue/0	wedge	calculated
p_rgh	0	zeroGradient	fixedValue/0	wedge	zeroGradient
T	348	fixedValue/348	zeroGradient	wedge	zeroGradient
C	1	fixedValue/1	zeroGradient	wedge	zeroGradient
r_A	0	zeroGradient	zeroGradient	wedge	zeroGradient

The expression for the conversion described by the APFR model is analogous to Eqn. (3.52). However, for the flow through an empty duct, a different expression for *Da* is applied, accounting for the specific surface area of a pipe ($S_V = 4/d_r$) and drop out of porosity and surface availability factor:

$$Da = 4\frac{k_{T0}^{''}}{U_S} \tag{6.17}$$

Table 6-5: Benchmark 3: DNS results compared to predictions by APFR model

quantities at reactor outlet	APFR model	DNS
$C_{A,f}$	0.814	0.88
ΔT_{rise}	2.3	1.4
X_A	0.186	0.121
η_{eff}	1	**0.65**

Simulation results indicate a depletion of species A in the proximity of the reactive surface, which is increasing with increasing distance from the inlet. Consequently, also the reaction rate decreases with increasing distance. The magnitude of the reaction rate´s decay, however, is not attributed to lower bulk concentration but to a local depletion of species A caused by mass transport resistance. This becomes clear if one compares the outlet concentration in proximity to the active surface (see Figure 6-4, panel b, in which C is approximately 0.5) to the bulk concentration predicted by the APFR model (Table 6-5, C approx. 0.81).

This explains the drastic drop in reaction rate by almost the half, compared to the reaction rate at inlet conditions. In general, mass transport resistance is caused by low flow velocity in proximity to the active surface (*no slip condition*) combined with a low diffusion coefficient of species A in the fluid. Thus, time scales for mass transport are in a similar range as time scales for chemical reaction. This observation is contrary to the APFR model simplification which assumes that mass transport is much faster than chemical reaction. As a consequence, the bulk concentration might not be an adequate quantity to predict the reaction rate precisely. In other words, the APFR model will generally over-predict the conversion (see also Figure 6-5).
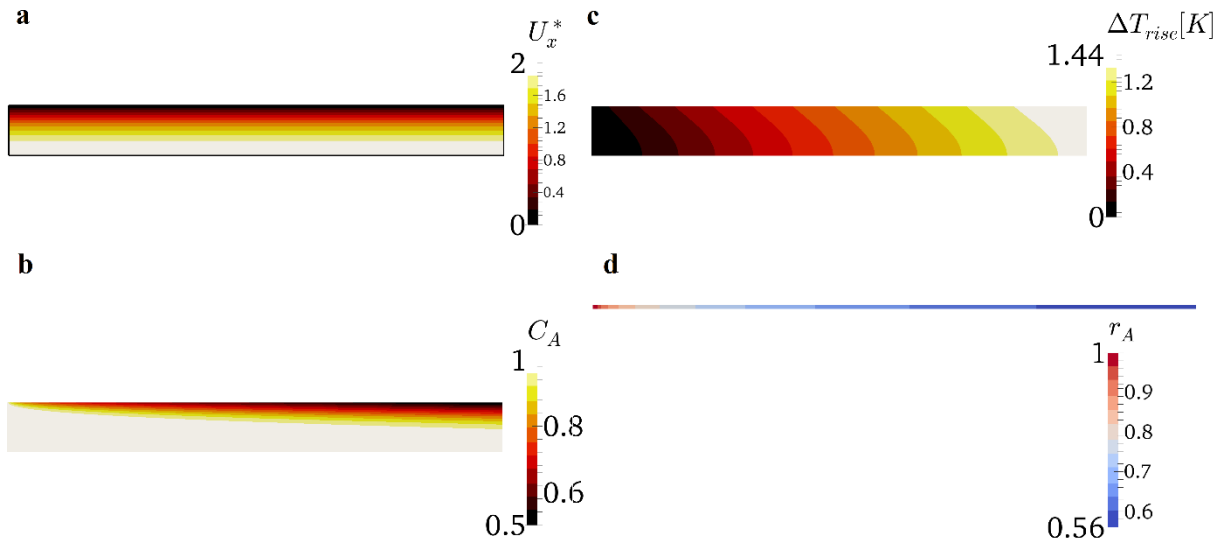
Figure 6-4: Laminar, reactive flow through a pipe wedge, flow from left to right, the upper wall is the reactive surface; Re = 1, Sc = 410; $\Delta T_{adiabat}$ = 12.3K, $C_{A,0}$ = 1; fields are normalized with inlet values; a) velocity profile b) concentration profile; c) temperature profile; d) reaction rate (view normal to wall surface)
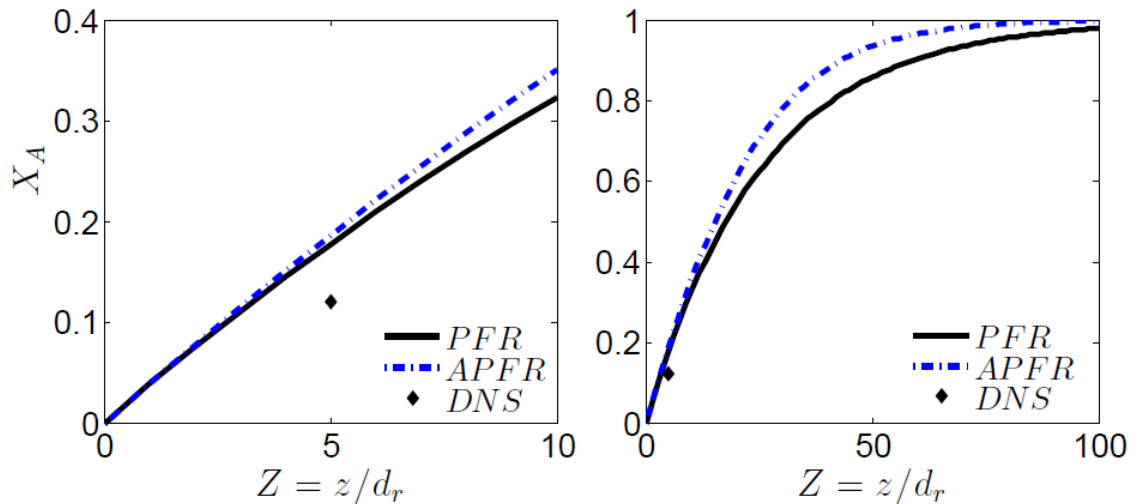
Figure 6-5: Predicted yields of PFR and APFR models compared to DNS results

From the asymptotic shape of the conversion curve in Figure 6-5 (right side, blue curve) it can also be concluded that the temperature influence on reaction rate is relatively low, since a strong temperature influence would lead to a S-shaped conversion curve [7]. Up to a conversion of approx. X_A = 0.4, the reaction rate is only affected marginally by temperature rise. The calculated effectivity factor of η_{eff} = 0.65 is estimated to represent the lower limit of η_{eff}, since here the fluid shows a laminar profile where no convective mixing occurs in the radial direction, contrary to deflected fluid flow in packed beds which increases radial dispersion [10].

6.4   Benchmark 4: Reactive Mass Transfer surrounding a Single Sphere

The simulations performed in this section treat reactive flow past a single sphere. Simulations were performed for the same geometry described in benchmark 2 for $Re_p = 1$. The active surface was placed into the cell centers of the adjacent cell layer of the sphere surface and reaction kinetics were set according to Table 4-5. As a crucial point, a "simple" single region setup is compared to a more realistic multi-region setup (all relevant parameters stay unchanged in the two cases) with the following specifications:

- Case I (single region): Reactive flow past an *insulated* sphere (no meshing of the solid region and applying a *zeroGradient* BC for *T* at the sphere surface)
- Case II (multi-region): reactive flow with conjugate heat transfer (meshing of solid region and applying a *coupled* BC for heat transport in the fluid region and heat conduction inside the solid region)

This should clarify to which extent overall conversion is affected by transient heat effects of the catalyst (if one takes into account that the heat of reaction heats up the solid particle). Besides overall conversion and the flow averaged outlet temperature (see Table 6-6), a local monitoring of the reaction rate is of particular interest since this represents the final result of interacting phenomena related to fluid flow, heat transport, mass transport and reaction kinetics. Therefore, sampling points are distributed in angular steps of $\gamma = 2°$ around the active surface in the center section of the sphere.
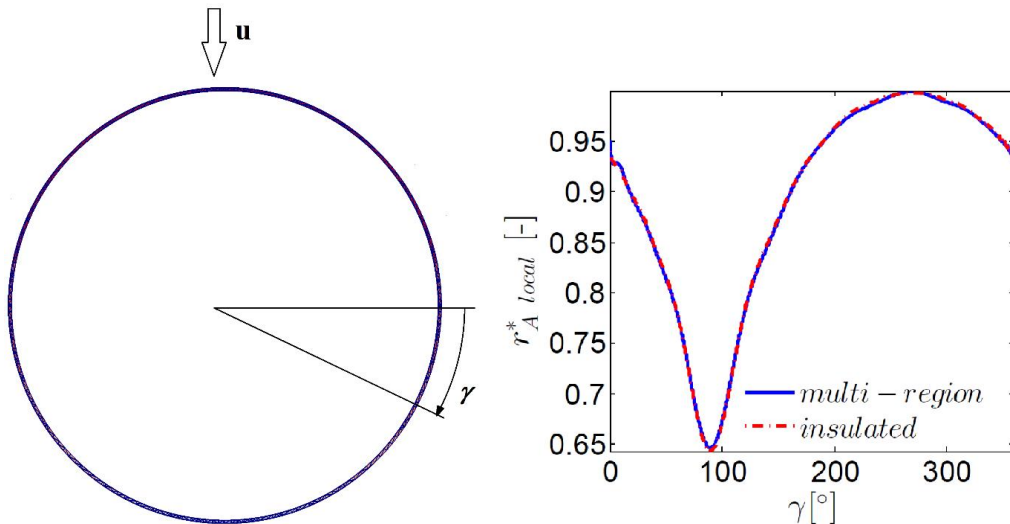
Figure 6-6: Reaction rate on active sphere surface; reaction rate is normalized with maximum reaction rate

It can be observed in both cases that reaction rate is lowest in the rear section of the sphere (corresponding to $\gamma = 90°$) due to slow transport of educts into this region. However, close to the forward stagnation point (corresponding to $\gamma = 270°$) the reaction rate is the highest. The reaction rate is nearly identical in both cases and the relative difference between highest and lowest reaction rate is approx. 35%.
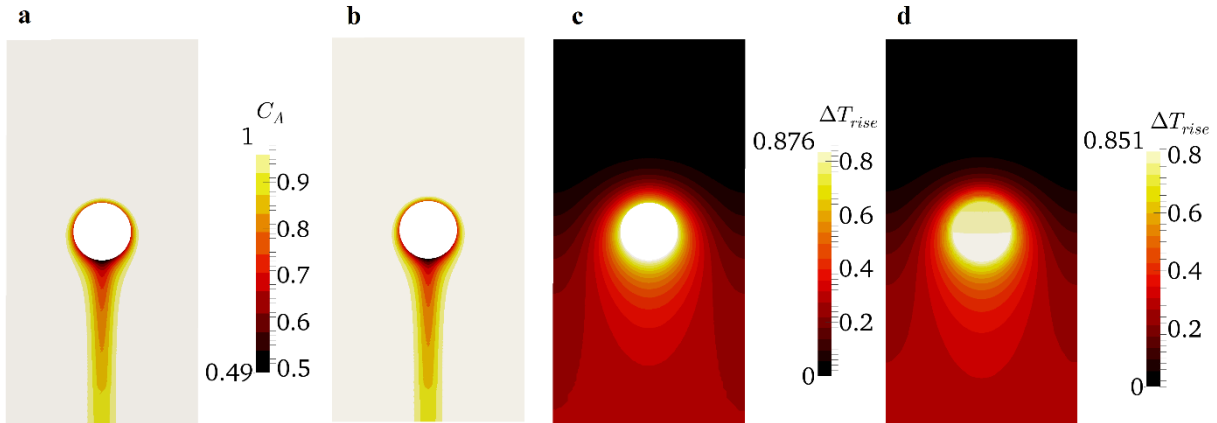
Figure 6-7: Reactive mass transfer surrounding a single sphere: comparison of case I ("insulated" sphere) with case II (conjugate heat transfer), flow from above, reactive zone is the sphere surface; a) concentration field, case I; b) concentration field, case II; c) temperature rise, case I; d) temperature rise, case II.

Case I and case II show slightly different temperature fields in the fluid region, with the maximum temperature rise being slightly higher in case I, since in this case no heat of reaction is transported into the solid region (see Figure 6-7 c).

In case II, the hottest region within the sphere is encountered in the rear section, attributed to a generally lower heat transfer in the rear section of a sphere. Concentration fields are nearly identical for both cases, since the temperature influence on reaction kinetics is relatively low (as already concluded in Benchmark 3).

Zones of species depletion are encountered in the rear section of the sphere, but mass transport limitation (species concentration locally lower than in the bulk) can be observed in all regions of the sphere's surface, even at zones of fast mass transport close to the forward stagnation point. The different length scales of mass- and energy transport (represented by *Sc* and *Pr*) are clearly indicated by the boundary layer thickness of the two respective fields. Summarizing overall simulation results (see Table 6-6), the conversion is marginally higher for case I because of the slightly higher surface temperature. A difference in the flow averaged outlet temperature was not noticed, however.

The final conclusion of these benchmark simulations is that a multi-region setup is not necessary for simulations of the entire particle bed, since the heat conducting effects within the particles (i.e., conjugate heat transfer) only affects the results marginally.

Table 6-6: Benchmark 4: Results for insulated sphere compared to multi-region sphere.

quantities at reactor outlet	single region	multi-region
$C_{A,f}$	0.99745	0.9975
ΔT_{rise}	0.269	0.269
X_A	0.00255	0.0025

7 Reactions in a Bed of Particles

The following simulations should provide detailed information on:

- Hydrodynamic effects (i.e., channeling, zones of stagnant flow, or even backflow)
- Pressure drop
- Temperature profile and flow averaged outlet temperature
- Concentration profile and overall conversion

To study the influence of flow rate and bed porosity, simulations are run for 3 different particle configurations (not swollen, 10% swollen, 15% swollen), as well as different flow rates (ranging from $Re_p = 0.5$ to $Re_p = 10$) for a bed section of height $Z=5$ and diameter $N=10$, containing 374 equally sized spheres. According to conclusions from benchmark 4, the case is set up using a single region, i.e., there is no internal mesh for the solid particles. Internal meshing of the particles is possible and has been tested, but the amount of grid cells in such a case would increase significantly.

Thermo-physical properties and reaction kinetics data are summarized in Table 4-5, and the applied boundary conditions are summarized in Table 7-1. The flow averaged outlet temperature and concentration was monitored the same way described in Eqn. (6.16). To facilitate subsequent explanations, the following case denomination is used:

- Case I: Re_p=0.5, no swelling of particles ($\varepsilon = 0.52$): represents half reactor throughput
- Case II: Re_p=1, no swelling of particles ($\varepsilon = 0.52$): represents a relatively loose bed
- Case III: Re_p=1, 15% swollen particles ($\varepsilon = 0.29$): expected operation conditions
- Case IV: Re_p=5, 15% swollen particles ($\varepsilon = 0.29$): five times higher throughput

Table 7-1: Fixed bed simulation setup: boundary conditions base case ($Re_p = 1$).

field	inlet	outlet	interface	solid
U_Z	uniform/$2.81 \cdot 10^{-7}$	inletOutlet/ 0	fixedValue/ 0	fixedValue/ 0
p	zeroGradient	fixedValue/0	calculated	calculated
p_rgh	zeroGradient	fixedValue/0	zeroGradient	zeroGradient
T	fixedValue/348	inletOutlet/348	zeroGradient	zeroGradient
C	fixedValue/0.152	inletOutlet/0	zeroGradient	zeroGradient
r_{react}	zeroGradient	zeroGradient	zeroGradient	zeroGradient

7.1 Hydrodynamics

Case II ($Re_p = 1$, $\varepsilon = 0.52$) represents a relatively loose packing, since the reference value of an infinitely wide bed of randomly packed spheres is $\varepsilon = 0.4$. In this relatively loose bed, an axial flow velocity of approx. $U_Z^*/\varepsilon = 2$ would represent the interstitial velocity and can be taken as reference value. Zones of accelerated flow are encountered in zones of high local porosity, which are expected to appear between tube wall and adjacent particles. In this case, however such zones also appear near the reactor center (see Figure 7-1). This is thought to be more likely a random effect of the particle insertion method than a physical effect. The particles are inserted at random points, so the formation of random zones with higher porosity is possible if the reactor is not completely filled with particles (this is the case if overall porosity is higher than $\varepsilon = 0.41$ for reactors with $N = 10$). The maximum axial velocity is approx. 11 times the inlet velocity, the minimum is approx. $-0.27 \cdot U_Z^*$ (these velocities are not visible in Figure 7-1, since the figure illustrates only a slice through the 3D domain).
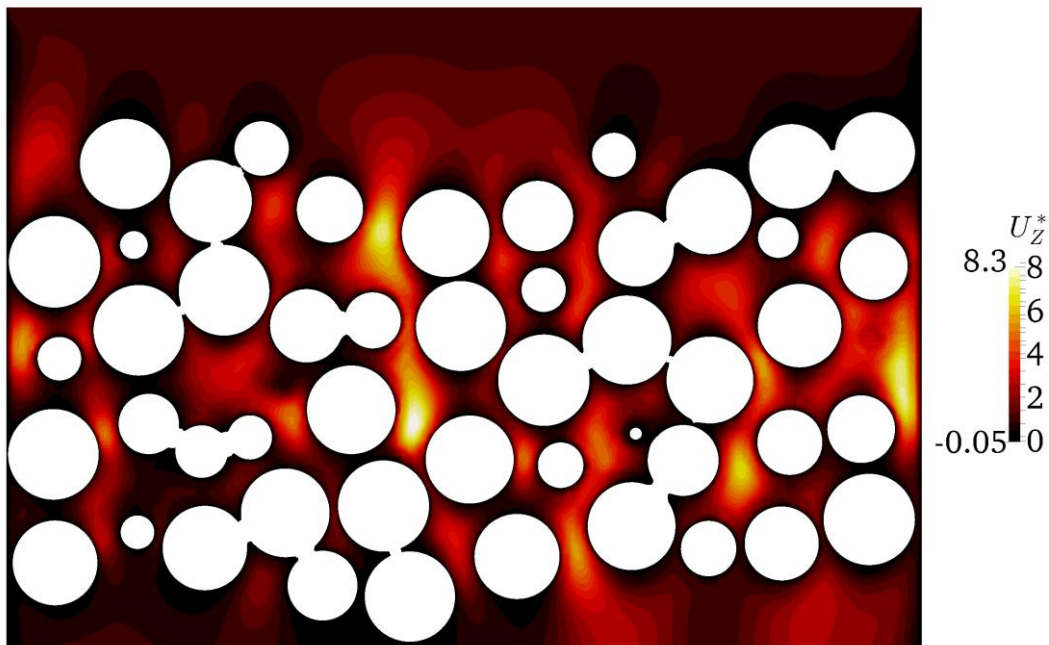
Figure 7-1: Axial velocity profile in slice through bed center; flow from above, normalized with inlet velocity, Re$_p$=1, ε=0.52;

For case III ($\varepsilon = 0.29$), the considerably compacted bed leads to a much more uneven flow field compared to case II (see Figure 7-2) with maximum axial velocities of $U_Z^* = 50$ and a minimum of $U_Z^* = -8$. This might directly affect reactor performance, since the hydraulic residence time is expected to widely differ from the average residence time. Case IV shows a similar velocity field to case III, with the main difference being the relatively large backflow zone beginning behind the sphere packing.
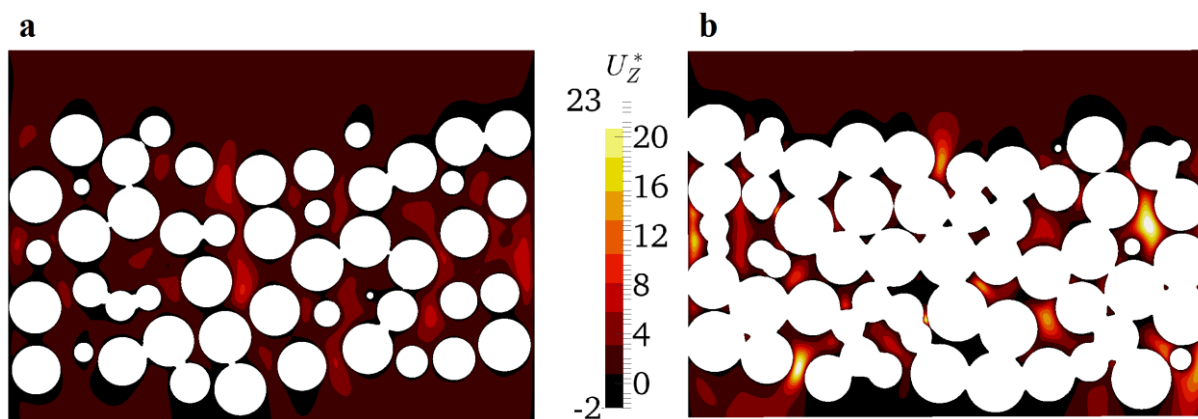
Figure 7-2: Axial velocity profiles for slices through the bed center; flow from above, normalized with inlet velocity; a) $Re_p = 1$, $\varepsilon = 0.52$; b) $Re_p = 1$, $\varepsilon = 0.29$

