



TU Dortmund University
Department of Biochemical and Chemical Engineering
Laboratory of Equipment Design

Technical Manual - Filter Belt Crystallizer Model

Maximilian Rainer Kattner

Supervisor: M.Sc. Stefan Höving

Contents

1 Modeling	3
1.1 DWSIM	3
1.2 Crystallization Model	4
1.2.1 Growth Kernel	5
1.2.2 Agglomeration Kernel	7
1.2.3 Nucleation Kernel	8
1.2.4 Solution Mass Balance and Supersaturation	9
1.2.5 Model Inputs and Outputs	10
1.2.6 Parameter Determination	11
1.2.1 Temperature Profiles	18
1.3 Solid-Liquid Separation Model	20
1.3.1 Filtration Model	20
1.3.1.1 Porosity Calculation	20
1.3.1.2 Model Equations and Assumptions	21
1.3.1.3 Parameter Determination	22
1.3.2 Deliquoring Model	23
1.3.3 Filter Cake Washing Model	26
1.3.4 Overall Model Inputs and Outputs	31
1.4 Drying Model	32
1.4.1 Model Equations	33
1.4.1.1 Nusselt Correlations	36
1.4.1.2 Discretization	39
1.4.2 Efficiency Function	40
1.4.3 Overall Model Inputs and Outputs	40
1.5 System Architecture and Graphical User Interface	41
Bibliography	45

Appendix	49
1.6 Supplementary Substance Data	49
1.7 Supplementary Information: Modeling	49
1.7.1 Drying	49
1.7.1.1 Derivation of PDE used in the drying model	49
1.8 Supplementary Information: GUI	53
1.9 Supplementary Information: Validation of DWSIM Database	59

1

Modeling

The aim of this thesis is to set up a framework for the modeling of the filter belt crystallizer [1]. Therefore, for each unit operation of the apparatus a suitable model has been set-up. The following section discusses the models for crystallization, filtration, deliquoring, filter cake washing and convective drying and introduces the respective model equations. Furthermore, an introduction into the software DWSIM which provides the framework for the integration of models is given.

1.1 DWSIM

DWSIM is an open-source process simulation software released in 2005. It is compliant with the CAPE-OPEN framework which was established in an effort to provide a “plug-and-play” standardization process towards more interoperable process simulation software and has since been adapted - at least in parts - by various process simulation tools [2, 3]. In contrast to most established process simulators, DWSIM offers extensive customizability as well as extensive support for the programming language *Python*. As such, DWSIM provides an extensive Python application programming interface (API_{pr}) which offers the possibility to access both DWSIM simulations and simulation results from external Python scripts as well as allowing to create Python scripts for custom unit operations [4, 5].

DWSIM thereby allows for two interpreters for integrating Python unit operations. On the one hand, it is possible to link an external Python installation up to version 3.9. While this allows for the integration of external packages such as *NumPy* [6] or *SciPy* [7], computational speed of the external installation is reduced compared to the second possibility which is given in the integrated IronPython installation (equivalent to a CPython 2.7 installation). While it is not possible to utilize external packages, the IronPython installation is able to access .NET equivalents of classes and functions. Since the IronPython interpreter offers

faster computing times and suffices further requirements such as access to differential equation solvers and Microsoft Office applications via the .NET framework, it has been utilized in this work. [4]

DWSIM can be applied in a large number of engineering tasks due to the multitude of available thermodynamic models such as the Peng-Robinson equations of state, the Soave-Redlich-Kwong equation of state or activity coefficient models such as UNIFAC and UNIQUAC [8]. Last but not least, DWSIM also offers a substance database which can easily be accessed in custom Python scripts using CAPE-OPEN compliant functions [9].