An effective way to point out the non-ideal hydrodynamics is to plot the zones of zero or negative axial velocity, since these zones are responsible for slow mass transport and back-mixing. Figure 7-3 indicates that for increasing flow rate and increasing particle swelling the volume of zones with stagnant flow and backflow increases.
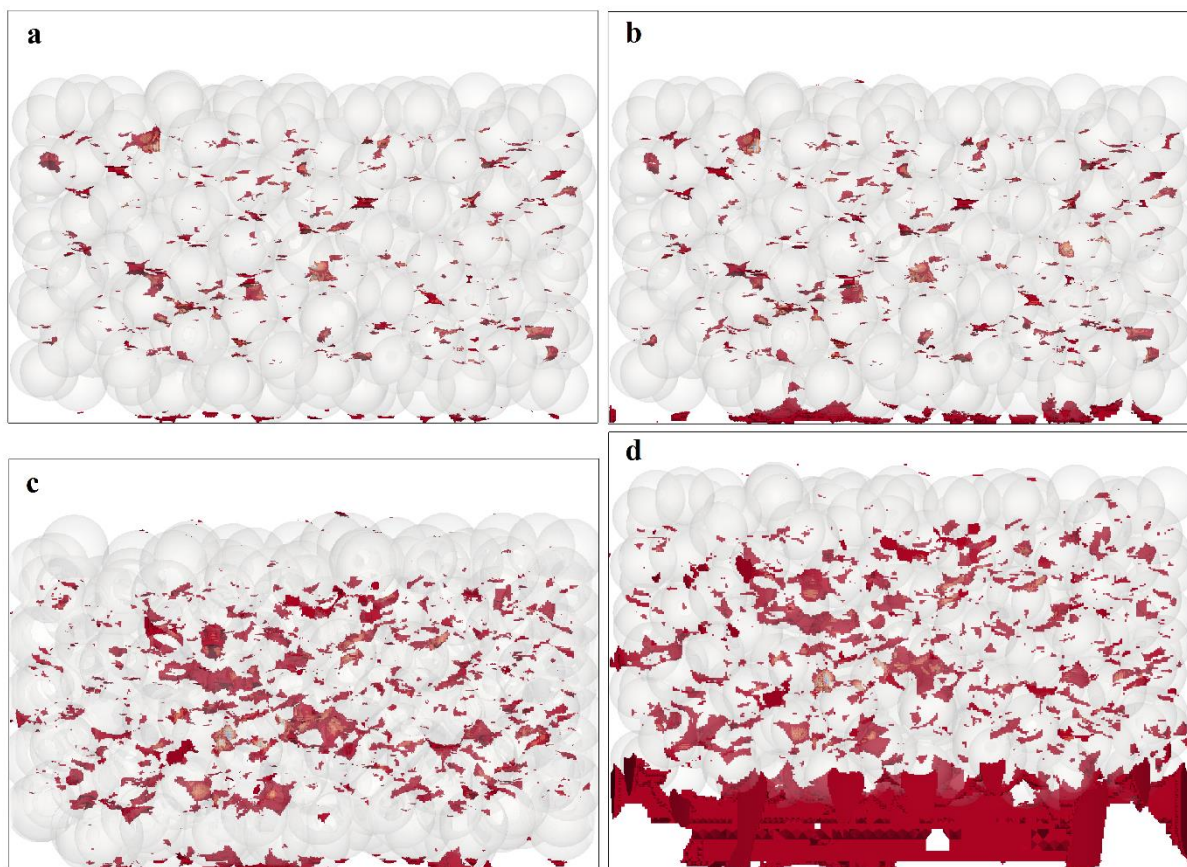
Figure 7-3: Zones of zero or negative axial velocity; flow from above; a) $Re_p = 1$, $\varepsilon = 0.52$; b) $Re_p = 5$, $\varepsilon = 0.52$; c) $Re_p = 1$, $\varepsilon = 0.29$; d) $Re_p = 5$, $\varepsilon = 0.29$

7.2 Pressure Drop

Solving the discretized Navier-Stokes equations does not only return the velocity field inside the particle bed, but also the pressure field. The overall pressure drop is then computed as difference between area averaged pressure at inlet and outlet. The DNS results are then compared to the correlative predictions provided in chapter 2.4. Additionally, pressure drop measurements were performed for the lab-scale reactor, i.e., also experimental data is available for validation of simulation results. Since the simulations are performed using the scaled bed geometry, the pressure drop results were rescaled to the real-world reactor dimensions. This is done according to Eqn. (2.5) (Euler number for low-Re flows).
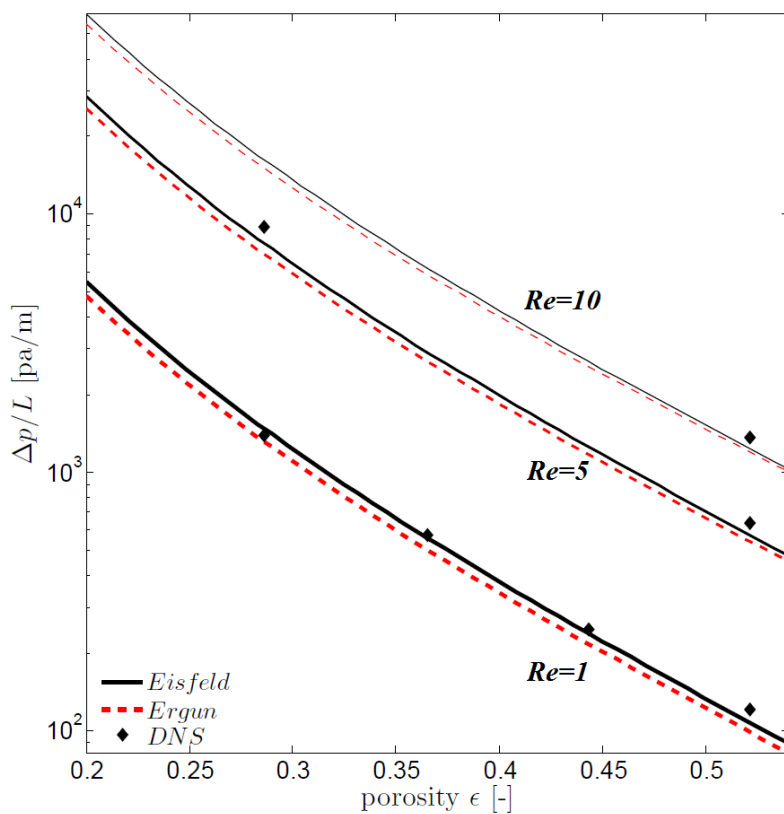
Figure 7-4: Relative pressure drop vs. overall bed porosity for different Re numbers

It can be shown that simulation results are in good agreement with predictions by Ergun or Eisfeld throughout the investigated range of Re_p and ε. Simulated pressure drop is systematically higher, except for one data point, see Figure 7-4. This is possibly caused by the contact point treatment of the meshing algorithm: it tends to connect solid zones when the gap is too small for a fluid cell to fit in. This especially happens in regions where the contact points (or -areas) between two spheres form a narrow digon. The consequence of this is a higher particle volume fraction than estimated, which naturally would lead to an increase in pressure

drop. The Ergun correlation deviates stronger from simulation results, which is clearly due to the fact that this correlation is not suitable for low-*N* reactors. Numeric results regarding hydrodynamics and pressure drop are summarized in Table 7-3.

Pressure Drop Measurements

Since online measurement of the pressure drop can provide useful information on feed pulsation, precipitation during chemical reaction or leakage, a pressure indicator (PI) was directly integrated into the experimental setup (see Figure 7-5). Consequently, the PI measures the pressure drop of the entire equipment, and not only that of the packed bed reactor. The total length of the tubing (0.76mm capillary) amounts to 2 m. Further components are 5 90°-knees and two frittes (5µm mesh size to keep the catalyst particles inside the reactor vessel) placed at inlet and outlet of the reactor vessel. The PI (range -1 – 20bar, resolution 0.1kPa and accuracy +/- 0.05%) is connected via a T-junction between feeding pump and pre-heating channels.

First, measurements were performed using an empty reaction vessel to verify *Δp* of the tubing equipment. To obtain information on the temperature dependency of the particle swelling behavior, measurements were conducted at two different temperatures (323K and 333K). Experiments were run using the pure solvent ethyl acetate to have accurate thermo-physical property at hand (i.e., fluid density, viscosity), and to avoid any falsification of the measurements due to chemical reaction (eventual precipitation, temperature rise, change in composition,…). The measurements were compared to the total calculated pressure drop, which is the sum of the calculated *Δp* of the tubing equipment and the *Δp* of the packed bed.
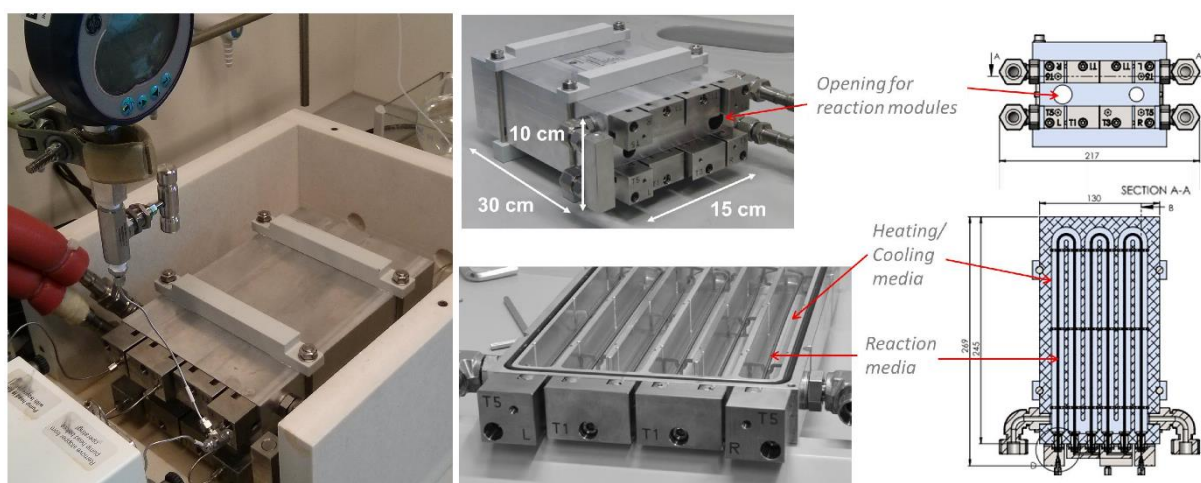
Figure 7-5: Experimental setup (left): pump, pressure indicator, reactor block and external tubing; right: detailed view of internal heating channels inside reactor block.

To calculate Δp of the tubing equipment (i.e. summing up partial pressure drops of each component), the extended Bernoulli equation is used. Assuming uniform geodetic level of the equipment and uniform velocity at inlet and outlet, the Bernoulli equation facilitates to:

$$\Delta p = \frac{\rho_l u^2}{2}\left(f\frac{l}{d} + \xi_{knee} n_{knee} + \xi_{frittes} n_{frittes} \right) \tag{7.1}$$

with f being the friction factor of the pipe (for laminar pipe flow the expression $f = 64/Re$ can be used) and ξ being the friction factor of the particular components.

The calculated Re number for the pipe flow is $Re = 70$. The friction factors were set to $\xi_{knee} = 10$ (following VDI [27] and $\xi_{frittes} = 50$, with the latter one being roughly estimated, since no data is available for low Re-numbers).

Table 7-2: Pressure drop: Measurements vs. calculated results

T [K]	323	333
Δp calculated ducts [kPa]	1.49	1.47
Δp calculated packed bed [kPa]	0.11	0.11
Δp total calculated [kPa]	**1.61**	**1.58**
Δp total measured [kPa]	**1.6**	**1.7**
relative Error [%]	**0.5**	**-6.9**

From the results presented in Table 7-2 one can see that the pressure drop of the equipment overrates pressure drop of the particle bed. Due to the small dimensions of the lab-scale equipment and low flow velocity, the total pressure drop is on a very low level compared to industrial applications involving fixed bed reactors. As a matter of fact, the used PI might not be the optimum choice for measuring pressure drop on such a low level. However, the choice was made considering that the equipment is a modular setup which allows installing additional reaction vessels in series arrangement. Also, if necessary, a second pre-heating or cooling channel can be used, and experiments can be conducted applying a higher operating pressure by installing a backpressure valve at the outlet. This could lead to a higher pressure level and also to a significantly higher pressure drop, justifying a PI with a broader operation range. It should be noted that uncertainties arise due to the low resolution of measured pressure drop (0.1kPa) for the actual range of pressure drop. However, the available information is sufficiently exact for the following observations:

- The calculated pressure drop is in good agreement with measurements (i.e., within 7% relative Error)

- The (measured) higher pressure drop when rising the fluid temperature by 10K indicates that the particle swelling rate might not have reached its maximum at 323K. Typically, if particle swelling rate and, consequently, bed porosity would be the same for both cases, one would expect a decrease in pressure drop due to the lower fluid viscosity.

- Thus, calculations under-predict pressure drop. One point of uncertainty is the pressure drop caused by the frittes, since therefore no values for ξ were provided, and it is believed that on such a low pressure level they might contribute considerably to overall pressure drop.

7.3 Chemical Reaction and Mass Transfer in a Fixed Bed

The simulations of non-isothermal reactive flow with mass transfer inside the particle bed are set up as summarized in Table 7-1 (note that the particle bed is treated as single region, so temperature is only calculated in the fluid region).

Besides the natural trend of decreasing species conversion with increasing flow velocity (i.e., decreasing Damköhler number), it can be shown that regions with low concentration of A are encountered generally at the downstream sections of particles. This is also the case for zones where particle contact points form narrow gaps (see Figure 7-6). Referring to the considerations from chapter 6.3, these zones have to be interpreted as zones of very slow mass transfer.

Another finding is that mass transfer resistance, which is present even at the forward stagnation point of the particles at $Re_p = 0.5 - 1$, disappears at $Re_p = 5$ due to a thinner flow boundary layer which enhances mass transfer (see Figure 7-6 panel d). Figure 7-6 (panel c) depicts very well the presence of wall effects: especially in densely packed beds, the fluid "slacks" through zones of higher local porosity (situated in proximity to the confining walls) without undergoing chemical reaction. Figure 7-6 (panel a) pronounces best the short length scales present at high Sc mass transport, resulting in significant local differences in species concentration, ranging from $C_A = 0.4$ up to approx. $C_A = 1$ at the outlet.
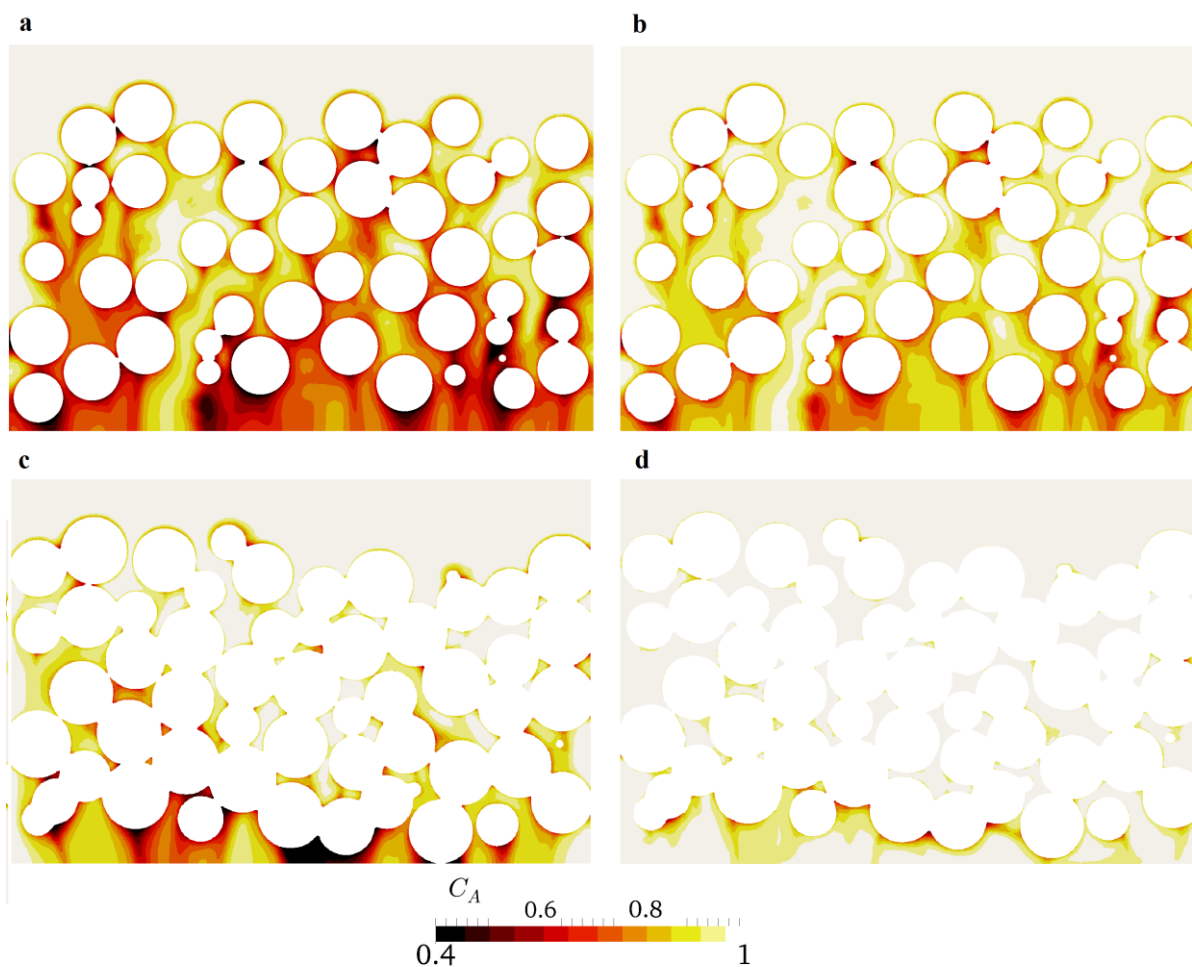
Figure 7-6: Concentration profiles (vertical slice through particle bed center); flow from above; normalized with inlet concentration; a) Re$_p$ = 0.5, ε = 0.52; b) Re$_p$ = 1, ε = 0.52; c) Re$_p$ = 1, ε = 0.29; d) Re$_p$ = 5, ε = 0.29.

Concerning heat transport, less pronounced local differences compared to mass transport are the case. Since the temperature dependency of the chemical reaction is relatively low, no formation of hot spots or ignited zones (i.e., zones of higher temperature accelerating the chemical reaction, which increases local heat production) can be detected (see Figure 7-7). Especially for the case of a rather loose bed and low flow velocity (see Figure 7-7, panel a), the temperature profile is approximately linearly increasing in axial direction. This is in good agreement to predictions using an adiabatic PFR model. A denser bed, however, shows more local differences. Analogous to the concentration field, wall effects can also be observed for the temperature field at higher flow velocities and higher bed density (see Figure 7-7, panel d: a "slack" of fluid which still has inlet temperature can be found at the outlet in proximity to the confining wall).
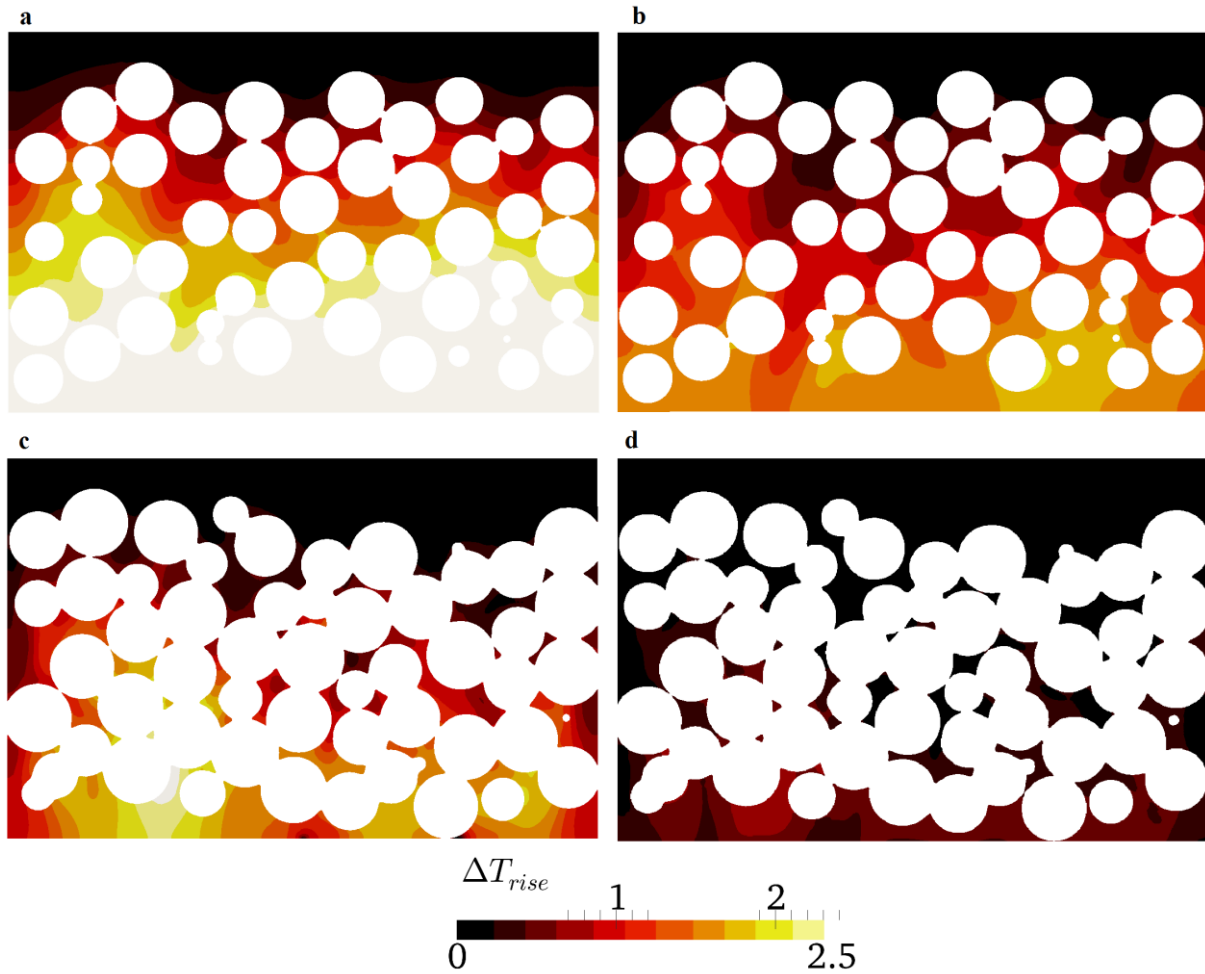
Figure 7-7: Temperature rise for an exothermic reaction in an adiabatic reactor configuration; $\Delta T_{adiabat}$ = 12.3K; vertical slice through particle bed center; flow from above; a) Re_p = 0.5, ε = 0.52; b) Re_p = 1, ε = 0.52; c) Re_p = 1, ε = 0.29; d) Re_p = 5, ε = 0.29.

To assess the influence of the observed non-ideal reactor behavior on overall reactor performance, the flow averaged concentration of *A* was determined at the outlet (corresponding to a bed length of $Z = 5$) and compared to the APFR model predictions (see Table 7-3).

Table 7-3: Summary of numeric results (hydrodynamics, pressure drop, overall reactor performance) of the fixed bed simulations (Z = 5, Case I-IV)

	Case I	Case II	Case III	Case IV
Reynolds number Re_p	$Re_p = 0.5$	$Re_p = 1$	$Re_p = 1$	$Re_p = 5$
overall bed porosity ε	$\varepsilon = 0.52$	$\varepsilon = 0.52$	$\varepsilon = 0.29$	$\varepsilon = 0.29$
Damköhler number Da	0.053	0.027	0.034	0.006
Hydrodynamics				
$U_{Z,max}$	11.6	11.5	50.0	49.5
$U_{Z,min}$	-0.26	-0.27	-8.45	-9.3
Relative pressure drop per bed length [kPa/m]				
$\Delta p_{Simulation}$	0.060	0.121	1.399	8.913
$\Delta p_{Eisfeld}$	0.053	0.108	1.481	7.7308
Δp_{Ergun}	0.049	0.099	1.321	7.029
relative Error Eisfeld	-11.6%	-11.1%	5.8%	-13.3%
relative Error Ergun	-18.9%	-18.1%	-5.6%	-21.1%
Overall reactor performance				
$X_{A,Simulation}$	0.210	0.117	0.108	0.024
$X_{A,APFR\ model}$	0.249	0.129	0.144	0.031
η_{eff}	**0.85**	**0.81**	**0.75**	**0.76**

7.4 Predictions for Overall Reactor Performance

The values of η_{eff} provided in Table 7-3 can now be used to calibrate the APFR model (i.e. incorporating η_{eff} into the APFR design equation accounts for all non-ideal effects occurring inside the reactor). Since simulation data is only given at the outlet (i.e., for $Z = 5$), the effectivity factor of the reactor is determined as:

$$\eta_{eff} = \frac{X_{A,Sim,Z=5}}{X_{A,APFR,Z=5}} \tag{7.2}$$

The values of η_{eff} already indicate that particle swelling generally affects the effectivity factor, as well as rising the flow velocity does. But it should be noted that case IV (5 times higher throughput than case III) shows a slightly higher η_{eff} than case III, which may be due to

enhanced mass transfer towards the active surface in the front sections of the particles as it was observed before (see Figure 7-6, panel c).

However, the effectivity factor must be viewed in context with the predicted conversion of the APFR model to assess the real reactor performance: the underlying theoretical model for packed beds predicts that a denser bed (represented by φ_p) up to a certain value can improve conversion. The limitation arises from the loss of available surface area, since η_{Surf} is nearly inversely proportional to φ_p. This is also indicated by the change of Da in Table 7-3. To facilitate the following comparisons, again, the in this work used definition of Da is: $Da = \dfrac{k_{T0}^{''} \varphi_p \, 6 \eta_{Surf}}{U_S}$

This theoretical trend is depicted in Figure 7-8, panel a, where the 10% swollen particle bed (ε = 0.37, red line) predicts the highest conversion, and the 15% swollen bed would still yield a higher conversion compared to the non-swollen bed. However, after calibrating the APFR model with DNS data, it becomes clear that due to the more pronounced non-ideal effects present in denser beds, the reverse trend is the case: The prediction for the conversion using the calibrated model is higher for the non-swollen bed (see Figure 7-8, panel b, thin black line, compared to thin blue line).
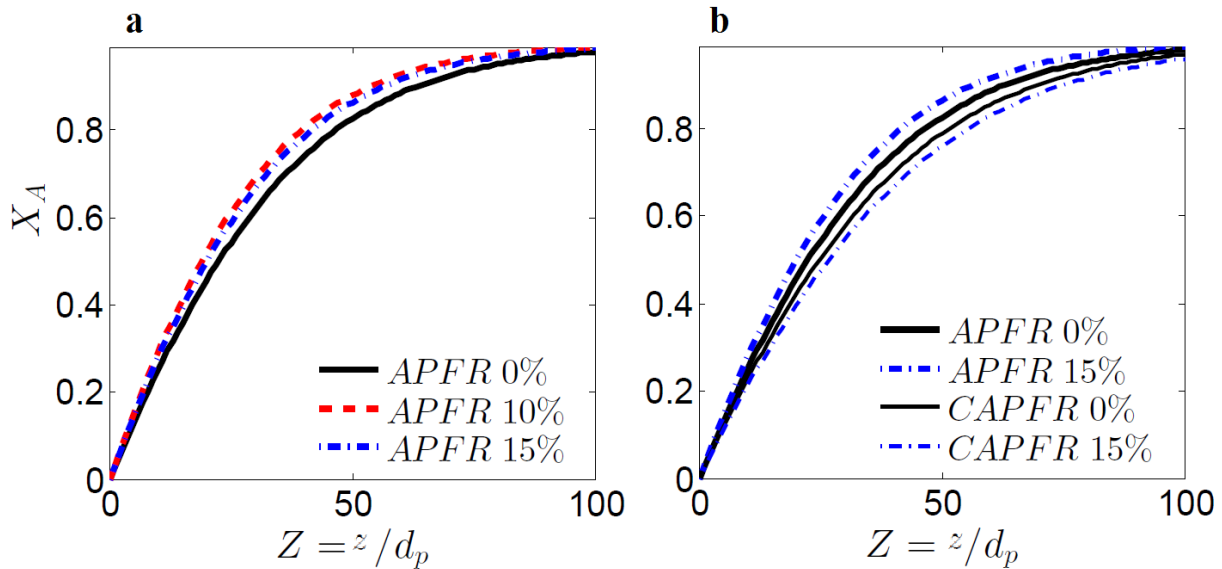
Figure 7-8: a) Influence of particle swelling rate on predicted conversion X_A;
b) APFR predictions vs. calibrated APFR Model for two different particle swelling rates

To illustrate to which extent incorporation of the non-ideal reactor behavior alters the predicted yield, the APFR model and the CAPFR model are also compared to the simplest reactor model, the PFR model. In Figure 7-9, the black lines (PFR model) represents isothermal reactor

operation. The blue lines represent adiabatic reactor operation (i.e., a perfectly insulated reactor). In all compared cases, the predicted conversion for a dimensionless bed length of $Z = 100$ lies in-between APFR and PFR model predictions. This means, in simple words, for the investigated system the non-ideal reactor behavior affects the conversion to a similar magnitude as the reaction heat enhances the conversion.

Interestingly, especially in case III (which represents best the expected operation conditions in the lab-scale reactor) the CAPFR prediction is nearly identical with the PFR prediction for conversions $X_A > 0.8$. For the sake of completeness it should be mentioned that, if assuming adiabatic reactor operation, the temperature evolution over Z would take the same shape as the lines for X_A in the APFR and CAPFR model. This is because a single relationship is left for the two quantities. For this reason, the temperature evolution is not presented here graphically.
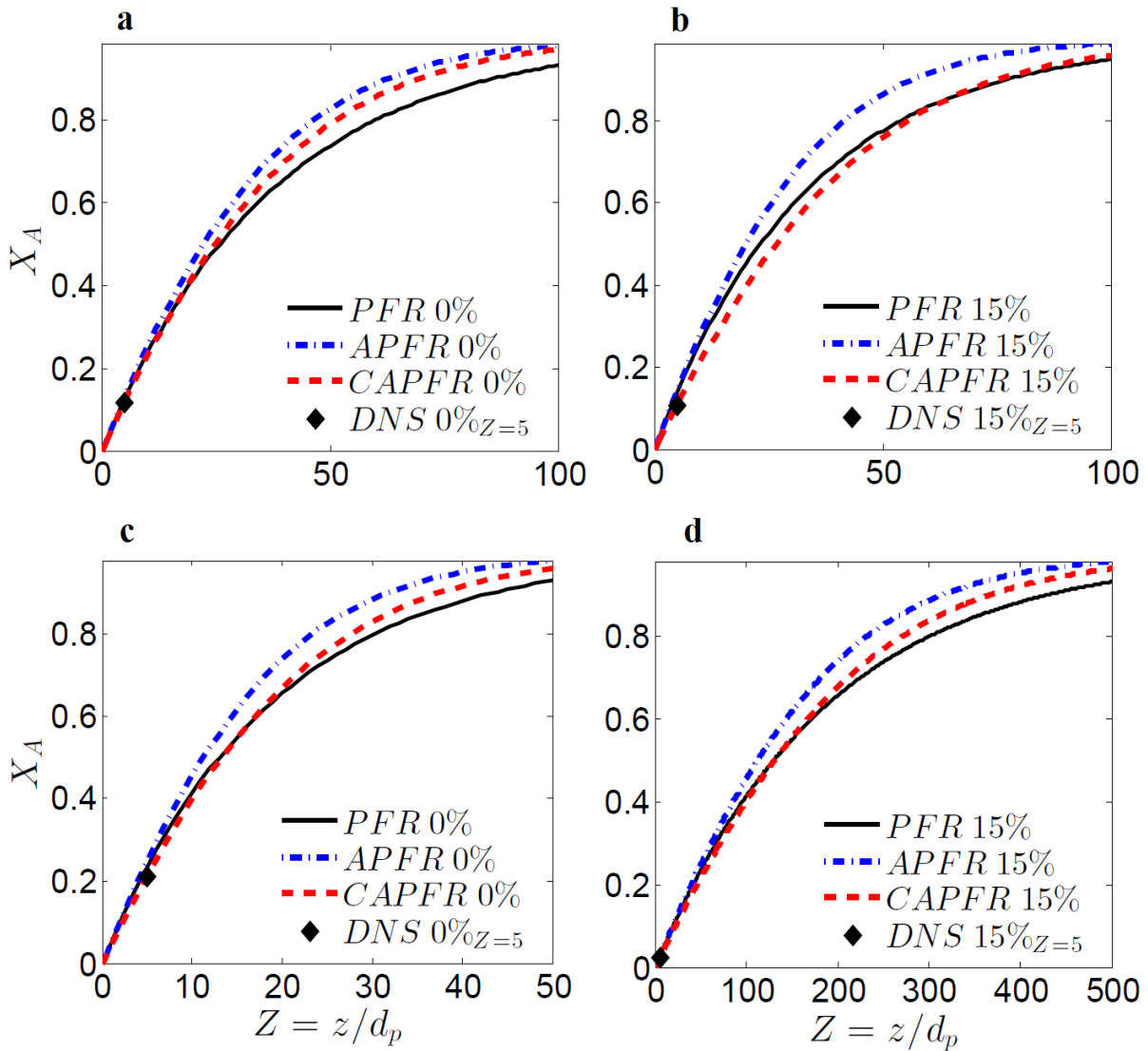
Figure 7-9: Comparison of different reactor models; a) $Re_p = 1$, $\varepsilon = 0.52$; b) $Re_p = 1$, $\varepsilon = 0.29$; c) $Re_p = 0.5$, $\varepsilon = 0.52$; d) $Re_p = 5$, $\varepsilon = 0.29$.

8   Conclusion and Outlook

In this work, a packed bed catalytic reactor with a low tube to particle diameter ratio (i.e., $N = 10$) was investigated by applying Direct Numerical Simulations (DNS). Preliminary work addressed collection of process relevant data for a model reaction (esterification of salicylic acid and acetic anhydride which yields acetylsalicylic acid), which was realized in a lab-scale reactor. Given the operation conditions of the experiments, following key results could be found:

- Reaction enthalpy: $\Delta H_{r,A} = -23.4$kJ/mol$_A$
- Adiabatic temperature rise: $\Delta T_{adiabat} = 12.3$K

Applying a dimension analysis, relevant dimensionless numbers specifying the problem were identified, with the most important ones being Re_p close to unity, $Pr = 4.3$, and $Sc = 410$. For the case of a first order heterogeneous catalytic reaction inside a bed of swelling spherical particles, a convenient formulation of the Damköhler number was derived. This dimensionless number illustrates very well the influencing parameters concerning chemical reaction:

$$Da = k_{T0}'' \varphi_p 6\eta_{Surf} / U_S .$$

The particle bed geometry was created applying a DEM simulation. It was found that the effect of particle swelling (determined experimentally to amount max. 15% of the initial particle diameter) complicates the calculation of geometric bed properties such as φ_p and η_{Surf}. Therefore, a Monte Carlo Integration method was implemented in the DEM code, which enables an exact computation of porosities and available surface area also for extremely dense beds. For the case of 15% swollen particles, computations yielded a particle volume fraction of $\varphi_p = 0.71$, as well as a loss of active surface of 28%. Investigations of the particle bed indicated oscillation of the local porosity, typical for low-N reactors. This oscillation can still be noticed at the reactor center. At higher swelling rates, the intensity of the porosity fluctuations decrease but, however, are still important.

Benchmark simulations on a single sphere revealed that under the present set of parameters, the heat transport processes taking place inside the particles (solid phase) just marginally affect the chemical reaction and the external temperature field (fluid phase). This led to the conclusion that the process can be described reasonably well by just using a single region setup (i.e., internal meshing of the particles is not necessary).

Transient conjugate heat transfer was studied more intensely, since therefore only little information can be found in literature. Results indicated that the analytic model provided by

Crank [26] for predicting the cooling time and the temperature distribution within a sphere is only valid for Re_p>100 for the set of parameters used in this work. This may be attributed to the fact that the Crank model (specifically, the Ranz-Marshall correlation used to predict heat transfer) assumes a fully developed temperature boundary layer. Such a developed situation is not achieved in transient problems involving low-Re flows, especially for liquid-particle systems. This highlights the importance of performing DNS for these type of problems.

Finally, simulations were run for a packed bed section containing five layers of spheres (374 spheres total), and local results for (i) fluid flow velocity, (ii) pressure, (iii) temperature, (iv) species concentration and (v) reaction rate were obtained. It was found that non-ideal hydrodynamic effects like channeling, stagnant flow or backflow increase with increasing Re_p and increasing particle swelling rate.

Simulation results for the pressure drop were found to be in fair agreement with the correlation provided by Eisfeld [15] (i.e., within 7.5%), whereas the commonly used Ergun correlation produced higher deviations (16%). Pressure drop measurements performed for different fluid temperatures agree with simulation results (relative errors within 7%). These measurements revealed that particle swelling is temperature dependent, and has not reached its maximum at 50°C. More generally, it can be summarized that measurements (using only solvent as fluid), as well as simulations indicate a low pressure drop of the system (i.e., $O(10^3Pa)$), even at the highest investigated bed density (i.e., $\varepsilon = 0.22$). Contrary to that, chemical reaction experiments performed within the group resulted in a significantly higher pressure drop ($O(10^4Pa)$). Thus, the origin of the high pressure drop is most likely attributed to precipitation of reaction products and blocking of the bed and/or the fritte at the reactor outlet rather than caused by bed compaction.

In the final simulations addressing mass transport and chemical reaction in the fixed bed, zones of slow mass transport were identified on a local level. Regarding local heat effects, no *hot spots* were detected and it can be stated that the local influence of reaction heat on reaction rate is of minor importance compared to local mass transfer.

To quantitatively compare real reactor performance to ideal model predictions, a reactor effectivity factor η_{eff} was introduced to calibrate PFR models by DNS data. η_{eff} was quantified to be 0.65 for the benchmark case of an adiabatic, laminar reactive pipe flow and between 0.75 and 0.85 for packed beds. It was shown that the bed compaction affects reactor performance, since a densely packed bed ($\varepsilon = 0.29$) resulted in a 6% lower conversion compared to a loose

packing ($\varepsilon = 0.29$). This is interesting since the theoretical model predictions are contrary (i.e., a denser bed results in a higher Damköhler number and consequently in a higher conversion). Thus, the Damköhler number alone is not sufficient to precisely describe reactive flow inside a low-N packed bed reactor.

Summarizing the work, all tasks were accomplished and a model to precisely predict the overall reactor performance of a low-N packed bed reactor (under the investigated parameter range) was established.

Outlook

During the meshing process of the 3D-bed geometry, shortcomings of the meshing tool were discovered, especially for the surface layer addition phase involving a multi-region setup. This is caused by the formation of contact zones which is inevitable involving swelling particles. Thus, future work should be addressed on contact point treatment to improve the layer addition process and the mesh quality in these zones.

Regarding transient, conjugate heat transfer, simulations applying a broader set of parameters would help to elucidate this particular problem. Especially a wider variation of the ratio λ_l/λ_s should be performed, since this was found in literature to be the main parameter affecting internal heat transport resistance.

Also, a systematic DNS-study of a broader range of influencing parameters (e.g. by also varying $N, Pr, Sc, \Delta H_{r,A}, E_a, k$) would be necessary to build a model that is applicable for a wide range of operating conditions and reaction kinetics.

In this work, the thermal modelling of the reactor was limited to isothermal or adiabatic reactor operation. This could be extended to the more general case of external heating or cooling since this possibly is of high importance for other reactions.

Treating the problem of high pressure drop observed during lab-experiments, it is recommended to use frittes with a coarser mesh size so that precipitation products can easily pass the outlet fritte. If particles with an even higher swelling rate (e.g. >20%) have to be used, a method to avoid complete bed compaction is the insertion of concentric bed support baffles, as described in [28]. Doing so, the newly introduced walls will counteract the extreme bed compaction.

9 References

[1]   A.G. Dixon, M.E. Taskin, M. Nijemeisland, E.H. Stitt, CFD Method To Couple Three-Dimensional Transport and Reaction inside Catalyst Particles to the Fixed Bed Flow Field, (2010) 9012–9025.

[2]   A.G. Dixon, G. Walls, H. Stanness, M. Nijemeisland, E.H. Stitt, Experimental validation of high Reynolds number CFD simulations of heat transfer in a pilot-scale fixed bed tube, Chem. Eng. J. 200-202 (2012) 344–356. doi:10.1016/j.cej.2012.06.065.

[3]   A.G. Dixon, Correlations for wall and particle shape effects on fixed bed bulk voidage, Can. J. Chem. Eng. 66 (1988) 705–708. doi:10.1002/cjce.5450660501.

[4]   T. Eppinger, K. Seidler, M. Kraume, DEM-CFD simulations of fixed bed reactors with small tube to particle diameter ratios, 166 (2011) 324–331. doi:10.1016/j.cej.2010.10.053.

[5]   J.J. Derksen, Simulations of solid–liquid mass transfer in fixed and fluidized beds, Chem. Eng. J. 255 (2014) 233–244. doi:10.1016/j.cej.2014.06.067.

[6]   M. Siebenhofer, Teaching Material for the Lecture "Reaction Engineering II," Graz University of Technology, Graz, 2013.

[7]   K.R. Westerterp, W.P. VanSwaaij, A. Beenackers, Chemical Reactor Design and Operation, John Wiley& Sons, 1984.

[8]   D.J. Gunn, Axial and radial dispersion in fixed beds, 42 (1987) 363–373.

[9]   M. Kraume, Transportvorgänge in der Verfahrenstechnik, Springer Verlag, 2012. doi:10.1007/978-3-642-25149-8.

[10]   M. Edwards, J. Richardson, Gas dispersion in packed beds, Chem. Eng. Sci. 23 (1968).

[11]   H. Tavassoli, E.A.J.F. Peters, J.A.M. Kuipers, Direct Numerical Simulation of fluid-particle heat transfer in dense arrays of non-spherical particles, Chem. Eng. Sci. (2013).

[12]   N.G. Deen, S.H.L. Kriebitzsch, M.A. van der Hoef, J.A.M. Kuipers, Direct numerical simulation of flow and heat transfer in dense fluid–particle systems, Chem. Eng. Sci. 81 (2012) 329–344. doi:10.1016/j.ces.2012.06.055.

[13]   P.A. Schmidt, R., Wittig, K. Nikrityuk, Single Particle Heating and Drying, in: P.A. Nikrityuk, B. Meyer (Eds.), Gasif. Process. Model. Simul., Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2014. doi:10.1002/9783527673186.ch5.

[14]   S. Ergun, a. a. Orning, Fluid Flow through Randomly Packed Columns and Fluidized Beds, Ind. Eng. Chem. 41 (1949) 1179–1184. doi:10.1021/ie50474a011.

[15]   B. Eisfeld, K. Schnitzlein, The Influence of confining walls on the pressure drop in packed beds, 56 (2001) 4321–4329.

[16] Rohm&Haas-Dow, Product data sheet Amberlite-IR120-H-, (2014). http://www.lenntech.de/produkte/harze/rohm-haas/rohm-haas-de.htm (accessed October 10, 2014).

[17] P. Feenstra, Personal communication, (2014).

[18] G. Lichtenegger, Personal communication, (2014).

[19] B. Schröder, L.M.N.B.F. Santos, I.M. Marrucho, J.A.P. Coutinho, Fluid Phase Equilibria Prediction of aqueous solubilities of solid carboxylic acids with COSMO-RS, 289 (2010) 140–147. doi:10.1016/j.fluid.2009.11.018.

[20] D.R. Kirklin, Enthalpy of combustion of acetylsalicylic acid, (2000) 701–709. doi:10.1006/jcht.1999.0650.

[21] J. Wang, A. Bindal, T.M. Leib, J.G. Khinast, Analysis of the complex nonlinear behavior of reacting bubble flows : steady-state multiplicity, 59 (2004) 5575–5585. doi:10.1016/j.ces.2004.07.073.

[22] H.P.A. Calis, J. Nijenhuis, B.C. Paikert, F.M. Dautzenberg, C.M. Van Den Bleek, CFD modelling and experimental validation of pressure drop and flow profile in a novel structured catalytic reactor packing, Chem. Eng. Sci. 56 (2001) 1713–1720.