1.2 Crystallization Model

The crystallization model utilized in the simulation of the filter belt crystallizer relies on the population balancing approach proposed by *Randolph* and *Larson* [10] with the general form being given in Equation 1.1.

$$\frac{\partial n}{\partial t} + \frac{\partial(Gn)}{\partial L} + n \frac{\partial V}{V \partial t} + D(L) - B(L) + \sum_k \frac{\dot{V}_i n_i}{V} = 0 \quad (1.1)$$

Here, G denotes the crystal growth rate, $B(L)$ and $D(L)$ denote birth and death of crystals by nucleation, agglomeration and attrition and n denotes the number density of crystals for batch crystallizers which is defined as shown in Equation 1.2 [11].

$$n_i = \frac{N_i}{V \cdot \Delta L_i} \quad (1.2)$$

Given the assumptions that the crystallizer is ideally mixed and thus no spatial gradients occur, a constant volume inside the crystallizer with no ingoing or outgoing streams and size independent crystal growth this equation simplifies as shown in Equation 1.3 [11, 12].

$$\frac{\partial n}{\partial t} = -G \frac{\partial n}{\partial L} - D(L) + B(L) \quad (1.3)$$

Utilizing the high resolution semi-discrete finite volume scheme by [13, 14] results in the discretized population balancing equation (PBE) shown in Equation 1.4.

$$\frac{dn_i}{dt} = \frac{1}{\Delta L_i} \left[\left(\frac{dn_i}{dt} \right)_{\text{nucleation}} + \left(\frac{dn_i}{dt} \right)_{\text{growth}} + \left(\frac{dn_i}{dt} \right)_{\text{agglomeration}} + \left(\frac{dn_i}{dt} \right)_{\text{breakage}} \right] \quad (1.4)$$

The number density function which is described by the terms denoted in the above equation is geometrically discretized into a total of 30 subdomains. Here, each crystal size L_i is calculated as $L_{i+1} = \sqrt[5]{2} \cdot L_i$. This classification enables a wide range of particle sizes

while allowing a reasonable resolution in the relevant size range. While literature suggests a spacing according to $L_{i+1} = \sqrt[3]{2} \cdot L_i$, as it doubles the crystal volume with each size increments which is advantageous especially in agglomeration modeling [15], this work has deliberately not employed the spacing rule as agglomeration has been shown to play a subordinate role in the given system.

The input distribution for the crystallization model is determined from experimental data and fitted to a logarithmic normal distribution as shown in Equation 1.5 using the GRG-Nonlinear solver in Microsoft Excel and adjusted manually to capture the general shape of the CSD.

$$Q_3(x) = \frac{1}{2} \cdot \operatorname{erfc} \left[-\frac{\ln \left(\frac{x}{x_{50,3}} \right)}{\sqrt{2}\sigma_{\text{LND}}} \right] \quad (1.5)$$

For the given case, the breakage term has been neglected in the calculation since previous work observed no significant influence of stirring rate on the size distribution of particles. Similarly, homogeneous nucleation which - together with secondary nucleation - makes up the term $\left(\frac{dn_i}{dt} \right)_{\text{nucleation}}$, has been neglected due to the tendency of sucrose solutions to remain stable even at high supersaturations [16].

In contrast, secondary nucleation is incorporated into the overall model. Past theses have established that the crystallization container is not ideally mixed due to its non-ideal geometry with particle deposition towards the bottom of the vessel and dead zones in the corners of the container. Thus, the secondary nucleation kinetic is supposed to mitigate effects of non-idealities occurring in the vessel.

The overall model which is made up of the remaining terms is set up as shown in Figure 1.1 and integrated as a custom unit operation using the IronPython installation in DWSIM where it is solved using a Runge-Kutta algorithm of 4th order contained in the *DotNumerics.ode* class [5, 17].

1.2.1 Growth Kernel

The integration of a growth kernel is required to calculate the crystal growth rate G . This rate serves as input for the discretized change in number density due to crystal growth $\left(\frac{dn_i}{dt} \right)_{\text{growth}}$ which is then integrated into the overall discretized population balancing equation shown in Equation 1.4. In order to adequately describe crystal growth, the discretization scheme developed by Koren [13] and adapted for crystallization by Qamar *et al.* [14] is employed. As can be seen in Equation 1.6 the change in number density in each crystal size class consists of an input term, $G_{i,-}$ which describes the growth of crystals from size