[23] S. Luding, Introduction to Discrete Element Methods Basics of Contact Force Models and how to perform the Micro-Macro Transition to Continuum Theory, (2008) 785–826.

[24] W. Ranz, W. Marshall, Evaporation from drops, Chem. Eng. Prog. (1952). http://dns2.asia.edu.tw/~ysho/YSHO-English/1000 CE/PDF/Che Eng Pro48, 141.pdf (accessed February 11, 2015).

[25] P.J. Roache, Verification and Validation in Computational Science and Engineering, Hermosa Publishers, Albuquerque, New Mexico, 1998.

[26] J. Crank, The Mathematics of Diffusion, 2. ed., Oxford University Press, Oxford, 1975.

[27] VDI, VDI Heat Atlas, 10th ed., Springer Verlag, Berlin, 2006.

[28] T. Lan, S. Gerontas, G.R. Smith, J. Langdon, J.M. Ward, N.J. Titchener-Hooker, Investigating the use of column inserts to achieve better chromatographic bed support., Biotechnol. Prog. 28 (2012) 1285–91. doi:10.1002/btpr.1597.

[29] M. Zlokarnik, Dimensional Analysis and Scale-Up in Theory and Industrial Application, in: Pharm. Process Scale-Up, Marcel Dekker Inc., New York, 2001.

[30] Machiraju, Möller, Teaching Material for the Lecture "Monte Carlo Techniques- Basic Concepts," n.d.

[31] S.S. Bu, J. Yang, M. Zhou, S.Y. Li, Q.W. Wang, Z.X. Guo, On contact point modifications for forced convective heat transfer analysis in a structured packed bed of spheres, 270 (2014) 21–33. doi:10.1016/j.nucengdes.2014.01.001.

10 Appendix A

10.1 Dimensional Analysis

A dimensional analysis has been performed to isolate the key dimensionless parameters that dictate the system behavior. To formulate the dimensionless numbers, the $\prod$ -Theorem introduced by Buckingham has be used. All relevant process parameters have to been specified to generate a coefficient matrix of the quantities of interest. The number of base units defines the rank r of the matrix. . By choosing a set of r linear independent process parameters (where each of the base units needs to appear at least once in the set), a core square matrix $r \times r$ can be defined. Finally, the number of relevant process parameters n defines n-r dimensionless quantities $\prod_j$ [29].

Considering the geometric parameters, one can define the following three simplexes characterizing the proportions of the reactor [21]:

$$N = \frac{d}{d_p} \quad \text{...relative tube diameter} \tag{7.3}$$

$$Z = \frac{l}{d_p} \quad \text{...relative reactor length} \tag{7.4}$$

$$Y_p = \frac{\varphi_p}{1-\varphi_p} \quad \text{...relative particle volume per fluid volume} \tag{7.5}$$

Also, the activation energy can be made dimensionless straight forward by considering the universal gas constant R and the bulk temperature:

$$\gamma = \frac{E_a}{RT_0} \quad \text{...dimensionless activation energy} \tag{7.6}$$

Table 10-1: List of relevant input parameters for reactive non-isothermal flow in a heterogeneous, adiabatic reactor

Symbol	Parameter	Base Unit
Geometrical Parameters		
d_p	Particle diameter	m
d	Tube diameter	m
l	Tube length	m
φ_p	Particle fraction	$m^3{}_s/m^3{}_{tot}$
Flow parameters		
U_S	Superficial velocity	m/s
μ	Dynamic viscosity	kg/(m·s)
ρ_l	Fluid density	kg/m³
Thermophysical Parameters		
λ_l	Heat conductivity fluid	kg·m/ (s³·K)
$c_{p,l}$	Heat capacity fluid	m²/(s²·K)
T_0	Bulk temperature (@ inlet)	K
Thermochemical Parameters		
$c_{A,0}$	Species concentration at inlet	kg/m³
D	Diffusion coefficient	m²/s
ΔH_r	Reaction enthalpy	m²/s²
$-r_A$	Reaction rate @ T_0	kg/(m³ s)
E_a	Activation energy	m²/s²

Dimension Matrix:

Because the geometric parameters are already dimensionless, they are not considered in the subsequent analysis. Thus, the following matrices are obtained:

Base quantity		core matrix				residual matrix				
	ρ_l	d_p	μ	λ_l	U_S	T_0	$c_{p,l}$	$-r_A$	D	$c_{A,0}$	ΔH_r
[kg]	1	0	1	1	0	0	0	1	0	1	0
L [m]	-3	1	-1	1	1	0	2	-3	2	-3	2
t [s]	0	0	-1	-3	-1	0	-2	-1	-1	0	-2
T [K]	0	0	0	-1	0	1	-1	0	0	0	0

Transformed Matrix:

Operations:	ρ_l	d_p	μ	λ_l	U_S	T_0	$c_{p,l}$	$-r_A$	D	$c_{A,0}$	ΔH_r
$\underline{M}$= M-$\underline{t}$+T	1	0	0	0	-1	-2	0	0	-1	1	-2
$\underline{L}$ = L+3·$\underline{M}$+T+$\underline{t}$	0	1	0	0	-1	-2	0	-2	0	0	-2
$\underline{t}$ = t-3·T·(-1)	0	0	1	0	1	3	-1	1	1	0	2
$\underline{T}$ = T·(-1)	0	0	0	1	0	-1	1	0	0	0	0

Dimensionless Numbers

The following dimensionless quantities can be extracted from each column of the residuals of the transformed matrix:

$$\prod{}_1 = \frac{U_S \rho_l d_p}{\mu} = Re_p \tag{7.7}$$

$$\frac{\prod{}_2 \prod{}_3}{\left(\prod{}_1\right)^2} = \frac{T_0 c_p}{U_s^2} = E_{therm} \tag{7.8}$$

$$\prod{}_3 = \frac{c_{p,l} \mu}{\lambda_l} = Pr \tag{7.9}$$

$$\frac{\prod_4}{\prod_1 \cdot \prod_6} = \frac{-r_A d_p}{c_{A,0} U_S} = Da_I \tag{7.10}$$

$$\left(\prod_5\right)^{-1} = \frac{\mu}{D\rho_l} = Sc \tag{7.11}$$

$$\prod_6 = \frac{c_{A,0}}{\rho_l} = C_{A,0} \tag{7.12}$$

$$\frac{\prod_7 \cdot \prod_6}{\prod_2 \cdot \prod_3} = \frac{\Delta H_r c_{A,0}}{\rho_l c_{p,l} T_0} = \beta \tag{7.13}$$

The physical meaning of the obtained dimensionless quantities is summarized in Table 3-1.

10.2 Scaling Restrictions

Table 10-2: Scaling restrictions after geometric normalizing of the system to $d_p=1$

restriction	value model	value simulation	equation
$d_{p,Sim} = 1$	$6 \cdot 10^{-4}$	1	$K = \dfrac{d_{p,Sim}}{d_{p,M}} = 1.667 \cdot 10^3$
$Pr_M = Pr_{Sim}$	4.28	4.28	$\mu_M = \mu_{Sim}$ $c_{p,l,M} = c_{p,l,Sim}$ $\lambda_{l,M} = \lambda_{l,Sim}$
$Sc_M = Sc_{Sim}$	410	410	$\rho_{l,M} = \rho_{l,Sim}$ $D_M = D_{Sim}$
$Re_M = Re_{Sim}$	$U_{S,M} = 4.677 \cdot 10^{-4}$	$U_{S,Sim} = 2.806 \cdot 10^{-7}$	$U_{S,Sim} = \dfrac{U_M}{K}$
$Da_M = Da_{Sim}$	$k_{0,M} = 1.82 \cdot 10^5$	$k_{0,Sim} = 0.0624$	$k_{0,Sim} = \dfrac{k_{0,M}}{K^2} \eta_{Surf}$
	$k_{0,M}'' = 19.75$	$k_{0,Sim}'' = 0.0119$	$k_{0,Sim}'' = \dfrac{k_{0,Sim} d_{p,Sim}}{Y_P 6 \eta_{Surf}}$

10.3 Monte Carlo Integration

The principle of this numerical integration method is a statistical evaluation of random sampling inside a defined space. After defining a restriction which allows to divide the samples into "hit" and "miss", the number of "hits" over the number of samples returns the expectancy of an event:

$$\overline{Y} = \frac{1}{N}\sum_{i=1}^{N} Y \tag{7.14}$$

With $\overline{Y}$ the expectancy, N the number of tries and Y the number of hits.

Applying this approach, the particle fraction and the accessible particle surface of overlapping particles inside a cylindrical domain can be computed.

Surface integration

Sample points are distributed in spherical coordinates on the particle surface (i.e., random numbers are generated for $\cos(\theta)$ and $\varphi/2\pi$, the sphere radius r is held constant). More details on sampling a spherical space are provided in [30]. Geometrical functions detect the distance of the tries (on the particle surface) to the neighboring spheres, as well as to the domain walls, and count these points as "miss" if an overlap occurs. The surface integration computes the accessible surface as quantity as well as the accessibility factor η_{Surf}, which is the accessible surface divided by the total surface of all particles.

Volume integration

Random samples inside the simulation domain are generated by a random number generator. To achieve a statistically equal distribution of tries, a high number of tries $o(10^6)$ has to be applied. The compute distinguishes between "hit" (point inside particle) and "miss" (point outside particle). Consequently, in case the point is inside the intersecting volume of 2 (or more) spheres, it is only counted once. Sampling the cylindrical space is done by generating random numbers for r^2, $\varphi/2\pi$ and the axial position z to distribute the tries evenly in the domain.

Additionally, the cylindrical domain can be sampled dividing it into several annular domains, which allows to compute the radial porosity distribution of the bed (as shown in Figure 5-2 in section 5.2).

10.4 LIGGGHTS Physical Parameters- Sensitivity Analysis

Table 10-3: LIGGGHTS sensitivity analysis of physical parameters; non-swollen particles

Parameters base case	value		
Youngs Modulus	$5 \cdot 10^6$		
Friction Coefficient	0.05		
Seed value	666		
Results base case	**value**		
N particles	374		
particle fraction φ_P	0.479		
accessibility factor η_{Surf}	0.948		
Youngs Modulus	**$5 \cdot 10^8$**	**$5 \cdot 10^4$**	**$5 \cdot 10^2$**
N particles	374	374	374
particle fraction φ_P	0.479	0.481	0.479
accessibility factor η_{Surf}	0.948	0.948	0.882
relative variation φ_P [%]	0.00	0.44	0.13
Friction Coefficient	**0.1**	**0.01**	**0.001**
N particles	374	374	374
particle fraction φ_P	0.480	0.480	0.479
accessibility factor η_{Surf}	0.950	0.952	0.948
relative variation φ_P [%]	0.27	0.27	0.10
Seed value	**$1 \cdot 10^6$**	**1000**	**10**
N particles	376	375	375
particle fraction φ_P	0.484	0.483	0.483
accessibility factor η_{Surf}	0.952	0.952	0.952
relative variation φ_P [%]	1.15	0.79	0.96

10.5 Meshing Process

After the simulated creation of the particle bed with LIGGGHTS, the bed geometry has to be handed over to a meshing utility where the fluid geometry is discretized into cells where finally the discretized transport equations can be solved by the CFD program OpenFOAM.

Therefore, LIGGGHTS contains a post processing utility which provides information on size, position, force, velocity… of each single particle. By using a matlab interface, the particle positions are handed over in a way that the OpenFOAM meshing utility SnappyHexMesh can read in the particle position and generate the mesh.

Starting from a simple utility called blockMesh, the size of the overall computational domain is defined and the domain is discretized into a hexagonal mesh by specifying initial discretization settings. After this, SnappyHexMesh is applied, which can generate simple geometries like cylinders and spheres. Doing so, the reactor tube is set up as a cylinder with the diameter d and the length Z. The particles, represented as spheres with constant diameter d_p, can now be inserted into the cylindrical bounding box. The meshing process consists of three basic steps:

1) Castellated Mesh: the basic geometries and regions (fluid and solid) are specified, defined surfaces and regions can be refined.
2) Snapping: solid and fluid region can be separated from each other, or the solid region is simply cut off the fluid region. Further surface refinement is done.
3) Layering: the mesh close to the particle surface is shrunk off the surface, so that thin layers can be introduced into the resulting gap, which allows a high resolution discretization of the boundary layer.

The whole meshing process is set up in parallel so it can be run on multiple processor cores. A tubular reactor with d=10 and Z=5 contains about 374 spherical particles, the hexahedra based 3D-mesh contains between $8 \cdot 10^6$ and $15 \cdot 10^6$ cells. In Figure 10-1, a zoomed view of the mesh geometry is given.
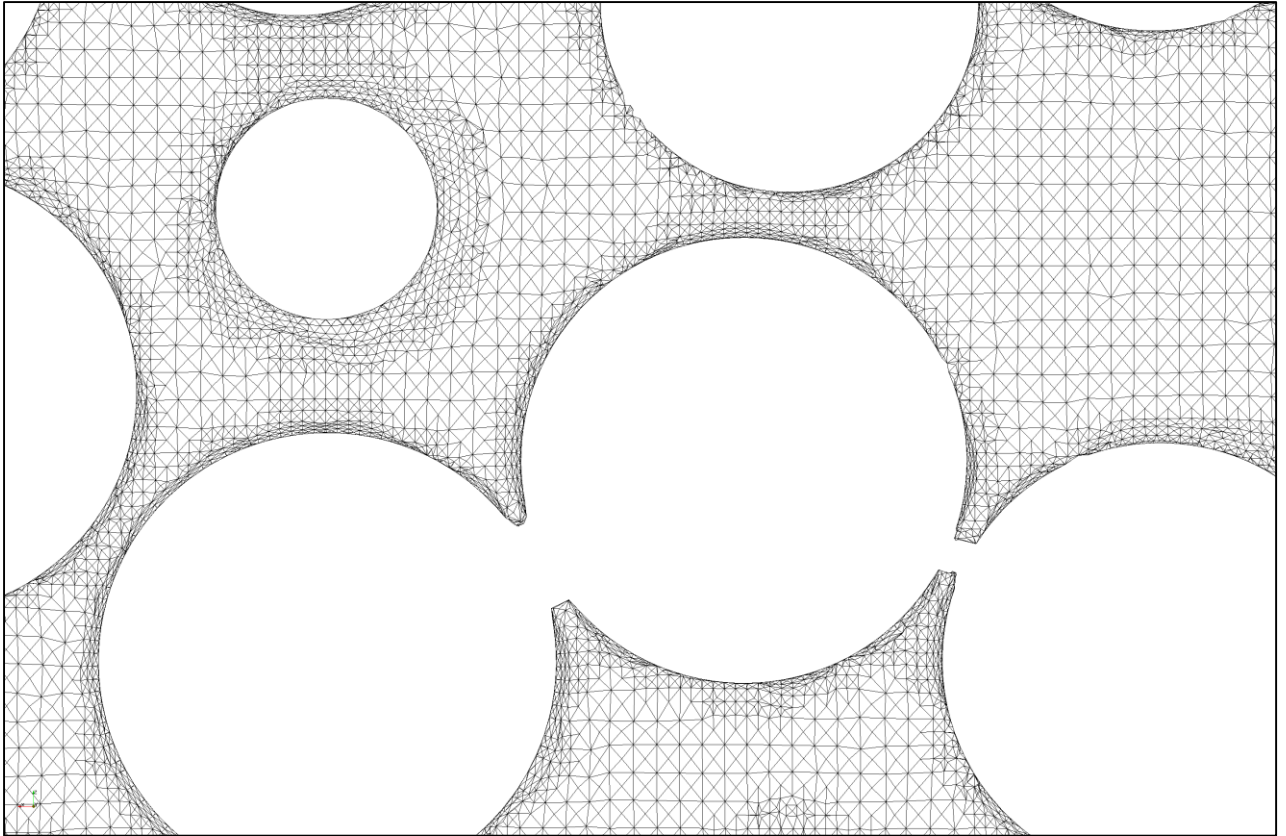
Figure 10-1: segment of a vertical slice through reactor center; ε=0.52 (0% swollen)

It is known from literature [31], [4], and can also be seen in Figure 10-1 that particle contact points are in general a problematic zone for the meshing process, as in these regions a large number of very small and distorted cells would be necessary to fully resolve the gap region, which leads to a large number of bad quality cells. Therefore, different contact point modifications are discussed in literature to overcome this problem. Used methods are shrinking the whole particle or cut off small caps next to the contact zone to increase the gap between two particles so that cells with reasonable geometric quality fit in between. Alternative methods are to produce a particle overlap or creation of cylindrical bridges in proximity to contact points, so that the contact points are converted into larger (and for meshing less problematic) contact regions.