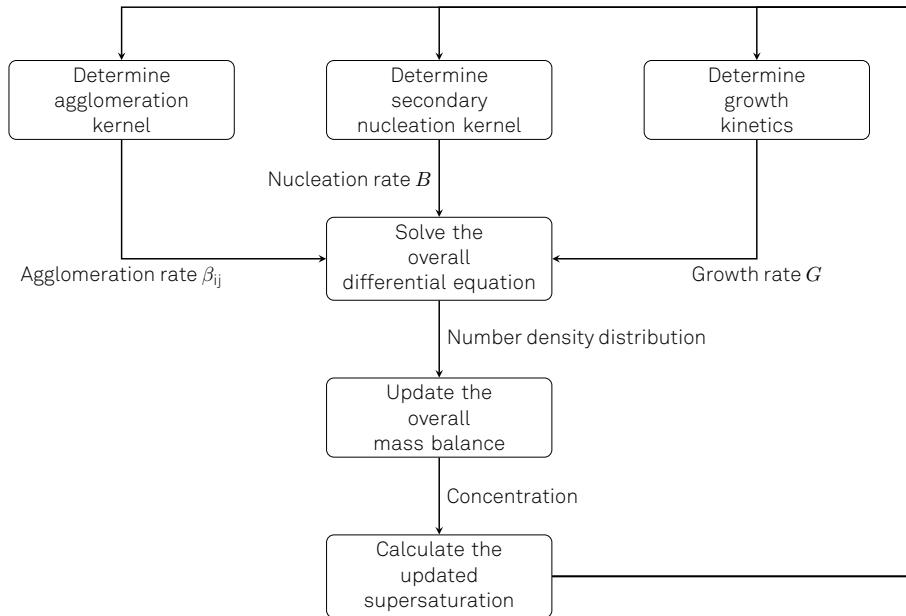


Figure 1.1: Schematic flowchart of the crystallization model. Adapted from [18]

class $i - 1$ to size class i and an output rate, $G_{i,+}$ which describes the opposed growth from size class i to size class $i + 1$.

$$\left(\frac{dn}{dt} \right)_{i,\text{growth}} = G_{-,i} - G_{i,+} \quad (1.6)$$

$$= \begin{cases} G[0 - 0.5(n_2 + n_1)] & i = 1 \\ G[0.5(n_2 + n_1) - (n_2 + 0.5\Upsilon_2(n_2 - n_1))] & i = 2 \\ G[(n_{i-1} + 0.5\Upsilon_{i-1}(n_{i-1} - n_{i-2})) - (n_i + 0.5\Upsilon_i(n_i - n_{i-1}))] & i = 3, \dots, N-1 \\ G[(n_{N-1} + 0.5\Upsilon_{N-1}(n_{N-1} - n_{N-2})) - (n_N + 0.5(n_N - n_{N-1}))] & i = N \end{cases}$$

In order to limit results from this discretization scheme to physically reasonable solutions, a flux limiter function Υ as shown in Equation 1.7 is introduced. By limiting convection through class boundaries, it additionally avoids oscillations and negative crystal size densities.

$$\Upsilon_i(\hat{r}_i) = \max \left(0, \min \left(2\hat{r}_i, \min \left(\frac{1}{3} + \frac{2}{3}\hat{r}_i, 2 \right) \right) \right) \quad (1.7)$$

Additionally, Υ is dependent on the upwind ratio of consecutive solution gradients \hat{r}_i given in Equation 1.8. Here, an additional numerical parameter $\epsilon = 10^{-10}$ is introduced in order to guarantee the avoidance of divisions by zero in uniform flux regions [13].

$$\hat{r}_i = \frac{n(L_{i+1}) - n(L_i) + \epsilon}{n(L_i) - n(L_{i-1}) + \epsilon} \quad (1.8)$$

For the description of the growth rate G itself, two different approaches have been integrated into the overall crystallization model. The first approach is given in an exponential growth rate approach which - in its most basic form shown in Equation 1.9 - represents the most widespread approach towards modeling crystal growth behavior containing only a rate constant k_g , the absolute supersaturation S and an exponential constant α_G . [19]

$$G = k_G \cdot S^{\alpha_G} \quad (1.9)$$

Here, an extended version of the general exponential growth rate based on an *Arrhenius* approach as can be seen in Equation 1.10 is utilized. Unlike the general form, an additional parameter as well as a temperature dependence of the overall growth rate is incorporated. [20, 21]

$$G = g_1 \cdot \exp\left(-\frac{g_2}{RT}\right) \cdot (S - 1)^{g_3} \quad (1.10)$$

The second approach on the other hand is given in the BCF-approach which is reliant on two model parameters A_{BCF} and B_{BCF} .

1.2.2 Agglomeration Kernel

Agglomeration is one of the most important secondary kinetics in all of crystallization. Therefore, a description of agglomeration is especially relevant in the modeling of crystallization processes. Agglomeration modeling typically relies on an agglomeration kernel, $\beta_{i,j}$, which is then combined with a function describing the occurrence of agglomeration based on particle size. The agglomeration kernel itself is a measure of frequency of collision and aggregation of particles from size classes i and j and highly dependent on the substance systems and the operating conditions during the crystallization. [22]

Depending on the primary mean of agglomeration in a process, various ways for the description of an agglomeration kernel exist. However, in most cases, a simple approach assuming the agglomeration to be constant $\beta_{i,j} = \beta_0$ provides a valid way in the theoretical description of agglomeration. [22]

In this work, a size dependent approach for the description of agglomeration according to *Faria et al.* has been chosen. As can be seen in Equation 1.11, this kernel incorporates two adjustable model constants. These include the parameter β_{ag} and the critical length L_{crit} . The critical length is based on theory which implies a minimum and maximum size for the

occurrence of agglomeration. Accordingly, a third significant length exists which describes the crystal size at which the highest probability of agglomeration exists. [23]

$$\beta_{\text{ag,j}} = \beta_0 \cdot \frac{(L_{\text{crit}} \cdot L_i \cdot L_j)^2}{\left(\frac{1}{2} \cdot L_{\text{crit}}^3 + L_j^3\right) \cdot \left(\frac{1}{2} \cdot L_{\text{crit}}^3 + L_j^3\right)} \cdot G \cdot \frac{w_s \cdot \rho_{\text{susp}}}{\phi \cdot \rho_{\text{cryst}}} \quad (1.11)$$

According to the agglomeration model, the agglomeration term in Equation 1.4 is further divided into birth and death terms. The overall birth term comprises two terms which describe the birth of particles due to the collision and congregation of particles of size class i with particles from other classes on the one hand, and, on the other hand with particles from size class i . Similarly, the death terms, which describe the flux of particles leaving size class i is made up of a term describing congregation with smaller classes and a second term describing the congregation with particles from larger classes. Combined together, this yields the expression shown in Equation 1.12.

$$\begin{aligned} \left(\frac{dn_i}{dt} \right)_{\text{agglomeration}} &= n_{i-1} \cdot \sum_{j=1}^{i-2} X_{ijk} \beta_{i-1,j} n_j + \underbrace{\frac{1}{2} \beta_{i-1,i-1} n_{i-1}^2}_{\text{birth with same class}} \\ &\quad \underbrace{\sum_{j=1}^{i-2} X_{ijk} \beta_{i-1,j} n_j}_{\text{birth with different classes}} \quad \text{birth terms} \\ &- \underbrace{n_{i-1} \sum_{j=1}^{i-1} 2^{j-1} \beta_{i,j} n_j}_{\text{death with smaller classes}} - \underbrace{n_i \cdot \sum_{j=i}^N \beta_{i,j} n_j}_{\text{death with larger classes}} \quad \text{death terms} \end{aligned} \quad (1.12)$$

Additionally, the equation contains an auxiliary variable X_{ijk} which controls the birth of agglomerates according to Equation 1.13.

$$X_{ijk} = 2^{i-1} \left(2 + \sum_{k=0}^{i-1} 2^k + \sum_{k=0}^{j-2} 2^k \right) \quad (1.13)$$

1.2.3 Nucleation Kernel

As already established, primary nucleation is neglected in the overall crystallization model. Therefore, the theoretical modeling will not be introduced further hereafter.