It is reported that the numerical contact bridges with a diameter of about $0.2 \cdot d_p$ reproduce best experimental results regarding pressure drop and heat transfer in packed beds; but also results obtained using particle overlaps are in good agreement with experimental results [31]. As in this work the particles are swelling and hence a particle overlap is produced inevitably, no further effort was taken regarding contact point modification.

```matlab
     %% I) Reactor modelling
     %% I.1))  material data file, to load into specific script
     format compact
     % Input parameters:
 5   Re=1;                          % target particle Reynolds number
     Pr_0 = 4.28;                   % Prandtl number

     % material properties:
     rho_s = 1400 ;                 % density solid  [kg/m³]
10   cp_s = 2567 ;                  % heat capacity  solid [J/kg K]
     lamda_s = 0.17;                % heat conductivity solid [W/m K]


     rho_l =908.6;                  % density liquid [kg/m³]
15   cp_l = 2100;                   % heat capacity liquid [J/kg K]
     mu = 2.55e-4;                  % dynamic viscosity of liquid [kg/m s]
     D_l = 0.68e-9;                 % diffusion coefficient of SA in EtAc at 70°C
     lamda_l=0.125;
     % reaction kinetics:
20   Cin = 0.152  ;                 % [kg A/kg l]  mass fraction of A
     T0 = 348;                      % fluid inlet temperature
     k0_experiment= 1.411e2 ;       % [m³/kgCat s] mass-based reaction velocity constant from Arrhenius-
     Experiments
     Ea = 5318.1    ;               % [K] =Ea/R ... Activation energy
     Hr= 1.695e5;                   % [J/kg A]    mass-based reaction heat
25
     % geometric properties
     d_particle = 6e-4;                         % average particle diameter [m]
     d_sim      = [1         1.1     1.15]; % diameter increase due to swelling
     x_scale    = d_sim(1) ./ d_particle(1);
30   d          = 10* d_particle(1);            % relative tube diameter =10xd_particle
     hbed = 6 ;                                 % maximum bed height (only used for pressure drop
     estimation;
     % 5 sphere diameters+ 2x sphere radius for most upper and lowest sphere position)


     % bed properties
35   epsilon = [0.52,  0.365, 0.286];           % overall bed porosity for 0%, 10% and 15% swelling
     n_particles =374;
     phi_bed = 1-epsilon  ;
     Scat= n_particles*pi .* d_sim .^2     ;             % theoretical initial particle surface
     Seff_LIGGGHTS = [1124   1231.57 1141.35  1114.8] ;
40   eta_surf = [ 0.95   0.847        0.715 ]   ;            % accessibility factor, obtained from LIGGGHTS
     MC-Integration
     Y_p = phi_bed ./ (1-phi_bed) ;                         % relative Catalyst volume per fluid volume

       % input parameters for pressure drop correlations for reactor columns:
       % pos:  epsilon,   L,    dp,       Us,      mu,     rho,      D
45   column1 =[epsilon(3),  0.2,  6e-4,  1.33e-3,  2.55e-4,  908.6,  0.004  ];     % parameters for
     column 1
     column2 =[epsilon(3),  0.04, 6e-4,  3.32e-4,  2.55e-4,  908.6,  0.008  ];     % parameters for
     column 2

     Sc  = mu/(rho_l*D_l);                         % Schmidt number
     a_s = lamda_s/ (rho_s *cp_s);                 % thermal conductivity solid [m²/s]
50   XA_target= 0.9;                               % aimed conversion of A
     dTad= Hr *Cin /cp_l   ;                       % adiabatic temperature rise at full conversion
     T_ad =T0+ dTad ;                              % maximum outlet temperature at full conversion
     gamma= Ea/T0 ;                                % dimensionless activation energy
     k_experiment= k0_experiment* exp(-gamma);     % temperature dependent mass based reaction velocity
     constant @ T0
55
     % Requirements for Pr_sim = Pr_0 :
     % assuming equal liquid heat conductivity (is not a input parameter in
     % OpenFoam!)
     % convergence of MultiRegion solver is sensitive vor values of heat capacity, cp mustn´t be <1000 !
60   % --> so the fluid viscosity shouldn´t be altered!
```

```
     % Adapting velocity for Re_sim = Re_0   (asuming equal density and viscosity):

     Us_0 = Re *mu /(rho_l *d_particle(1))  % superficial fluid velocity [m/s]
 65  Us_sim= Us_0 / x_scale                  % scaled velocity
     A_tube = 10^2*pi/4;                     % crossection area of reactor tube
     Fv0_sim= A_tube*Us_sim;                 % scaled throughput


 70  % Adapting heat conductivity of solid for Bi_sim = Bi_0, assuming
     % constant k_l and equal Re and Pr- numbers, so also constant Nu-Number:
     % Bi_sim = Nu/2*k_l/k_s
     lamda_s_sim =lamda_s * x_scale;

 75  % Adapting diffusion coefficient of species for Pe_m_sim = Pe_m_0
     Pe_m_0= Us_0 *d_particle(1) /D_l;
     DL_sim= Us_sim*d_sim /Pe_m_0;

     % Scaling reaction constant for fluid phase equations  for Da_sim = Da_0
 80
     k0_2= k0_experiment .*rho_s /6 .*d_particle;     % surface based reaction velocity constant,
     unscaled
     k_2= k_experiment .*rho_s /6 .*d_particle ;

     k0 = k0_2 .*Y_p  *6 ./d_particle             % [1/s]  fluid phase reaction velocity constant
 85  k = k_2 .*Y_p  *6 ./d_particle        ;       % [1/s] --> transforms surface based constant to
     fluid phase reaction velocity constant

     k_Tadiabat = k0*exp(-Ea/T_ad)             ;      % reaction velocity constant taking into acount the
     adiabat temperature rise
     acceleration_k_adiabat = (k_Tadiabat-k) /k*100
     % nominal acceleration factor for reaction taking into account maximum
 90  % temperature rise (=complete conversion)


     Da = k .*d_particle ./ Us_0                       % conventional formulation of Damköhler number
     Da_realU= k_2 ./Us_0 .*Y_p *6 .*eta_surf .*epsilon  % adapted formulation of Da for particle bed
 95
     k0_scale = k0 ./ (x_scale^2) ;                    % only intermediata values!
     k_scale = k ./ (x_scale^2) ;                      % only intermediata values!

     k0_2sim = k0_scale ./Y_p /6                       % [m³/m²Cat s] =Input parameter for OpenFOAM
     scalarTransport properties
100  k_2sim = k_scale ./Y_p /6  ;

     k0_corr = k0_scale .* eta_surf                    % [1/s] corrected fluid phase reaction velocity
     constant (loss of surface is considered here)
     k_corr = k_scale .* eta_surf ;
     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105  %% I.2) Reactor Models: PFR model, adiabatic PFR model, calibrated adiabatic PFR model
     %% load material data and upscaling dependencies
     materialData

     i=100 ;              % dimensionless reactor length
110  nmax= 3;             % number of data set entries
     ii=i+1;
     z=[0:1:i];           % maximum reactor length (100*d_particle)

     conversions_PFR = zeros(ii,nmax) ;
115  conversions_APFR = zeros(ii,nmax);
     conversions_mod = zeros(ii,nmax);
     XA_z5 =zeros(3, nmax);

     for n =1:nmax
120
     %% 1) evaluation of simulation results
     %                    0%        10%        15%
     Cout_sim       = [  0.134272  0.1343      0.1356];   % DNS results
```

```matlab
        Tout_sim        = [   349.424   349.395      349.301] ; % DNS results
125
        XA_sim = 1-(Cout_sim ./ Cin)        ;          % simulation result for conversion
        XA_z5(1,:) =XA_sim;

        % check if XA predicted with temperature coincides with XA calculated with C:
130     XA_sim_temp = (Tout_sim-T0) ./ dTad ;

        %% 2) Comparing results for conversion: PFR: Plug flow model
        % reaction heat is not taken into account
        % concentration boundary layer is not taken into account
135
        % Estimation of reactor length needed for aimed conversion X_target= 0.9
        z_target_PFR = d_sim(1) ./Da_realU *log(1/(1-XA_target))  ;

        XA_PFR = 1-exp( -Da_realU(n) .*z ./d_sim(1) )  ;
140     XA_z5(2,n) = XA_PFR(6) ;
        conversions_PFR(:,n) = XA_PFR;

        %%  2)Comparing resultsfor simulated reactor length with adiabatic PFR model
        (APFR)
        [z T_APFR] = ode45(@(z,T) adiabat(z, T, T_ad, Ea, Da_realU(n), T0, d_sim(1), i), z, T0) ;
145
        XA_APFR= (T_APFR-T0) ./ dTad ;
        XA_z5(3,n) = XA_APFR(6)        ;              % Yield for reactor length Z=5, adiabatic model
        conversions_APFR(:,n) = XA_APFR;

150     eta_XA_sim = XA_sim ./ XA_APFR(6)     ;      % overall reactor efficiency compared to adiabatic PFR
        model, for z=5
        T_APFR_z5(n)   = T_APFR(6)              ;              % outlet temperature for reactor length z=5,
        adiabatic model

        %% Adapting the APFR reactor model by incorporating eta_XA_sim to
        %% takeinto account non-ideal reactor performance
155
        [Z T_mod ] = ode45(@(z,T) adiabat_adapted(z, T, T_ad, Ea, Da_realU(n), T0, d_sim(1) ,i, eta_XA_sim
        (n)), z, T0) ;
        T_mod_z5 = T_mod(6)    ;                    % outlet temperature for reactor length z=5, adiabatic model
        XA_mod= (T_mod-T0) ./ dTad ;
        XA_z5(4,n) = XA_mod(6)        ;              % Yield for reactor length z=5, adiabatic model
160     conversions_mod(:,n) = XA_mod ;

        end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %% I.3) function: adiabat.m
165
        function [dTdz] = adiabat( z, T, T_ad, Ea, Da_realU, T0, d_sim, i )

        dTdz = (T_ad -T) .* Da_realU ./d_sim .*exp(-Ea .*(1./T- 1./ T0)) ;

170     end
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %% I.4) function: adiabat_adapted.m for incorporating effectivity factor
        function [dTdz] = adiabat( z, T,T_ad, Ea, Da_realU, T0, d_sim, i,  eta_XA_sim)

175     dTdz = (T_ad -T) .*Da_realU ./d_sim .*exp(-Ea .*(1./T- 1./ T0)) .*eta_XA_sim ;

        end


        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
180
        %% II) evaluation of transient conjugate heat transfer on single particle and comparison with model
        provided by Crank
        %% II.1) Computing the Nu-Number and heat transfer rate for each timestep for transient heat
        transfer

        %% parameters to change for analytic solution:
```

```
185   Re_target = 1 ;
      steps = 100 ;              % length of datafile to read in (necessary for computing difference time
      of last time step,
      steps_end = steps -1;     % cut off one step before end)

      % process parameters of real case
190   rho_s = 1400 ;     % density solid  [kg/m³]
      cp_s = 2567 ;       % heat capacity  solid [J/kg K]
      lambda_s = 0.17;          % heat conductivity solid [W/m K]
      a_s = lambda_s/ (rho_s *cp_s)  ;% thermal conductivity solid [m²/s]

195   rho_l =830;            % density liquid [kg/m³]
      cp_l = 2100;            % heat capacity liquid [J/kg K]
      Pr = 4.28;             % Pr-number
      D_l = 0.68*10^-9   ;
      mu = 2.55e-4;              % dynamic viscosity of liquid [kg/m s]
200
      T_0_sphere = 400 ;   % start temperature of sphere [K]
      T_inlet = 300;       % inlet temperature of fluid [K]
      deltaT_max = T_0_sphere-T_inlet;    % maximum of temperature difference

205   d_particle = 0.6e-3 ;    % average particle diameter [m]
      A_sphere= d_particle ^2*pi ;    % sphere surface
      Us = Re_target*mu/(rho_l* d_particle)              % superficial fluid velocity of HPLC-Column 1 [m/
      s]

      Re = rho_l*Us*d_particle/mu   % Re-number
210   Sc = mu/(rho_l*D_l) ;         % Schmidt number
      lambda_l = mu*cp_l/Pr;        % heat conductivity liquid
      a_l = lambda_l/ (rho_l *cp_l) ; % thermal conductivity solid [m²/s]
      zeta = lambda_l/lambda_s;

215   % Nu-Number Correlation from Book: Bubbles, Drops and Particles (Clift,
      % Grace, Weber, 1978, Chapter 5)
      % Valid for single sphere at 1 <= Re <= 400
                          % 0.25 <= Pr <= 100
      Nu_CGW = (1+(1/Re*Pr)) ^(1/3) *Re^0.41 *Pr^(1/3) +1
220
      % Ranz-Marshall Correlation
      % Valid for single sphere at  10 < Re < 10e5,
                          % 0.6 < Pr < 380
      Nu_Ranz_Marshall = 2    + 0.6* Re^(1/2) *Pr^1/3
225
      % Feng-Michaelides Correlation
      % Valid for single sphere at  1 < Re < 10e5,
                          % 0.6 < Pr < 380
      Nu_corr =Nu_Ranz_Marshall ;
230   alpha_corr = Nu_corr *lambda_l /d_particle;
      Bi= (alpha_corr *d_particle) / lambda_s

      %% load data from postprocessing files (cellSource and FaceSource data)

235   filename1= '/home/diplo/OpenFOAM/diplo-2.2.2/run/049_chtMultiRegion_Re10_developedT_1/
      postProcessing/bedOfSpheres/T_sphere_integral/0/cellSource.dat';
      filename2= '/home/diplo/OpenFOAM/diplo-2.2.2/run/049_chtMultiRegion_Re10_developedT_1/
      postProcessing/bedOfSpheres/T_sphere_mean/0/cellSource.dat';
      filename3= '/home/diplo/OpenFOAM/diplo-2.2.2/run/049_chtMultiRegion_Re10_developedT_1/
      postProcessing/bedOfSpheres/T_surface_mean/0/faceSource.dat';

      [a,data1] = hdrload(filename1);
240   [a,data2] = hdrload(filename2);
      [a,data3] = hdrload(filename3);


      time = data1(1:steps ,1);
245
      volIntegral_T = data1(1:steps,3);      % volume-integrated temperature of sphere
```

```matlab
      volMean_T = data2(1:steps ,3);           % volume- averaged temperature of sphere
      surfaceMean_T = data3(1:steps,3);        % average temperature on sphere surface
                                      % flow averaged fluid temperature at
      outlet
250   sum_alpha_mean = 0;
      sum_Fo = 0;

      for i = 1:(length(volMean_T)-1)

255       t_step(i)  = time(i+1) - time(i) ;

          deltaHeat_sphere(i) = volIntegral_T(i+1)- volIntegral_T(i) ;

          qdot(i) = rho_s*cp_s*deltaHeat_sphere(i)/(t_step(i)*A_sphere) ;
260
          deltaT_volume(i) = volMean_T(i) -  T_inlet ;
          deltaT_surface(i) = surfaceMean_T(i) -  T_inlet  ;

          alpha_volume(i) = - qdot(i) /deltaT_volume(i) ;
265       Nu_volume = alpha_volume*d_particle/lambda_l;

          alpha_surface(i) = - qdot(i) /deltaT_surface(i) ;
          Nu_surface = alpha_surface*d_particle/lambda_l;

270       Fo(i) = time(i)*a_s/ ((d_particle/2)^2);

          sum_alpha_mean = alpha_surface(i)*t_step(i) + sum_alpha_mean;

          heatTransferCranck(Fo,Bi,T_0_sphere, T_inlet, d_particle )
275
           i=i+1;
      end

      alpha_mean_t = sum_alpha_mean/ sum(t_step)
280   Nu_mean= alpha_mean_t*d_particle/lambda_l

      relativeError_corr = (Nu_mean-Nu_corr)/Nu_corr*100;
      NuSurfMax=max(Nu_surface);
      Numax=max([Nu_CGW; Nu_corr; NuSurfMax]);
285
      Nu_vector= Nu_corr.*ones(1,steps_end);

       Fo(steps)= Fo(steps_end)+1e-20;
       heatTransferCranck(Fo,Bi,T_0_sphere, T_inlet, d_particle )
290
       load('Cranck.mat', 'coolingSphere')

      Theta_vol = (volMean_T-T_inlet) / deltaT_max;
      Theta_surf = (surfaceMean_T-T_inlet) / deltaT_max;
295   Theta_surfCranck = (coolingSphere.T_surfCranck-T_inlet) / deltaT_max;

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %% II.2) %% This is the analytical solution for cumputing the 1-D temperature distribution
      %% due to surface evaporation on a single sphere
300   %% after Crank - Marethematics of diffusion page 96 eqn. 6.40
      %% not suitable for Biot-numbers <10e-3

      function [coolingSphere] =heatTransferCranck(Fo,Bi,T_0_sphere,T_inlet, d_particle)

305   %% numeric parameters
      nmax = 3;                      % maximum number of periodic zero values of root function beta_n
      numb_grid_points = 10;          % radial grid resolution inside sphere

      %Simulation parameters:
310   sum_h = 0;                       % sum varialble
      sum_m = 0;                       % sum variable for total transport
      n = 1;                           % running variable for sum
```

```matlab
      r =  1E-12;                          % start point inside sphere for temperature function computing
315   r_particle =d_particle/2;

      %Domain parameters
      XMIN = 1E-10;                        % minimum distance from pole, necessary for steady solution

320   dx=(r_particle-r)/(numb_grid_points-1);   %increment between grid points

      %% Analytical solution of surface evaporation %%

      results = zeros(nmax,1); % preallocating the vectorspace of result vector
325
      fun = @(beta_n) beta_n*cot(beta_n)+Bi-1; % root function of beta_n

          while r <=(r_particle+XMIN)                      %Loop over radial distance

330         %fid=fopen(surface_evaporation_solution,'w');

                          while n <= nmax
                      % This function computes the zero values of the function:
                      % beta_n *cot(beta_n)+Bi-1=0 ,
335                   % needed for the analytical solution of surface evaporation on a sphere
                      % (n-1)*pi+0.1 is the start value for searching the roots (function is pi-periodic)

                          results = fsolve(fun,(n-1)*pi+0.1);     %finding the roots of function "fun"

340                          h= sin(results*r/r_particle) ...
                               /( sin(results) ) ...
                             *  exp(-Fo*results*results) ...
                               /( results*results+Bi*(Bi-1) );

345                          sum_h = h + sum_h   ;

                             m = (6*Bi^2 * exp(-Fo*results*results))/ ...   % total sum of transported scalar
                                 (results^2*(results^2+Bi*(Bi-1))) ;

350                          sum_m = m + sum_m  ; %Perform summation for analytical solution
                             n=n+1;
                             end

              T_rad = (   (T_0_sphere - T_inlet) *2*Bi*r_particle/r*sum_h  )...
355                   +T_inlet;

              m_total = 1-sum_m   ;

              radius = r/r_particle; %relative distance from sphere center [0-1]
360
              r=r+dx;           %increasing increment of radial position
              h = 0;
              sum_h = 0; % resetting the simulation parameters for inner loop
              m =0 ;
365           sum_m = 0 ;
              n = 1;
              x0 = 1;

      % PLOT internal temperature profile within sphere
370   %          subplot(1,2,1), plot (radius, T_rad, 'o');
      %          xlabel ('relative distance from sphere center');
      %          ylabel ('Temperature [K]');
      %          axis ([0 1 T_inlet T_0_sphere]);
      %          hold on ;
375
          end
        T_surf= T_rad;
        coolingSphere = struct;
        coolingSphere.T_surfCranck = T_surf;
```

```matlab
380     coolingSphere.m_total = m_total;
      save Cranck coolingSphere ;

      end

385   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %% III Pressure Drop
      %% III.1) %% Simulation results  compared to Eisfeld and Ergun correlation

      materialData        % load material data script
390
      epsilon_max = 0.54;                    % lowest and highest porosity range in simulated cases
      epsilon_min = 0.2;
      L         = 6* d_particle(1);       % bed height for pressure drop correlations (L =L_bed+1*dp)
      L_bed     = 5* d_particle(1);       % number of sphere layers
395   i_max     = 50;

      rho_sim1 = 1;                          % only needed for Re=10 simulation
      mu_sim1 = 1 ;                          % only needed for Re=10 simulation
      Us_sim1 = 1;                           % only needed for Re=10 simulation
400
      %% Re-scaling of Simulation Results for pressure drop %
        % Results for Re=1:
      epsilon_sim = [0.521      0.443      0.365        0.286];
      deltaP_sim_1 = [1825.65   3739.18    8617.25      21111.9] ;
405   deltaP_m_z1 = deltaP_sim_1 /L_bed *x_scale *mu /mu_sim1   *Us_0 /Us_sim1    % *rho_sim1/rho_l
      % scale the simulation results to fit size of experiments

        % Results for Re=0.5:
      epsilon_sim_05_1 = epsilon_sim(1) ;
      deltaP_sim_05_1 = 6.5196e-08;
410   deltaP_m_z05_1 = deltaP_sim_05_1 /L_bed *x_scale^2            % Velocity was scaled down in
      that case

       % Results for Re=5:
      epsilon_sim_5_1 = epsilon_sim(1) ;
      deltaP_sim_5_1 = 6.86699e-07;
415   deltaP_m_z5_1 = deltaP_sim_5_1 /L_bed *x_scale^2             % Velocity was scaled down in that
      case

       epsilon_sim_5_4 = epsilon_sim(4) ;
      deltaP_sim_5_4 = 9.62691e-06;
      deltaP_m_z5_4 = deltaP_sim_5_4 /L_bed *x_scale^2
420
      % Results for Re=10:
       epsilon_sim_10 = epsilon_sim(1) ;
      deltaP_sim_10 = 20531.4;
      deltaP_m_z10 = deltaP_sim_10 /L_bed *x_scale* mu/mu_sim1   *Us_0/Us_sim1         % Velocity, rho_l
      and mu were set to 1 in that case
425
      %% calculating trend lines for relative pressure drop in [pa/m] and compared with simulation
      for Re= 1

        Us= Re.*mu./ (d_particle(1).* rho_l)
430     epsilon = linspace(epsilon_min, epsilon_max, i_max);
        result = pEisfeld(epsilon, L,d_particle(1),Us,mu,rho_l,d);
      end
      clear Eisfeld.mat
      clear pressuredrop
435
      result_z1= pEisfeld(epsilon_sim, L,d_particle,Us,mu,rho_l,d);
      load('Eisfeld.mat', 'pressuredrop')

      dp_Eisfeld_1= pressuredrop.pEisfeld ./L_bed  % relative pressure drop/m
440   dp_Ergun_1= pressuredrop.pErgun ./L_bed
      dp_Carman_1= pressuredrop.pCarman ./L_bed
```

```
          clear Eisfeld.mat
          clear pressuredrop
445
      %% trend lines for Re=0.5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      for Re = 0.5

          Us= Re.*mu./ (d_particle.* rho_l);
450       result = pEisfeld(epsilon, L,d_particle,Us,mu,rho_l,d);
      end

      result_Re05= pEisfeld(epsilon_sim(1), L,d_particle,Us,mu,rho_l,d);
      load('Eisfeld.mat', 'pressuredrop')
455   dp_Eisfeld_05= pressuredrop.pEisfeld ./L_bed  % relative pressure drop/m
       dp_Ergun_05= pressuredrop.pErgun ./L_bed

       clear Eisfeld.mat
       clear pressuredrop
460   %% trend lines for Re=5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      for Re = 5

          Us= Re.*mu./ (d_particle.* rho_l);
              result = pEisfeld(epsilon, L,d_particle,Us,mu,rho_l,d);
465   end
       clear Eisfeld.mat
       clear pressuredrop

      result_z5 = pEisfeld(epsilon_sim, L,d_particle,Us,mu,rho_l,d);
470    load('Eisfeld.mat', 'pressuredrop')
       dp_Eisfeld_5=  pressuredrop.pEisfeld ./L_bed   % relative pressure drop/m
       dp_Ergun_5=    pressuredrop.pErgun ./L_bed

       clear Eisfeld.mat
475   clear pressuredrop
      %% simulation results for Re=10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      for Re = 10

              Us= Re.*mu./ (d_particle.* rho_l);
480           result = pEisfeld(epsilon, L,d_particle,Us,mu,rho_l,d);
      end

       clear Eisfeld.mat
       clear pressuredrop
485
       result_z10= pEisfeld(epsilon_sim, L,d_particle,Us,mu,rho_l,d);
       load('Eisfeld.mat', 'pressuredrop')
       dp_Eisfeld_10= pressuredrop.pEisfeld ./L_bed
       dp_Ergun_10= pressuredrop.pErgun ./L_bed
490
       clear Eisfeld.mat
       clear pressuredrop
       clear Us

495   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %% III.2) pressure drop of columns used in experiments, 15% swollen bed %

       column1 =[0.22, 0.2,  6e-4, 1.33e-3,    3.18E-04, 908.6, 0.004  ];
       column2 =[0.22, 0.04, 6e-4, 3.32e-4,  3.18E-04, 908.6, 0.008  ];
500
       Re_column1= column1(4)*column1(3)*column1(6)/column1(5);
       Re_column2= column2(4)*column2(3)*column2(6)/column2(5);


505    result_column1 = pEisfeld(column1(1),column1(2),column1(3),column1(4),column1(5),column1(6),column1
      (7))
       result_column2 = pEisfeld(column2(1),column2(2),column2(3),column2(4),column2(5),column2(6),column2
      (7))
```

```
        Re= [0.2:0.1:3];

510       Us= Re.*mu./ (d_particle.* rho_l);
          dp_Re_column1 = pEisfeld(column1(1),column1(2),column1(3),Us,column1(5),column1(6),column1(7));
          load ('Eisfeld.mat', 'pressuredrop')


        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %
515     %% III.3) pressureDrop of ractor tubing equipment
        n_knee =5 ;
        n_frittes =2 ;
        %n_180bows = 4;       % not used, because r/di bow > 20 --> no data available
                             % --> bows treated as normal capillarys with L and di
520     % zeta-Values for pipe components, laminar flow (Estimated at 50°C --> Re =70)
        % VDI-Wärmeatlas, Lac6

        %zeta_180bows = 5    ;    % r/di > 10
        zeta_frittes =50 ;        % Assumption
525     zeta_knee = 10 ;          % at Re = 75; for Re=50: zeta_knee=15

        Fv0=1.6666667e-8;         % 1ml/min

        % density EtAc at different temperatures:
530     % from http://www.ddbst.com/en/EED/PCP/DEN_C21.php:

        % mu EtAc at different temperatures:
        % from: http://www.ddbst.com/en/EED/PCP/VIS_C21.php
        % T [K]        mu [Pas]        rho [kg/m³]
535
        % 30°:         4.40E-04         888
        % 50°:         3.23E-04         863.3
        % 60°:         3.18E-04         850.9
        % 75°:         2.55E-04         832.4
540
        % ESTIMATIONS FOR 60°C:
        rho= 850.9 ;
        mu=  3.18E-04
        L= 8*0.245+0.204                % 6* preheating +inlet+outlet+ bow length
545     di_cap= 0.762e-3 ;              % 0.03"
        r_180bow =16.25e-3;

        U_cap = Fv0/(di_cap^2*pi/4)
        Re_cap= U_cap*di_cap*rho/mu
550
        Ptot = rho*(U_cap)^2/2* (L/ di_cap *64/Re_cap+ n_knee*zeta_knee + n_frittes*zeta_frittes)


        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %% IV) LIGGGHTS to SnappyHexMesh:
555     %%Script to read in the particle data from LIGGGHT-dump-files to create
        %% output files which can be used for advanced meshing with SnappyHexMesh

        % IV.1) Read in the particle data from Liggghts dump-file for one timestep:
        %    Input is dump file name with path
560     srcDir = '/home/diplo/LIGGGHTS/run/swellingBed_p0.5_0%/post/'
        cd(srcDir)

        [results] =readdump_all('dump80000.swellingBed');

565     % Output is in the form of a structure with following variables
        % .timestep      --> Vector containing all time steps
        % .Natoms        --> Vector containing number of atoms at each time step
        % .x_bound       --> [t,2] array with xlo,xhi at each time step
        % .y_bound       --> [t,2] array with ylo,yhi at each time step
570     % .z_bound       --> [t,2] array with zlo,zhi at each time step
        % .atom_data     --> 3 dimensional array with data at each time step stored
        %                    as atomdata(:,:,t)
```

```matlab
     %% 2) data manipulation
575  nParticles = results.Natoms;          % number of particles
     scaleVector = ones(nParticles,3);     % here mono-sized particles are used
     %particleID = results.atom_data(:,1);
     position = results.atom_data(:,3:5); % x, y and z position of sphere center
     basisSphereName = 'Sphere_0';         % name of basis sphere
580
      mkdir('/home/diplo/Pichler_MA/Matlab_routines/SphereFiles');
      cd('/home/diplo/Pichler_MA/Matlab_routines/SphereFiles');

     %% 3) write sphere position and ID to files
585  for i= 1:nParticles

     sphereName= ['Sphere' num2str(i) ];

     writeSpherePositionToFile(sphereName,basisSphereName,scaleVector(i,:),...
590                              position(i,:));
     i=i+1;
     end

     disp('writing the include-command file ');
595  include = 'include.dat';
     fid = fopen( include,'w')

     %write text to file
     for  i= 1:nParticles
600
     command = ['#include "SphereFiles/Sphere' int2str(i) '.dat" ' ];
     fprintf(fid,'%s \n',command);
     i=i+1;
     end
605  fclose(fid);


     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     %% IV.2) function readdump_all to read in data from LIGGGHTS dump-files

610  function [varargout] = readdump_all(varargin)
     % Reads all timesteps from a LAMMPS dump file.
     % Input is dump file name with path
     % Output is in the form of a structure with following variables
     % .timestep     --> Vector containing all time steps
615  % .Natoms       --> Vector containing number of atoms at each time step
     % .x_bound      --> [t,2] array with xlo,xhi at each time step
     % .y_bound      --> [t,2] array with ylo,yhi at each time step
     % .z_bound      --> [t,2] array with zlo,zhi at each time step
     % .atom_data    --> 3 dimensional array with data at each time step stored
620  %                   as atomdata(:,:,t)
     % Example
     %       data = readdump_all('dump.LAMMPS');
     %
     % See also readdump_one, scandump
625  %
     %  Author :  Arun K. Subramaniyan
     %            sarunkarthi@gmail.com
     %            http://web.ics.purdue.edu/~asubrama/pages/Research_Main.htm
     %            School of Aeronautics and Astronautics
630  %            Purdue University, West Lafayette, IN - 47907, USA.

     try
         dump = fopen(varargin{1},'r');
     catch
635      error('Dumpfile not found!');
     end

     i=1;
     while feof(dump) == 0
640      id = fgetl(dump);
```

```matlab
                if (strcmpi(id,'ITEM: TIMESTEP'))
                        timestep(i) = str2num(fgetl(dump));
             else
              if (strcmpi(id,'ITEM: NUMBER OF ATOMS'))
645                   Natoms(i) = str2num(fgetl(dump));
              else
               if (strncmpi(id,'ITEM: BOX BOUNDS',15))
                        x_bound(i,:) = str2num(fgetl(dump))
                        y_bound(i,:) = str2num(fgetl(dump))
650                     z_bound(i,:) = str2num(fgetl(dump))
                else
                 if (strncmpi(id,'ITEM: ATOMS',10))
                        for j = 1 : 1: Natoms
                            atom_data(j,:,i) = str2num(fgetl(dump));
655                     end
                        i=i+1;
                 end
                end
               end
660        end
       end
       %----------Outputs-------------
       %OUTPUTS IN SAME VARIABLE STRUCTURE
       varargout{1}.timestep = timestep;
665    varargout{1}.Natoms = Natoms;
       varargout{1}.x_bound = x_bound;
       varargout{1}.y_bound = y_bound;
       varargout{1}.z_bound = z_bound;
       varargout{1}.atom_data = atom_data;
670    %-----------------------------
       fclose(dump);


       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       %% IV.3) function: writeSpherePositionToFile:
675    %% Script that reads in the particle position data from LIGGGHT-dump-files
       %% and creates output files to use for advanced meshing with SnappyHexMesh

       function [ ] = writeSpherePositionToFile( sphereName, ...
                                                 basisSphereName, ...
680                                               scale, ...
                                                 centerPosition ...
                                                 )
       % writes sphere coordinates for use in a "searchableSurfaceCollection"
       % Usage:
685    % writeSpherePositionToFile(
       %              sphereName,...        <-- name of the sphere, MUST be UNIQUE
       %              basisSphereName, ... <-- name of the basis sphere
       %              scale,    ...<-- vector containing the x, y, and z scaling
       %              centerPosition ...   <-- the center of the sphere
690
       %open file
       cd ('/home/diplo/Pichler_MA/Matlab_routines/SphereFiles');
       fileName = [sphereName,'.dat'];
       disp(['writing to file ',fileName]);
695    fid = fopen( fileName,'w')

       %write text to file
       fprintf(fid,'%s \n',sphereName);
       fprintf(fid,'{ \n');
700    fprintf(fid,'   surface %s; scale (%.3f %.3f %.3f); \n',basisSphereName, ...
                       scale(1), scale(2), scale(3));
       fprintf(fid,'   transform \n');
       fprintf(fid,'   { \n');
       fprintf(fid,'     type     cartesian; \n');
705    fprintf(fid,'     origin  (%.5f %.5f %.5f); \n', centerPosition(1), ...
                       centerPosition(2), centerPosition(3));
       fprintf(fid,'     e1      (1 0 0); e3 (0 0 1); \n');
```

```matlab
      fprintf(fid,'  } \n');
      fprintf(fid,'} \n');
710
      %close file
      fclose(fid);
      cd ('/home/diplo/Pichler_MA/Matlab_routines');
      end
715
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %% V) Calculation of Collision contact time for time step size in LIGGGHTS
      function [result] = hookContactTime(collType, dP, rhoP, Y, nu, ep, vChar)
      % *********************************************************************
720   % ***** Graz University of Technology *********
      % copyright: Stefan Radl, 2012
      % Matlab/octave routines to interface with
      %                              LIGGGHTS
      % ***** Graz University of Technology *********
725   % Function to calculate the characteristic contact time for a Hookean
      % spring-dashpot model
      %[result] = hookContactTime(collType, dP, rhoP, Y, nu, ep, vChar)
      % INPUT
      %   collType    ...=0 for particle-particle collisions
730   %               ...=1 for particle-wall collisions
      %   dP          ...particle diameter
      %   dP          ...density
      %   Y           ...Young's modulus
      %   nu          ...Poisson ratio
735   %   ep          ...coefficient of restitution
      %   vChar       ...characteristic velocity

      % *********************************************************************

740   YStar = Y / (1-nu.^2);
      r = dP ./ 2;
      mass = 4./3 .* r.^3 .* pi .* rhoP;
      if(collType==0) %particle-particle collision
        m_Eff = mass./2;
745     r_Eff = r./2;
        YStar = YStar./2;
      elseif(collType==1) %particle-wall collision
        m_Eff = mass;
        r_Eff = r;
750   end

      k_n =  16./15 .* sqrt(r_Eff) .* YStar ...
         .*( 15./16 .*m_Eff.*vChar.^2/sqrt(r_Eff)/YStar ).^0.2;

755   c_n = sqrt( 4.*m_Eff.*k_n ...
                  ./( 1.+(pi/log(ep)).^2 ) ...
               );

      eta_0 = c_n ./ (2.*m_Eff);
760
      t_c   = pi ./ sqrt(k_n/m_Eff-eta_0.^2);

      dt_h = 2.87*( m_Eff.^2/(r_Eff*YStar^2*vChar) )^0.2

765   dt_DEM = t_c / 50

      result = struct;
      result.tContact = t_c;
      result.kN        = k_n;
770   result.cN        = c_n;

      end

      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
775  %% VI) Grid Convergence Index GCI for grid independency study::
     %% It indicates how much the solution would change with a further refinement of the grid.
     %% A small value of GCI indicates that the computation is within the asymptotic range.
     %% http://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html

780  f1 = 2.7931;                      % simulation result  with finest grid
     f2 = 2.785;
     f3 = 2.7755;                      % simulation result  with coarsest grid

     FS = 1.25;                        % Safety factor; FS =1.25 if 3 different grid spacings are used;
     FS= 3 if only two grid spacings
785  r= 1.5 ;                          % refinement ratio

     p = abs(log((f3-f2)/(f2-f1))/log(r))  % order of convergence

     f_h0 = f1 + (f1-f2)/(r^p-1)       % Solution f for zero grid spacing, "completely resolved solution"
790
     relError1 = abs(f1-f2)/f1      % relative Error 1
     relError2 = abs(f2-f3)/f2      % relative Error 2


795  GCI_1= (FS * relError1)/(r^p -1)*100  % Grid Convergence Index

     GCI_2= (FS * relError2)/(r^p -1)*100

     asymptoticRange = GCI_2/(r^p*GCI_1)
```