In order to compensate secondary nucleation, a corresponding approach is incorporated into the crystallization model. In literature various calculation approaches for modeling nucleation exist. However, since secondary nucleation does not simply consist of one unified kinetic but much rather comprises several different kinetics, a deterministic physical model is practically nonexistent [24]. Thus, exponential rate approaches dominate especially the mechanism of secondary nucleation which can include mechanisms like attrition,

fluid shear or surface breeding. Empirical relationships for secondary nucleation generally take the form shown in Equation 1.14 and comprise an empirical rate constant k_{birth} , the impeller speed in the crystallization vessel ω_i , the suspension density M_T as well as the degree of supersaturation $S - 1$. [24–26]

$$B_{\text{sec}} = k_{\text{birth}} \omega_i^{k_{\text{birth},2}} M_T^{k_{\text{birth},3}} (S - 1)^{k_{\text{birth},3}} \quad (1.14)$$

It is important to note that the above model is not all-encompassing in literature and various different approaches to the empirical nucleation approach exist. Specifically, similar approaches to the exponential growth rate incorporate Arrhenius-like approaches [24, 26]. Additionally, it is possible to modify the model based on the type of secondary nucleation which is assumed to be dominating. Since the nucleation rate in the developed, underlying model is supposed to especially compensate poorly mixed zones in the crystallization vessel, the approach given in Equation 1.15 has been chosen which neglects impeller speed in the crystallization vessel [27]. This variation also reduces the number of fitting parameters to three which reduces the computational expenditure required for fitting the model.

$$B_0 = p_1 \cdot (S - 1)^{p_2} \cdot \underbrace{\left(\frac{m_{\text{ML}} \cdot (w^*(\vartheta) - w(\vartheta)) + m_{\text{seed}}}{V} \right)^{p_3}}_{\text{suspension density } M_T} \quad (1.15)$$

1.2.4 Solution Mass Balance and Supersaturation

As indicated in the overall crystallization model structure shown in Figure 1.1, the final step for each iteration step is the determination of the mass balance and subsequently the determination of supersaturation which is a prerequisite for the growth kernel, agglomeration kernel and nucleation kernel. Proceeding from the updated number density function calculated by solving the differential equation system, the updated crystal mass is calculated as shown in Equation 1.16.

$$m_{\text{cr}}(t) = V \cdot \rho_{\text{cr}} \cdot \phi \cdot \sum_i L_i^3 n_i(t) \Delta L_i \quad (1.16)$$

Since the crystal mass for the time step is now known, it is now possible to calculate the updated concentration in the solution. The required parameters here are the initial mass of the solute in the solution m_0 as well as the seed mass m_{seed} which serves as input parameter for the overall crystallization model. Combining all parameters yields Equation 1.17 for

calculation of the mass fraction in the solution which is implemented into the crystallization model.

$$w(t) = \frac{m_0 - (m_{\text{cryst}} - m_{\text{seed}})}{m_0 - (m_{\text{cryst}} - m_{\text{seed}}) + m_{\text{solvent,init}}} \quad (1.17)$$

While DWSIM offers the possibility to apply activity coefficient models to simulations which would offer the possibility to determine the updated supersaturation, due to issues with substance data which will be discussed later on, this work refrained from using the internal DWSIM methods. Hence, the supersaturation was calculated from as the ratio of the current concentration $w(t)$ to the equilibrium concentration $w^*(\vartheta(t))$ according to Equation 1.18.

$$\mathcal{S} = \frac{w(t)}{w^*(\vartheta(t))} \quad (1.18)$$

1.2.5 Model Inputs and Outputs

An application of the introduced model equation produces the overall model integrated into a custom DWSIM script. In order to fully utilize the script a number of parameters need to be predefined. A schematic overview is given in Figure 1.2. The predefined parameters include properties of the seed including the mass and the initial distribution of the seed crystals which are passed as number per size class and converted to the number density distribution used in calculations, the model parameters for the modeling equations and information on substance data and the crystallizer configuration.

Thereby, substance data includes the density of crystal and mother liquor, as well as the solubility curve of the system. On the other hand, the crystallizer configuration comprises the number of crystallization modules and cycle time which are used to calculate the total crystallization time, the temperature profile and starting and end temperature.

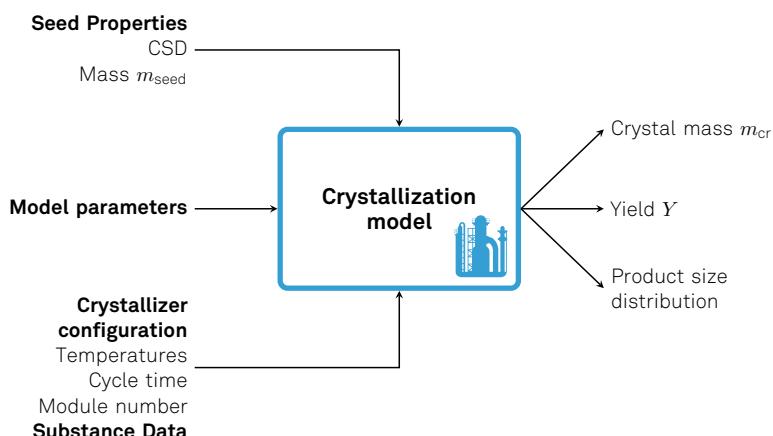


Figure 1.2: Schematic overview of the most important input and output parameters passed to and from the crystallization model.

On the other hand, the crystallization model provides information as output which are then used as calculation parameters in subsequent models. The main parameters of interest are the crystal mass, the resulting yield and the product size distribution which is converted to a Q_3 distribution as shown in Equation 1.19.

$$\Delta Q_3 = \frac{V \rho_{\text{cr}} \phi L_i^3 n_i \Delta L_i}{\sum_i (V \rho_{\text{cr}} \phi L_i^3 n_i \Delta L_i)} = \frac{L_i^3 n_i \Delta L_i}{\sum_i (L_i^3 n_i \Delta L_i)} \quad (1.19)$$

However, other calculated parameters which are not necessarily of interest in subsequent calculations are output as well. These include the crystal growth rate, calculated concentration as well as temperature and solubility for validation purposes.

1.2.6 Parameter Determination

Depending on the kernel function used for the determination of the growth rate, the overall crystallization model features a total of 7 (for the BCF-kernel function) or 8 (for the exponential kernel function) unknown parameters. In an effort to determine the parameters two optimization algorithms have been implemented.

The starting point of the optimization is given in a general least squares function that is set to be minimized. In the given case, two general least squares functions have been investigated for the regression of parameters to experimental data published in [28]. These include a function minimizing the Δ between significant diameters such as the d_{25} , d_{50} and d_{75} and is given in Equation 1.20. The second optimization problem was given in the least squares function involving the number of particles in the reaction vessel as shown in Equation 1.21 which was determined from data by Schmidt under the assumption that particles can be considered perfect spheres.

$$\min_{G_1, G_2, G_3, \beta_0, L_{\text{crit}}} \sqrt{(\Delta d_{25})^2 + (\Delta d_{50})^2 + (\Delta d_{75})^2} \text{ with } \Delta d_i = d_{i,\text{sim}} - d_{i,\text{exp}} \quad (1.20)$$

$$\min_{p_1, p_2, p_3} \sqrt{(\Delta N_{\text{particles}})^2} \quad (1.21)$$

There are several things important to note about the underlying optimization. First of all, the parameters which are set to be varied in the objective functions exert the most influence on the respective parameter. However, each objective function introduces a bias for the second function. For instance, a high agglomeration rate reduces the number of particles which influences results of the second function. On the other hand, a high nucleation rate introduces a large number of particles in the fine fraction of the particle size distribution which - in turn - introduces a bias of the first objective function. Since the optimization problem itself is highly nonlinear and complex to solve as it is given the fact that a regres-

sion needs to be performed purely based on experimental size distribution, this mutual biasing of objective functions further complicates the given problem.

In order to attempt to solve the underlying optimization problems, the *Nelder-Mead* algorithm which can be classified a direct search method and *Particle Swarm Optimization* (PSO) which can be classified as a *metaheuristic* are employed. While PSO is introduced hereafter, the *Nelder-Mead* algorithm is explained in more detail in chapter 1.5 since it is unable to accommodate for both objective simultaneously.