Batch reactor experiments for Determination of reaction kinetics
performed by G. Lichtenegger

Determination of conversion at 75°C:

t [min]	signal (A 304 nm)	Rest	conversion XA
0	1.9189	1	0
15	1.193	0.622	37.813
30	0.690	0.360	64.030
45	0.390	0.203	79.654
60	0.247	0.129	87.117
75	0.173	0.090	90.974
90	0.140	0.073	92.697

XA at different temperatures
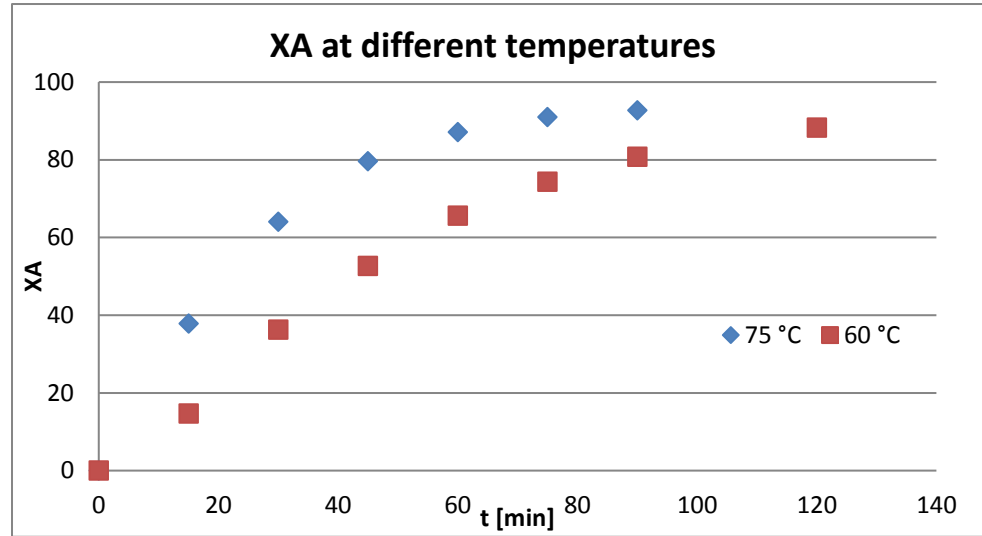
Determination of reaction order:

t [min]	c (SA) [mol/L]	1st order ln(cA0/cA)	2nd order 1/c[A]-1/c[A0]
0	0.855	0.000	0.000
15	0.532	0.475	0.711
30	0.307	1.022	2.083
45	0.174	1.592	4.580
60	0.110	2.049	7.911
75	0.077	2.405	11.792
90	0.062	2.617	14.849
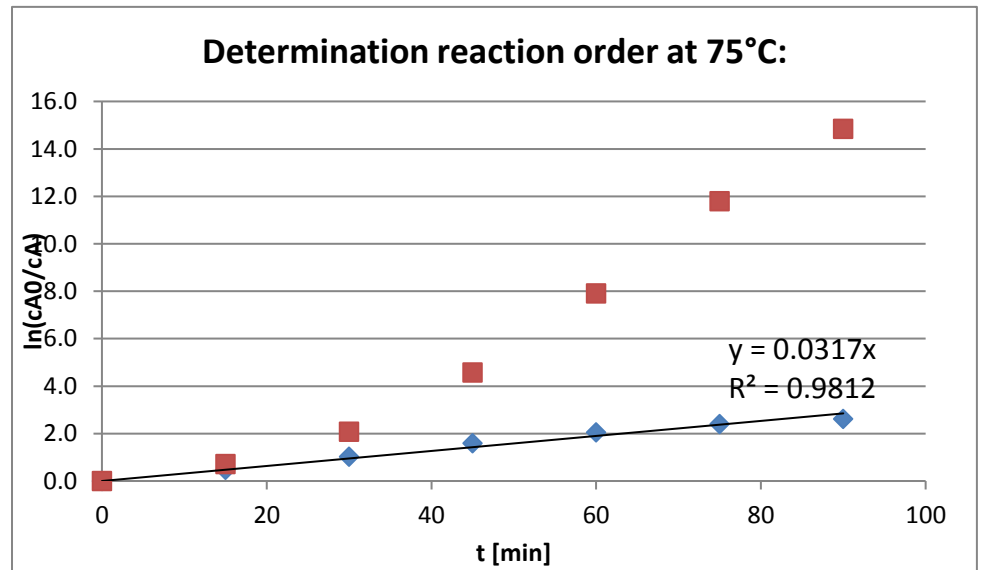
--> **first order reaction in CA**

Determination reaction order at 75°C:

$y = 0.0317x$
$R^2 = 0.9812$

Determination of reaction rate dependency on CA:

mcat [g]	t [min]	dt [min]	X (at 75°C)	dX	C_A [mol]	dC_A [mol]	R_A [mol/(g cat·min)]
0.2997	0		0		0.0145		
	15	15	37.813	37.813	0.0090	0.012	-0.122
n SA0 [mol]	30	15	64.030	26.216	0.0052	0.007	-0.085
0.01453	45	15	79.654	15.625	0.0030	0.004	-0.051
	60	15	87.117	7.463	0.0019	0.002	-0.024
	75	15	90.974	3.857	0.0013	0.002	-0.012
	90	15	92.697	1.723	0.0011	0.001	-0.006

Reaction rate vs. CA

$y = -10.803x$

$R^2 = 0.9806$
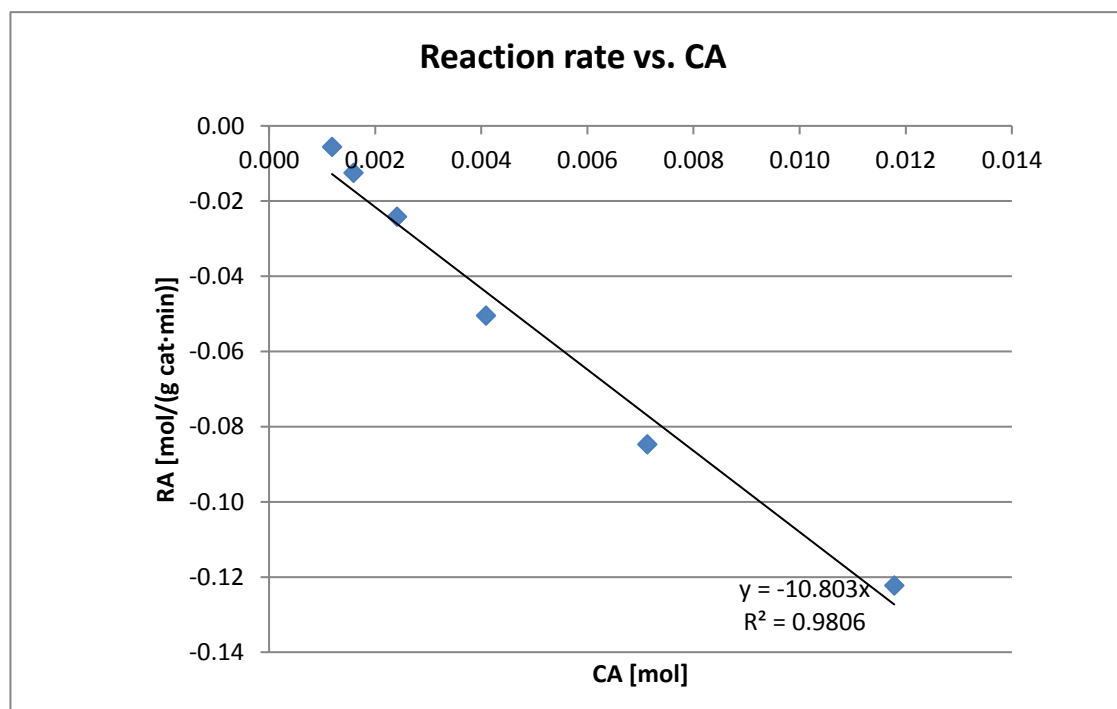
Determination of k' [L/(kg cat·min)] at 75°C:

t [min]	V reactor [L]	m cat [kg]	n S_A [mol]	ln(n S_A)	ln(n S_{A0}/n S_A)	"-(V/W)"	V/W·ln($N_{A,0}$/N_A) [L/kg cat]
0	0.017	0.0003	0.015	-4.231	0.000	-56.723	0.000
15	0.017	0.0003	0.009	-4.707	0.475	-56.723	-26.945
30	0.017	0.0003	0.005	-5.254	1.022	-56.723	-57.998
45	0.017	0.0003	0.003	-5.824	1.592	-56.723	-90.320
60	0.017	0.0003	0.002	-6.281	2.049	-56.723	-116.240
75	0.017	0.0003	0.001	-6.637	2.405	-56.723	-136.423
90	0.017	0.0003	0.001	-6.848	2.617	-56.723	-148.436

k´ [L/(kg cat·min)]	1.953

Determination of k' [L/(kg cat·min)] at 75°C

y = -1.953x
R² = 0.9986
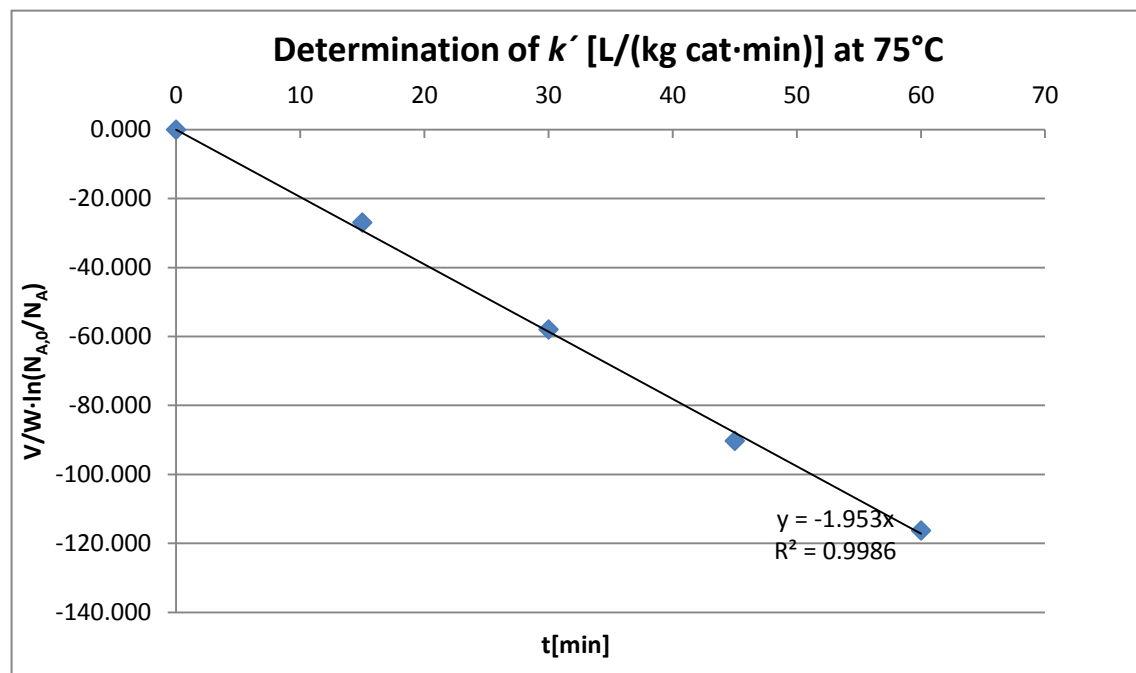
Determination of temerature dependency k (Arrhenius Equation):

T [°C]	k´ [L/(kg cat·s)]	1/T [1/K]	ln(k)
75	0.0317	0.00287	-3.4514
60	0.0178	0.00300	-4.0286
50	0.0096	0.00309	-4.6460

$\ln(k) = - E_a/R \cdot 1/T + \ln(k_0)$

ln (k0)	11.857
k0´[L/(kg cat·s)]	1.41E+05
Ea/R [K]	5318.1

temerature dependency k

1/T [K⁻¹]

y = -5318.1x + 11.857
R² = 0.9871

Cp /(J/(g*K))

Value: 90.0 °C, 6.586 J/(g*K)

Value: 80.0 °C, 3.712 J/(g*K)

Value: 70.0 °C, 2.063 J/(g*K)

Value: 50.0 °C, 1.446 J/(g*K)

Value: 60.0 °C, 1.416 J/(g*K)

[1,3]