Particle Swarm Optimization

In contrast to the *Nelder-Mead* algorithm, PSO is a metaheuristic or a solution method combining local improvement procedures and a higher level strategy. Metaheuristics are oftentimes capable of providing a robust and efficient search method in a given solution space. [29]

PSO itself was derived in an attempt to describe the behavior of swarms of birds or fishes and later on found to provide adequate solution to optimization problems. The initialization of PSO is performed by generating a swarm of n particles for which the solution to the optimization is calculated. Each particle thereby has a position as determined by the vector X , a velocity determined by the vector v and a vector p_{best} which denotes the local best position. [29, 30]

After the results of the first iteration is calculated, the vector g_{best} is determined which stores the parameters of the globally best found parameter combination. As the iteration is finished, the velocity vector of each particle is updated according to Equation 1.22 where the weight w and the *behavioral* and *cognitive* components c_1 and c_2 are used as hyperparameters. Additionally, the updated velocity is based on the parameters r_1 and r_2 which are chosen randomly in the range $]0, 1[$. Subsequently, the position of the particles is updated as shown in Equation 1.23. [30, 31]

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 (p_{best_i} - X_i(t)) + c_2 \cdot r_2 (g_{best} - X_i(t)) \quad (1.22)$$

$$X_i(t+1) = X_i(t) + v_i(t+1) \quad (1.23)$$

An overall flowchart of the algorithm is given in Figure 1.3. An application to an exemplary function is further illustrated in chapter 1.5.

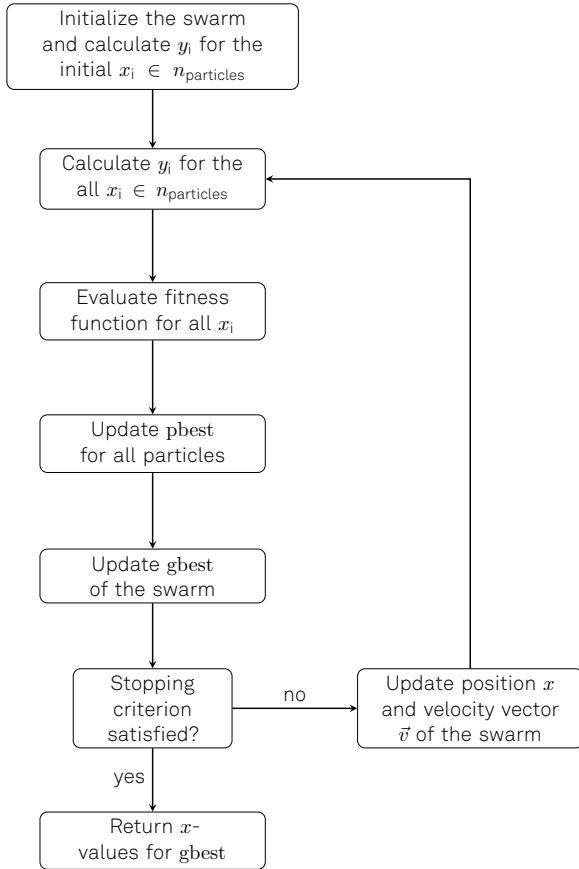


Figure 1.3: Schematic flowchart of particle swarm optimization.

Nelder-Mead Algorithm

The *Nelder-Mead* algorithm was proposed in 1965 and is a variation of conventional simplex algorithms [32]. There, a simplex (n -Simplex) is a polytope with n dimensions which is formed as the convex hull of $n + 1$ vertices [33]. In a two-dimensional optimization problem, a Simplex can be visualized as a triangle and is made up of $n + 1$ vertices, i.e. three points. In contrast to basic simplex methods, the *Nelder-Mead* algorithm does not only mirror the initial simplex based on the values of objective functions of each vector, but much rather includes the operations of reflecting, expanding, contracting and compressing the simplex as shown schematically in Figure 1.4.

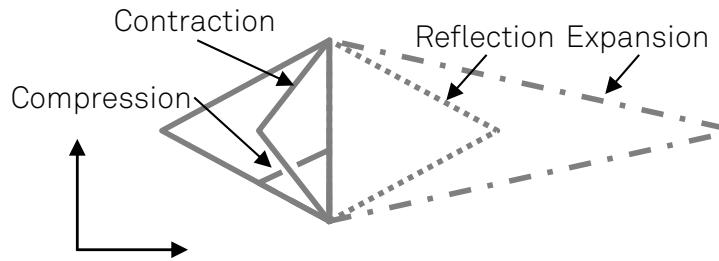


Figure 1.4: Illustration of the actions performed to the simplex in the *Nelder-Mead* algorithm. The original simplex is depicted with the solid line. Depicted are the operations reflection, expansion, contraction and compression.