Temperature /°C
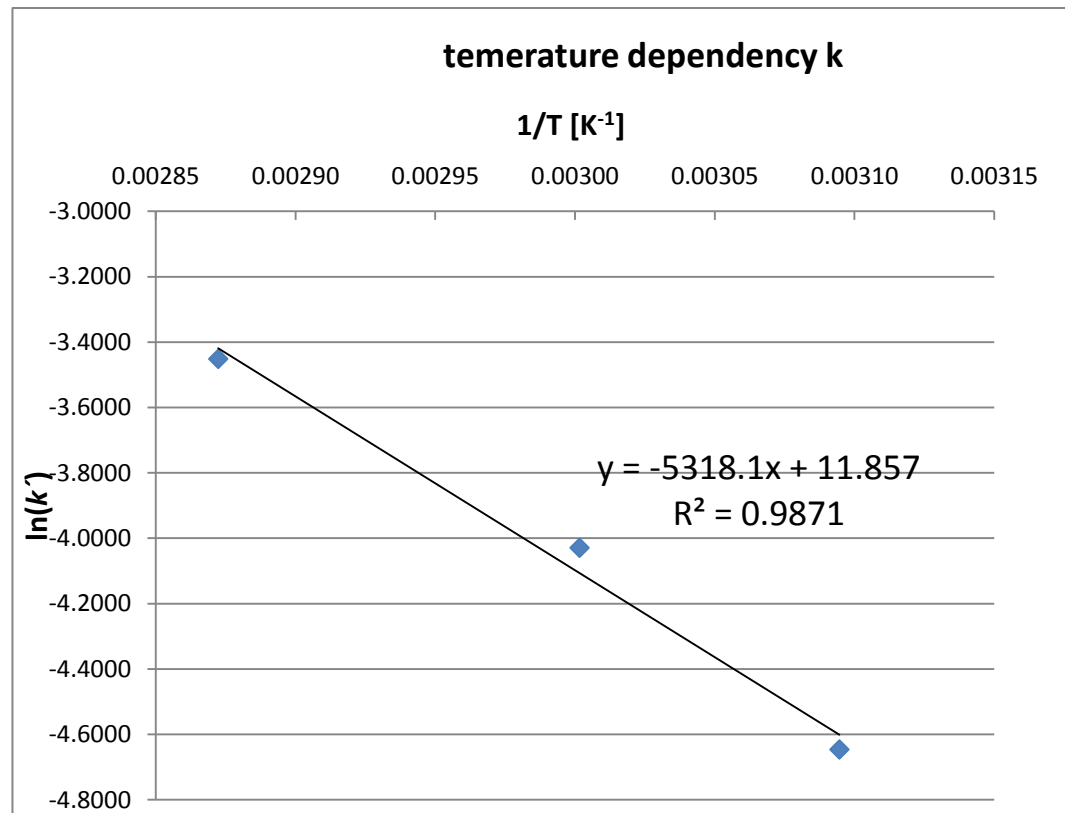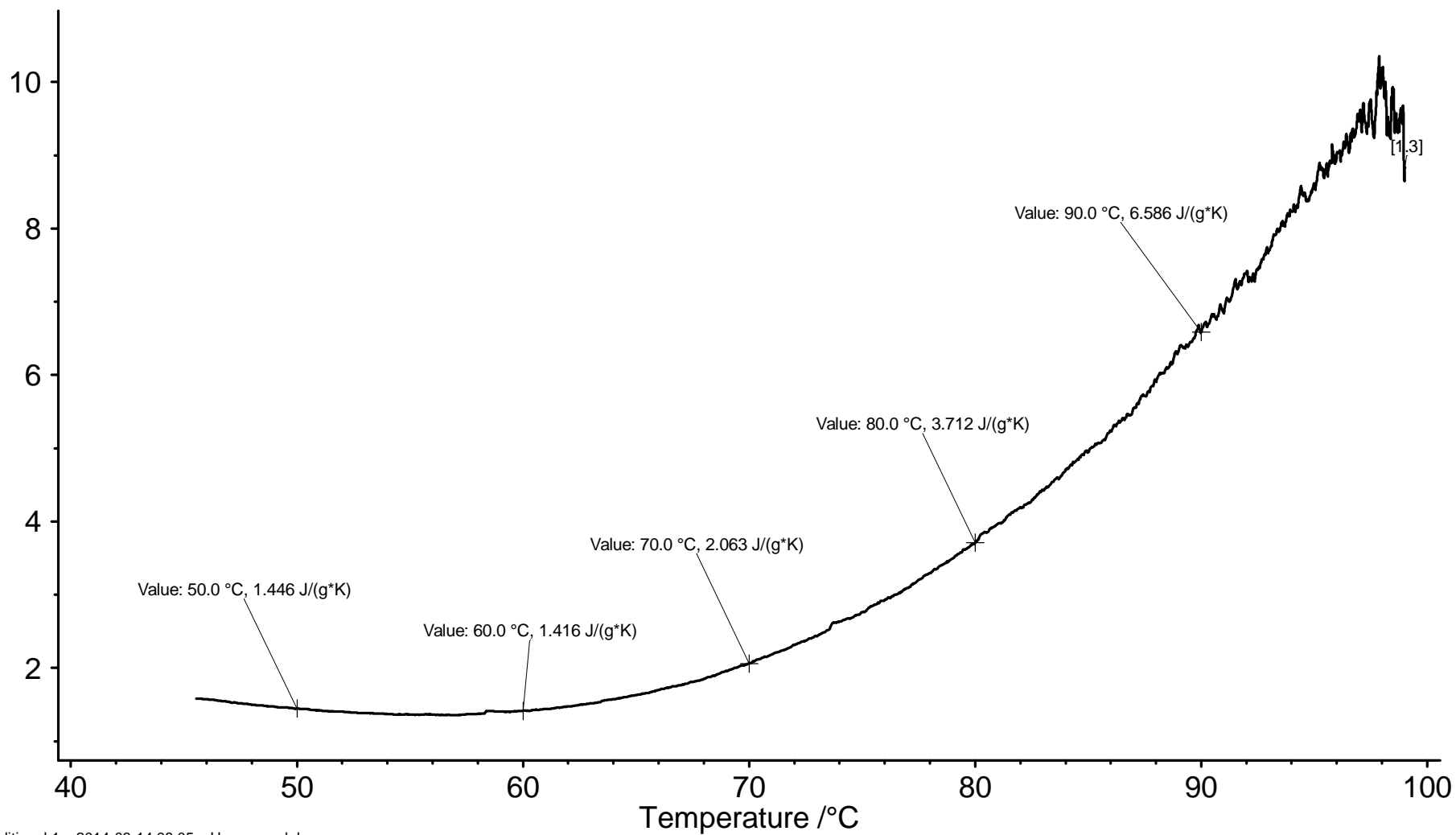
Additional 1   2014-08-14 08:05   User: rcpe-labor

Instrument :	NETZSCH DSC 204F1 240-10-330-K		**File :** C:\NETZSCH\Proteus61\data\DA_IPPT\Amberlite_IR120_630µm_Cp_1.ngb-dd3		
Project :	IPPT	**Reference :**	-----,0 mg	**Mode/type of meas. :**	DSC / sample with correction
Identity :	Amberlite_IR120_630µm_Cp_1	**Material :**	Amberlite	**Segments :**	3/4
Date/time :	13.08.2014 16:27:18	**Corr./temp.cal :**	Blank.ngb-bd3 / DSC204 2K 20ml.ngb-td3	**Crucible :**	Pan Al, pierced lid
Laboratory :	RCPE	**Sens.file :**	DSC204 2K Cp.ngb-ed3	**Atmosphere :**	N2, 20.0ml/min / N2, 20.0ml/min
Operator :	Hainschitz	**Range :**	40°C/3.0(K/min)/100°C	**Corr/m. range :**	01/5000 µV
Sample :	Amberlite_IR120_630µm_Cp_1, 24.55 mg	**Sample car./TC :**	DSC 204F1 t-sensor / E		

Created with NETZSCH Proteus software

Sympatec GmbH
System-Partikel-Technik

QICPIC-Partikelgrößenanalyse
WINDOX 5

QICPIC (QP0112) & RODOS, 0.50 63.0 mm - M6 (5...1705µm) 438 +/- 0
Amberlight   2013-06-11, 14:56:32,748

x_{10} **= 441,81 +/- 0,00 µm** x_{50} **= 602,40 +/- 0,00 µm** x_{90} **= 812,36 +/- 0,00 µm**

x_{16} = 469,15 +/- 0,00 µm x_{84} = 761,19 +/- 0,00 µm x_{99} = 989,63 +/- 0,00 µm

VMD = 714,18 µm SMD = 681,98 µm C_{opt}
= 0,04 +/- 0,00 % [0,00 %]

Konditionen:		Benutzerparameter:	
Produkt:	Amberlight	Benutzer: Piller	
Auswertung:	USER_1 (5.6.0.0)	Probenbezeichnung: Amberlite Acetat	
Trigger:	Standard400Hz_Trocken20..	#Messung: 1,00	
Disp.Meth.:	Standard_Trocken_Granul..	Förderhöhe:	

Verteilungssumme mit Standardabweichung

x_0/µm	Q_0/%	SQ_0/%abs	x_0/µm	Q_0/%	SQ_0/%abs	x_0/µm	Q_0/%	SQ_0/%abs
5,00	0,00	0,00	51,53	0,00	0,00	531,10	29,60	0,00
6,07	0,00	0,00	62,59	0,00	0,00	645,06	62,21	0,00
7,38	0,00	0,00	76,02	0,00	0,00	783,47	88,18	0,00
8,96	0,00	0,00	92,33	0,00	0,00	951,59	98,77	0,00
10,88	0,00	0,00	112,14	0,00	0,00	1155,78	100,00	0,00
13,22	0,00	0,00	136,21	0,00	0,00	1403,78	100,00	0,00
16,05	0,00	0,00	165,43	0,00	0,00	1705,00	100,00	0,00
19,50	0,00	0,00	200,93	0,00	0,00	1826,94	100,00	0,00
23,68	0,00	0,00	244,05	0,00	0,00			
28,76	0,00	0,00	296,41	0,17	0,00			
34,93	0,00	0,00	360,02	0,61	0,00			
42,43	0,00	0,00	437,27	9,00	0,00			

Fraktionen:

x₀/µm	Fraktion dQ₀/%	Fraktion*Partikelanzahl
5,00	0,00 +/- 0,00	0,00 +/- 0,00
6,07	0,00 +/- 0,00	0,00 +/- 0,00
7,38	0,00 +/- 0,00	0,00 +/- 0,00
8,96	0,00 +/- 0,00	0,00 +/- 0,00
10,88	0,00 +/- 0,00	0,00 +/- 0,00
13,22	0,00 +/- 0,00	0,00 +/- 0,00
16,05	0,00 +/- 0,00	0,00 +/- 0,00
19,50	0,00 +/- 0,00	0,00 +/- 0,00
23,68	0,00 +/- 0,00	0,00 +/- 0,00
28,76	0,00 +/- 0,00	0,00 +/- 0,00
34,93	0,00 +/- 0,00	0,00 +/- 0,00
42,43	0,00 +/- 0,00	0,00 +/- 0,00
51,53	0,00 +/- 0,00	0,00 +/- 0,00
62,59	0,00 +/- 0,00	0,00 +/- 0,00
76,02	0,00 +/- 0,00	0,00 +/- 0,00
92,33	0,00 +/- 0,00	0,00 +/- 0,00
112,14	0,00 +/- 0,00	0,00 +/- 0,00
136,21	0,00 +/- 0,00	0,00 +/- 0,00
165,43	0,00 +/- 0,00	0,00 +/- 0,00
200,93	0,00 +/- 0,00	0,00 +/- 0,00
244,05	0,00 +/- 0,00	0,00 +/- 0,00
296,41	0,17 +/- 0,00	0,74 +/- 0,00
360,02	0,45 +/- 0,00	1,95 +/- 0,00
437,27	8,39 +/- 0,00	36,74 +/- 0,00
531,10	20,59 +/- 0,00	90,20 +/- 0,00
645,06	32,61 +/- 0,00	142,84 +/- 0,00
783,47	25,97 +/- 0,00	113,76 +/- 0,00
951,59	10,59 +/- 0,00	46,39 +/- 0,00
1155,78	1,23 +/- 0,00	5,38 +/- 0,00
1403,78	0,00 +/- 0,00	0,00 +/- 0,00
1705,00	0,00 +/- 0,00	0,00 +/- 0,00
1826,94	0,00 +/- 0,00	0,00 +/- 0,00

gemittelt wurde über die folgenden Messungen:

Datum, Uhrzeit und Gerät	Benutzer	Probenbezeichnung
2013-06-11 14:56:32.7480 0222 Q	Piller	Amberlite Acetat

Sympatec GmbH
System-Partikel-Technik

QICPIC-Partikelgrößenanalyse
WINDOX 5

QICPIC (QP0112) & OASISDRY/L, 0.50 63.0 mm - M6 (5...1705µm)
Amberlight Amberlite Acetat   **2013-06-11, 14:56:32,748**

| s_{10} = 0,9299 | s_{50} = 0,9398 | s_{90} = 0,9480 | Berechnungsmodus: USER_1 |
| s_{16} = 0,9315 | s_{84} = 0,9467 | s_{99} = 0,9498 | Klassengrenzen: M6 |
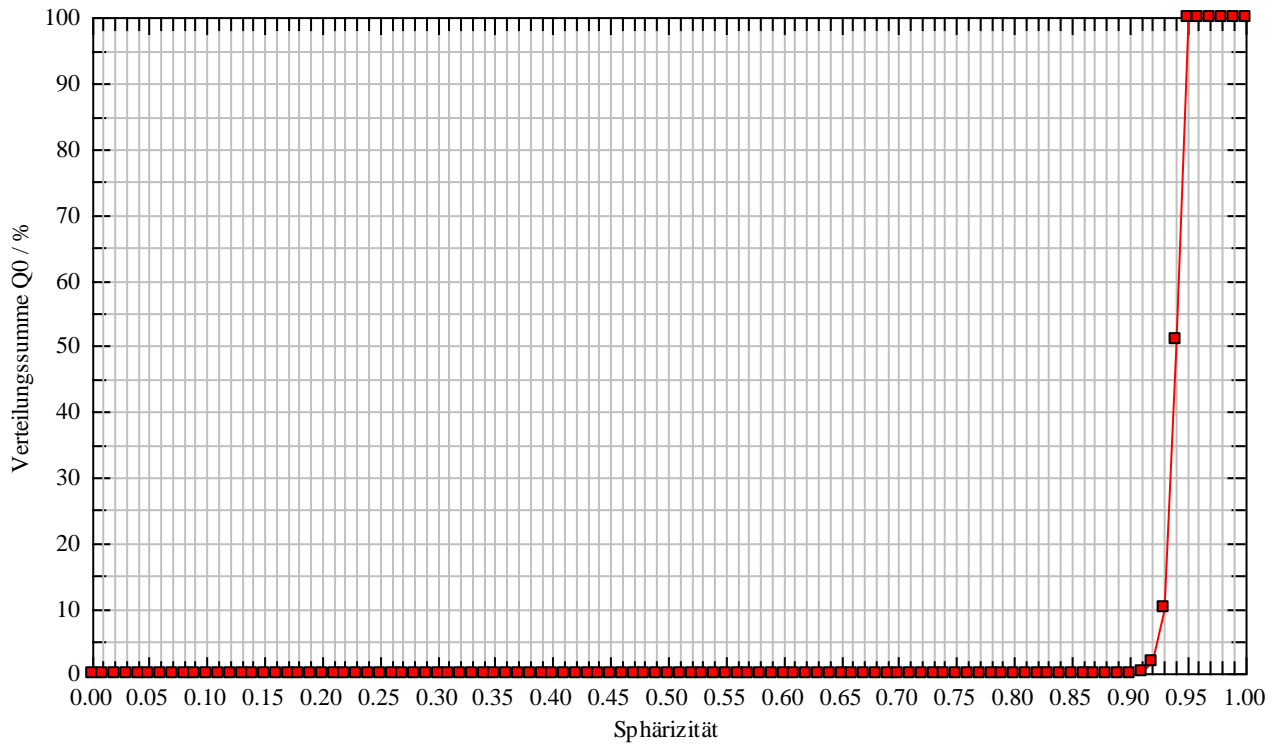
Verteilungssumme

Sphärizität Intervallobergrenze s_o	Verteilungs- summe Q_0 / %	Rückstand (1-Q_0) / %
0,050	0,00	100,00
0,100	0,00	100,00
0,150	0,00	100,00
0,200	0,00	100,00
0,250	0,00	100,00
0,300	0,00	100,00
0,350	0,00	100,00
0,400	0,00	100,00
0,450	0,00	100,00
0,500	0,00	100,00
0,550	0,00	100,00
0,600	0,00	100,00
0,650	0,00	100,00
0,700	0,00	100,00
0,750	0,00	100,00
0,800	0,00	100,00
0,850	0,00	100,00
0,900	0,00	100,00
0,950	100,00	0,00
1,000	100,00	0,00

Verteilungsdichte

Sphärizität Intervallmitte s_m	Verteilungs- dichte q_0
0,045	0,000
0,095	0,000
0,145	0,000
0,195	0,000
0,245	0,000
0,295	0,000
0,345	0,000
0,395	0,000
0,445	0,000
0,495	0,000
0,545	0,000
0,595	0,000
0,645	0,000
0,695	0,000
0,745	0,000
0,795	0,000
0,845	0,000
0,895	0,000
0,945	49,087
0,995	0,000

Molecular Dynamics- Simulated Diffusion coefficient of Salicylic Acid in Ethyl Acetate

Performed by P. Feenstra

T [K]	D [cm²/s]	+/-
298	0.553	0.0045
333	0.62	0.67
343	0.67	0.47
353	0.71	0.01

For our simulations all components were manually sketched using the AMBER11 [1–3] module xleap. The corresponding OPLS-AA (Optimized Potentials for Liquid Simulations – All Atoms) force field [4] parameters (including charge) were applied to all. All MD simulations were performed using the GROMACS software package version 4.5.3.[5–8]

For the calculations of the mean square displacement the analyte molecule was put inside a cubic box with the side length of 3.5 nm followed by solvation with the solvent at ambient density. Periodic boundary conditions were applied and van der Waals interactions were cut off at 1.4 nm. Long-range electrostatics was accounted for using particle mesh Ewald.[9] The box was equilibrated at the given temperatures using a Berendsen thermostat. [42]

The pressure was held constant at 1 bar by a Berendsen barostat with a time constant of 0.5 ps and compressibility of 4.5×10^{-5} bar^{-1}.[10] 5,000 steps of steepest descent minimization were followed by 100,000 steps of equilibration. The equations of motion were solved using an MD integrator with a time step of 2 fs. Production runs for the LIE calculations spanning XXX ns of simulation time were performed for each temperature. The diffusion coefficients were calculated using the gromacs utility g_msd.

1. Pearlman, D. A. *et al.* AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Comp. Phys. Commun.* **91,** 1–41 (1995).

2. Case, D. A. *et al.* The Amber biomolecular simulation programs. *J. Comput. Chem.* **26,** 1668–1688 (2005).

3. Case, D. A. *et al.* AMBER. *Univ. California, San Fr.* **11,** (2010).

4. Jorgensen, W. L. & Tirado-Rives, J. The OPLS [optimized potentials for liquid   simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.* **110,** 1657 (1988).

5. Berendsen, J. C., van der Spoel, D. & van Drunen, R. GROMACS: A message-passing parallel molecular dynamics implementation. *Comp. Phys. Com.* **91,** 43–56 (1995).

6. Lindahl, E., Hess, B. & van der Spoel, D. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *J. Mol. Mod.* **7,** 306–317 (2001).

7. Van der Spoel, D. *et al.* GROMACS: Fast, Flexible, and Free. *J. Comp. Chem.* **26,** 1701–1719 (2005).

8. Hess, B., Kutzner, C., van der Spoel, D. & Lindahl, E. Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation . *J. Chem. Theory Comput.* **4,** 435–447 (2008).

9. Essmann, U. *et al.* A smooth particle mesh Ewald method. *J. Chem. Phys* **103,** 8577–8592 (1995).

10. Berendsen, J. C., Postma, J. P. M., van Gunsteren, W. F., DiNola, A. & Haak, J. R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **81,** 3684–3690 (1984).

```
      ### Creating a dense particle bed with mono-sized, swelling Particles
      ### and computing the particle fraction and accessible surface with Monte Carlo Integration

      ##variables used:
 5    variable          finalZ equal 5.0              # final domain height befor compaction (symmetrical!)
      variable          CylRadius equal 5.0           # radius of cylindrical domain
      variable          phiCompact equal 0.5          # particle volume fraction of experimental tube filling
      variable          phiStart equal 0.2            # limit of volume particle fraction for fast insert/pack
      method
      variable          compactZ equal ${phiStart}/${phiCompact}        # estimated compaction-factor in Z-
      direction
10
      variable          initialZ equal ${finalZ}/${compactZ}            # initial domain height befor
      compaction
      variable          domainVolume equal ${finalZ}*${CylRadius}^2*PI  # final domain volume after
      compaction
      variable          radius0 equal 0.5                               # initial particle radius
      variable          swelling equal 1.15                             # swelling of particle radius in
      solvent-experiment
15    variable          radiusFinal equal ${radius0}*${swelling}        # particle radius after swelling

      variable          youngsModulus equal 5.e6       # E-Modul of particles
      variable          frictionCoefficient equal 0.05  # friction coefficient
      variable          triesPerParticle equal 10000    # number of sample points per particle for
      MCIntegrator
20    variable          seed equal 1000000              # seed for random point generation

      variable          ngrowts equal 100              # number of growing steps
      variable          growts equal 40000             # number of time steps while growing happens
      variable          relaxts equal 20000
25
      ##
      atom_style        granular
      atom_modify       map array
      boundary          f f p              #fixed cylyndrical wall boundaries in x,y, periodic in z
30    newton            off

      communicate       single vel yes
      units             si

35    region            reg block -${CylRadius} ${CylRadius} -${CylRadius} ${CylRadius} 0 ${initialZ} units
      box
      create_box        1 reg
      neighbor          0.05 bin
      neigh_modify      delay 0

40    #Material properties required for new pair styles

      fix               m1 all property/global youngsModulus peratomtype ${youngsModulus}
      fix               m2 all property/global poissonsRatio peratomtype 0.45
      fix               m3 all property/global coefficientRestitution peratomtypepair 1 0.95
45    fix               m4 all property/global coefficientFriction peratomtypepair 1 ${frictionCoefficient}
      #fix              m5 all property/global characteristicVelocity scalar 2.

      #New pair style
      pair_style gran model hertz tangential history #  Hertzian without cohesion
50    pair_coeff        * *

      timestep          0.000025

      #fix              gravi all gravity 9.81 vector 0.0 0.0 -1.0
55
      fix cylwalls all wall/gran model hertz tangential history primitive type 1  zcylinder ${CylRadius}
      0. 0.

      #region of insertion
      region            bc cylinder z 0. 0. ${CylRadius}-${radius0} 0 ${initialZ} units box
```

```
60      ### Particle distributions and insertion

        fix             pts1 all particletemplate/sphere 1 atom_type 1 density constant 1.0 radius
        constant        ${radius0}
        fix             pdd1 all particledistribution/discrete 1.  1 pts1 1.0
65
        fix             ins all insert/pack seed ${seed} distributiontemplate pdd1 vel constant 0.0 0.0 0.0
        &
                        insert_every once overlapcheck yes all_in yes volumefraction_region ${phiStart}
        region bc

        #apply nve integration to all particles
70      fix             integr all nve/sphere

        #insert the first particles so that dump is not empty
        run             1
        dump    dmp all custom 2000 post/dump*.swellingBed id type x y z fx fy fz radius
75
        #run more timesteps to insert large particles
        run             2000 upto
        unfix   ins

80      ### Compressing of the bed ###
        ##fix deform1 all deform 10 z scale ${compactZ} units box
        fix deform1 all deform 10 z final 0.0 ${finalZ}  units box

        run             20000 upto
85      unfix deform1

        ### Swelling of the particles ###
        variable        growevery equal (${growts}/${ngrowts})          # time steps between 2 growing steps
        variable        Rgrowrate equal ((${radiusFinal}/${radius0})^(1/${ngrowts})) # growrate of radius
90      compute         rad all property/atom radius
        variable        dgrown atom ${Rgrowrate}*2.*c_rad

        fix             grow all adapt ${growevery} atom diameter v_dgrown
        #run
95      run             ${growts}
        #let the packing relax
        unfix   grow
        run             ${relaxts}

100     ### Monte Carlo Integration:
        ### Computing the global accessible particle surface after swelling (Surface Integration)

        variable        numberParticles equal count(all)                        # number of particles in
        domain
        variable        nTries  equal ${numberParticles}*${triesPerParticle}    # total number of tries for
        MCIntegrator
105
        compute         MCIntS all monteCarloIntegral ntry ${nTries} axis 0 0 1 origin 0 0 0
        radiusHeightCyl ${CylRadius} ${finalZ} seed ${seed} radiusInner 0.0  doSurfaceIntegration

        ## Computing the local particle fraction in annular regions (Volume Integration)

110     label       loopa
        variable    a loop 50
        print       "A= $a"
        variable outerRadius equal 0.1*${a}
        variable deltaRadius equal 0.1
115     variable innerRadius equal ${outerRadius}-${deltaRadius}
        variable RadiusMean equal (${outerRadius}+${innerRadius})/2


        #Run multiple times
120     compute         MCIntV${a} all monteCarloIntegral ntry ${nTries} axis 0 0 1 origin 0 0 0
```

```
      radiusHeightCyl ${outerRadius} ${finalZ} seed ${seed} radiusInner ${innerRadius}

      thermo_style    custom step atoms ke c_MCIntS c_MCIntV${a}
      thermo                  1

125   run 0

      ### variables and prints for postprocessing ###

      variable        numberParticles equal count(all) # number of particles in domain
130   variable        particleFraction0 equal ((${numberParticles}*${radius0}^3*PI*4/3)/${domainVolume})
      variable        particleFraction equal ((${numberParticles}*${radiusFinal}^3*PI*4/3)/${domainVolume}
      )
      variable        MCIntS equal c_MCIntS       # MonteCarlo-computed accessible particle surface
      variable        MCIntV equal c_MCIntV${a}        # MonteCarlo-computed particle volume fraction

135   print           "The number of particles is ${numberParticles}"
      #print          "The particleFraction before swelling is approx. ${particleFraction0}"
      #print          "The final particleFraction is approx. ${particleFraction}"
      print           "MC-computed particle fraction is ${MCIntV}"
      print           "accessible particle surface is ${MCIntS}"
140
      uncompute MCIntV${a} #delte the compute
      next       a
      jump       in.swellingComplete loopa
```

```
       - CONTROL DICT

      FoamFile
      {
  5       version     2.0;
          format      ascii;
          class       dictionary;
          location    "system";
          object      controlDict;
 10   }
      // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

      //Main Switches for solver
      solveFluidFlow;
 15   solveEnergyEqn;

      //time step so save slice and bed surface
      deltaTSlice     5000;

 20   application     buoyantPimpleFoamReact;

      startFrom       latestTime;

      //startTime       0;
 25
      stopAt          endTime;

      endTime         2e8;

 30   deltaT          1e-2;

      writeControl    timeStep;

      writeInterval   $deltaTSlice;
 35
      purgeWrite      3;

      writeFormat     ascii;

 40   writePrecision  6;

      writeCompression compressed;

      timeFormat      general;
 45
      timePrecision   6;

      runTimeModifiable true;

 50   maxCo           0.30;

      adjustTimeStep  yes;

      maxDi           5.0;
 55
      // *********************************************************************** //
      //Calculating reaction and scalar transport
      functions
      (
 60

      pressureDrop
      {
          type            faceSource;
 65       functionObjectLibs ("libfieldFunctionObjects.so");
          log             true;
          valueOutput     false;
```

```
            outputControl    timeStep;
            outputInterval   5000;
 70         surfaceFormat    null;
            source           sampledSurface;
            sampledSurfaceDict
                {

 75             type patchInternalField;
                patches ( ".*inlet.*" );
                offsetMode normal;
                distance 0.0;
                interpolate false;
 80             }
            operation        areaAverage;
            fields
            (
                p
 85         );
        }


        heatFluxOut
        {
 90         type             faceSource;
            functionObjectLibs ("libfieldFunctionObjects.so");
            log              true;
            valueOutput      false;
            outputControl    timeStep;
 95         outputInterval   5000;
            surfaceFormat    null;
            source           patch;
            sourceName       outlet;
            operation        weightedAverage;
100         weightField      phi; //surface flux
            fields
            (
                T C
            );
105     }


        C_Out
        {
            type             faceSource;
110         functionObjectLibs ("libfieldFunctionObjects.so");
            log              true;
            valueOutput      false;
            outputControl    timeStep;
            outputInterval   5000;
115         surfaceFormat    null;
            source           patch;
            sourceName       outlet;
            operation        areaAverage;
            fields
120         (
                T C U
            );
        }


125 Rreact_integral
        {

            type                     cellSource;
            functionObjectLibs ("libfieldFunctionObjects.so");
130         outputControl    timeStep;
            outputInterval   5000;
            log                      yes;
            valueOutput              false;
            source                   all;
```

```
135              operation          volIntegrate;
                 surfaceFormat        raw;
            fields
            (
                rReact
140         );
    }

    surfaceConcentrationAverage
    {
145     type                faceSource;
        functionObjectLibs ("libfieldFunctionObjects.so");

        log                 true;
        valueOutput         false;
150
        outputControl       timeStep;
        outputInterval      5000;

        surfaceFormat       null;
155     source              patch;
        sourceName          bed;
        operation           areaAverage ;

        fields
160     (
            T C
        );
    }

165 surfaceConcentrationMin
    {
        type                faceSource;
        functionObjectLibs ("libfieldFunctionObjects.so");
        log                 true;
170     valueOutput         false;
        outputControl       timeStep;
        outputInterval      5000;
        surfaceFormat       null;
        source              patch;
175     sourceName          bed;
        operation           min ;

        fields
        (
180         T C
        );
    }

    surfaceConcentrationMax
185 {
        type                faceSource;
        functionObjectLibs ("libfieldFunctionObjects.so");
        log                 true;
        valueOutput         false;
190     outputControl       timeStep;
        outputInterval      5000;
        surfaceFormat       null;
        source              patch;
        sourceName          bed;
195     operation           max ;
        fields
        (
            T C
        );
200 }
```

```
        sliceData0
        {
                type surfaces;
205             functionObjectLibs ("libsampling.so");
                interpolationScheme cell;
                outputControl    timeStep;
                outputInterval  $deltaTSlice;
                surfaceFormat vtk;
210             fields ( U p T C );
                surfaces
                (
                    slice_parallel_0
                    {
215                     type cuttingPlane;
                        planeType pointAndNormal;
                        pointAndNormalDict
                        {
                            basePoint (0 0 0);
220                         normalVector (0 1 0);
                        }
                     interpolate true;
                    }
                );
225     }

        bedSurface
        {
            type surfaces;
230         functionObjectLibs ("libsampling.so");
            interpolationScheme cell;
            outputControl    timeStep;
            outputInterval  $deltaTSlice;
            surfaceFormat vtk;
235         fields ( T C );
            surfaces
            (
                slice_parallel_1
                {
240                 type patch;
                    patches (bed);
                }
            );
        }
245
        );
```

```
      /*--------------------------------*- C++ -*----------------------------------*\
      | =========                 |                                                 |
      | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
      |  \\    /   O peration     | Version:  2.2.1                                 |
 5    |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
      |    \\/     M anipulation  |                                                 |
      \*---------------------------------------------------------------------------*/
      FoamFile
      {
10        version     2.0;
          format      ascii;
          class       dictionary;
          object      snappyHexMeshDict;
      }
15    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

      // Which of the steps to run
      castellatedMesh         true;
      snap                    true;
20    addLayers               true;

      radius1          0.575;
      cylRadius1       5.000;

25    refineLevelGap  2;
      refineLevelSurf 3;

      // Geometry. Definition of all surfaces. All surfaces are of class
      // searchableSurface.
30    // Surfaces are used
      // - to specify refinement for any mesh cell intersecting it
      // - to specify refinement for any mesh cell inside/outside/near
      // - to 'snap' the mesh boundary to the surface
      geometry
35    {

          //definition of base spheres
          Sphere_0 { type searchableSphere; centre (0 0 1e-99); radius $radius1; } //center MUST NOT be
      exactly at (0 0 0)!

40    // definition of particle bed

          bedOfSpheres  {  type searchableSurfaceCollection;  mergeSubRegions true;

          //Type 1 spheres, include all include files
45
      #   include "include.dat"

                      } //end bedOfSpheres

50    };

      // Settings for the castellatedMesh generation.
      castellatedMeshControls
      {
55
          // Refinement parameters
          // ~~~~~~~~~~~~~~~~~~~~~~

          // If local number of cells is >= maxLocalCells on any processor
60        // switches from from refinement followed by balancing
          // (current method) to (weighted) balancing before refinement.
          maxLocalCells 90000000;

          // Overall cell limit (approximately). Refinement will stop immediately
65        // upon reaching this number so a refinement level might not complete.
          // Note that this is the number of cells before removing the part which
```

```
          // is not 'visible' from the keepPoint. The final number of cells might
          // actually be a lot less.
          maxGlobalCells 90000000;
 70
          // The surface refinement loop might spend lots of iterations refining just a
          // few cells. This setting will cause refinement to stop if <= minimumRefine
          // are selected for refinement. Note: it will at least do one iteration
          // (unless the number of cells to refine is 0)
 75       minRefinementCells 100;

          // Number of buffer layers between different levels.
          // 1 means normal 2:1 refinement restriction, larger means slower
          // refinement.
 80       nCellsBetweenLevels 2;


          // Explicit feature edge refinement
          // ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 85
          // Specifies a level for any cell intersected by its edges.
          // This is a featureEdgeMesh, read from constant/triSurface for now.
          features
          (
 90       );


          // Surface based refinement
          // ~~~~~~~~~~~~~~~~~~~~~~~~~
 95
          // Specifies two levels for every surface. The first is the minimum level,
          // every cell intersecting a surface gets refined up to the minimum level.
          // The second level is the maximum level. Cells that 'see' multiple
          // intersections where the intersections make an
100       // angle > resolveFeatureAngle get refined up to the maximum level.

          refinementSurfaces
          {
105           bedOfSpheres
              {
                  // Surface-wise min and max refinement level
                  level ($refineLevelSurf $refineLevelSurf);

110               // Optional specification of patch type (default is wall).
                  // No constraint types (cyclic, symmetry) etc. are allowed.

                  regions
                  {
115               }

                  // Optional specification of patch type (default is wall). No
                  // constraint types (cyclic, symmetry) etc. are allowed.
                  patchInfo
120               {
                      type        wall;
                      inGroups    (wall);
                  }
              }
125       }

          // Resolve sharp angles on fridges
          resolveFeatureAngle 70;
130
          // Region-wise refinement
          // ~~~~~~~~~~~~~~~~~~~~~~~
```

```
135        // Specifies refinement level for cells in relation to a surface. One of
           // three modes
           // - distance. 'levels' specifies per distance to the surface the
           //   wanted refinement level. The distances need to be specified in
           //   descending order.
140        // - inside. 'levels' is only one entry and only the level is used. All
           //   cells inside the surface get refined up to the level. The surface
           //   needs to be closed for this to be possible.
           // - outside. Same but cells outside.

145        refinementRegions
           {

           }

150        // Mesh selection
           // ~~~~~~~~~~~~~~~

           // After refinement patches get added for all refinementSurfaces and
           // all cells intersecting the surfaces get put into these patches. The
155        // section reachable from the locationInMesh is kept.
           // NOTE: This point should never be on a face, always inside a cell, even
           // after refinement.
           locationInMesh (0.001 0.001 5.999); //CHOOSE CAREFULLY!

160
           // Whether any faceZones (as specified in the refinementSurfaces)
           // are only on the boundary of corresponding cellZones or also allow
           // free-standing zone faces. Not used if there are no faceZones.
           allowFreeStandingZoneFaces true;
165    }


    // Settings for the snapping.
    snapControls
170    {
           //- Number of patch smoothing iterations before finding correspondence
           //  to surface
           nSmoothPatch 80;

175        //- Relative distance for points to be attracted by surface feature point
           //  or edge. True distance is this factor times local
           //  maximum edge length.
           tolerance 3.0;

180        //- Number of mesh displacement relaxation iterations.
           nSolveIter 80;

           //- Maximum number of snapping relaxation iterations. Should stop
           //  before upon reaching a correct mesh.
185        nRelaxIter 80;


           // Feature snapping

190            //- Number of feature edge snapping iterations.
               //  Leave out altogether to disable.
               nFeatureSnapIter 10;

               //- Detect (geometric) features by sampling the surface (default=false)
195            implicitFeatureSnap false;

               //- Use castellatedMeshControls::features (default = true)
               explicitFeatureSnap false;
       }
200
```

```
          // Settings for the layer addition.
          addLayersControls
205       {
              // Are the thickness parameters below relative to the undistorted
              // size of the refined cell outside layer (true) or absolute sizes (false).
              relativeSizes true;

210           // Per final patch (so not geometry!) the layer information
              layers
              {
                  "bedOfSpheres.*"
                  {
215                   nSurfaceLayers 3;
                  }
              }

              // Expansion factor for layer mesh
220           expansionRatio 1.0;

              // Wanted thickness of final added cell layer. If multiple layers
              // is the thickness of the layer furthest away from the wall.
              // Relative to undistorted size of cell outside layer.
225           // is the thickness of the layer furthest away from the wall.
              // See relativeSizes parameter.
              finalLayerThickness 0.65;

              // Minimum thickness of cell layer. If for any reason layer
230           // cannot be above minThickness do not add layer.
              // Relative to undistorted size of cell outside layer.
              // See relativeSizes parameter.
              minThickness 0.07;

235           // If points get not extruded do nGrow layers of connected faces that are
              // also not grown. This helps convergence of the layer addition process
              // close to features.
              // Note: changed(corrected) w.r.t 17x! (didn't do anything in 17x)
              nGrow 1;
240

              // Advanced settings

              // When not to extrude surface. 0 is flat surface, 90 is when two faces
245           // are perpendicular
              featureAngle 70;

              // Maximum number of snapping relaxation iterations. Should stop
              // before upon reaching a correct mesh.
250           nRelaxIter 60;

              // Number of smoothing iterations of surface normals
              nSmoothSurfaceNormals 5;

255           // Number of smoothing iterations of interior mesh movement direction
              nSmoothNormals 3;

              // Smooth layer thickness over surface patches
              nSmoothThickness 10;
260
              // Stop layer growth on highly warped cells
              maxFaceThicknessRatio 0.5;

              // Reduce layer growth where ratio thickness to medial
265           // distance is large
              maxThicknessToMedialRatio 0.3;
```

```
            // Angle used to pick up medial axis points
            // Note: changed(corrected) w.r.t 16x! 90 degrees corresponds to 130 in 16x.
270         minMedianAxisAngle 90;

            // Create buffer region for new layer terminations
            nBufferCellsNoExtrude 0;

275
            // Overall max number of layer addition iterations. The mesher will exit
            // if it reaches this number of iterations; possibly with an illegal
            // mesh.
            nLayerIter 120;
280     }



        // Generic mesh quality settings. At any undoable phase these determine
285     // where to undo.
        meshQualityControls
        {
            #include "meshQualityDict"

290         // Advanced

            //- Number of error distribution iterations
            nSmoothScale 6;
            //- amount to scale back displacement at error points
295         errorReduction 0.75;
        }


        // Advanced
300
        // Flags for optional output
        // 0 : only write final meshes
        // 1 : write intermediate meshes
        // 2 : write volScalarField with cellLevel for postprocessing
305     // 4 : write current intersections as .obj files
        debug 0;


        // Merge tolerance. Is fraction of overall bounding box of initial mesh.
310     // Note: the write tolerance needs to be higher than this.
        mergeTolerance 1e-6;


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

```
      - FV SCHEMES

    ddtSchemes
    {
 5      default backward; //Euler;
    }

    gradSchemes
    {
10      default           Gauss linear;
    }

    divSchemes
    {
15      default           none;
        div(phi,U)        Gauss limitedLinearV 1.0; //upwind;
        div(phi,C)        Gauss limitedLinear 1.0;
        div(phi,K)        Gauss limitedLinear 1.0;
        div(phi,h)        Gauss limitedLinear 1.0; //upwind;
20      div(phi,k)        Gauss limitedLinear 1.0;
        div(phi,epsilon) Gauss limitedLinear 1.0;
        div(phi,R)        Gauss  limitedLinear 1.0;
        div(R)            Gauss linear;
        div((muEff*dev2(T(grad(U))))) Gauss linear;
25  }

    laplacianSchemes
    {
        default       Gauss linear limited corrected 0.333;
30      laplacian(muEff,U) Gauss linear limited corrected 0.333;
        laplacian(Dp,p_rgh) Gauss linear limited corrected 0.333;
        laplacian(alphaEff,h) Gauss linear limited corrected 0.333;
        laplacian(alphaEff,e) Gauss linear limited corrected 0.333;
        laplacian(DkEff,k) Gauss linear limited corrected 0.333;
35      laplacian(DepsilonEff,epsilon) Gauss linear limited corrected 0.333;
        laplacian(DREff,R) Gauss linear limited corrected 0.333;
    }

    interpolationSchemes
40  {
        default           linear;
    }

    snGradSchemes
45  {
        default           limited corrected 0.333;
    }

    fluxRequired
50  {
        default           no;
        p_rgh;
    }
    // ********************************************************************* //
```

```
      - FV SOLUTION

     solvers
     {
 5       "(rho|rhoFinal)"
         {
             solver          PCG
             preconditioner  DIC;
             tolerance       1e-7;
10           relTol          0;
         }

         p_rgh
         {
15           solver          GAMG;
             tolerance       1e-7;
             relTol          0.01;
             smoother        GaussSeidel;

20           cacheAgglomeration true;
             nCellsInCoarsestLevel 10;
             agglomerator    faceAreaPair;
             mergeLevels     1;
             maxIter         100;
25       }

         p_rghFinal
         {
             $p_rgh;
30           tolerance       1e-7;
             relTol          0;
             maxIter         200;
         }

35       "(U|e|k|epsilon|R|C)"
         {
             solver          PBiCG;
             preconditioner  DILU;
             tolerance       1e-9;
40           relTol          0.01;
         }

         "(U|e|k|epsilon|R|C)Final"
         {
45           $U;
             tolerance       1e-11;
             relTol          0;
         }

50       h
         {
             $U;
         }

55       hFinal
         {
             $h;
             tolerance       1e-11;
             relTol          0;
60       }
     }

     PIMPLE
     {
65       momentumPredictor   on;
         nOuterCorrectors    2;
         nCorrectors         2;
```

```
         nNonOrthogonalCorrectors 1;
     }
70
     relaxationFactors
     {
         fields
         {
75       }
         equations
         {
             "h.*"            0.4;
             "U.*"            0.4;
80       }
     }
     // ************************************************************************* //
```

```
     - THERMOPHYSICAL PROPERTIES

     thermoType
     {
 5       type              heRhoThermo;
         mixture           pureMixture;
         transport         const;
         thermo            hConst;
         equationOfState   rhoConst;
10       specie            specie;
         energy            sensibleEnthalpy;
     }

     mixture
15   {
         specie
         {
             nMoles          1;
             molWeight       1;
20       }
         thermodynamics
         {
             Cp              2100;
             Hf              0.0;
25       }
         transport
         {
             mu              2.55e-4;
             Pr              4.28;
30       }

     equationOfState
         {
             rho     908.6;
35       }
     }

     // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

40   -MASS TRANSPORT PROPERTIES

     //Mass Transport Parameters
     DT              DT [ 0 2 -1 0 0 0 0 ] 0.68e-9;

45   //Reaction Parameters
     CboundMax            0.152; //bound max concentration by this value
     nCorr                1;
     relaxCoeff           1;
     nameReactivePatch "bed";
50   k_0                  0.0119;
     E_a               5318.1;
     nu_11                -1;

     deltaHReact     deltaHReact [ 0 2 -2 0 0 0 0 ] 1.695e5;     // [J/kg]
```