Which operation is performed is entirely dependent on the values of the objective functions of each vector. The course of one iteration, starting from the initial simplex can be visualized by the decision tree shown in Figure 1.5. In order to elucidate the algorithm further, an illustration on an exemplary function is given hereafter.

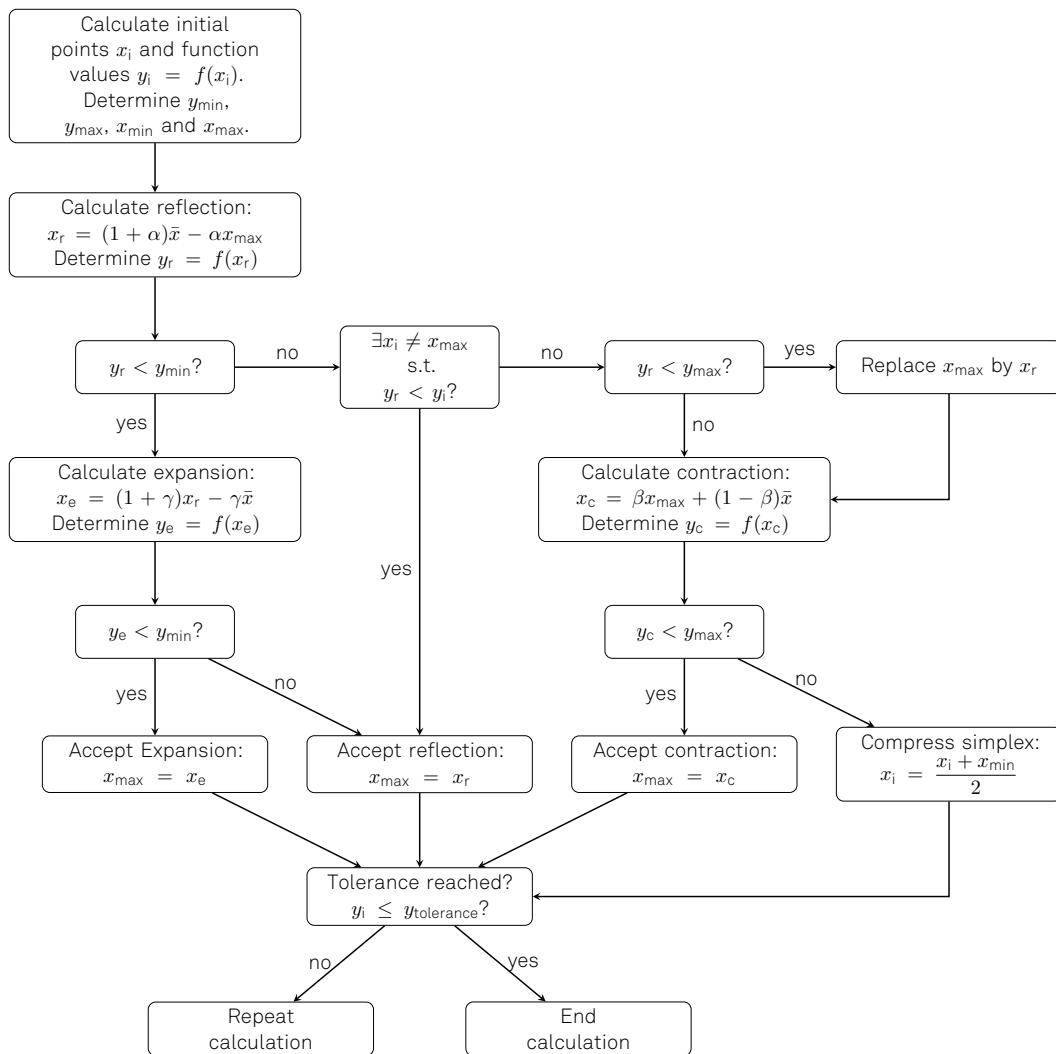


Figure 1.5: Schematic flowsheet of the *Nelder-Mead* algorithm. Adapted from [32]

Exemplary function for the illustration of optimization algorithms

$$f(x, y) = (x - \pi)^2 + (y - e)^2 + \sin(3x + 1.41) + \sin(4y - 1.73) \quad (1.24)$$

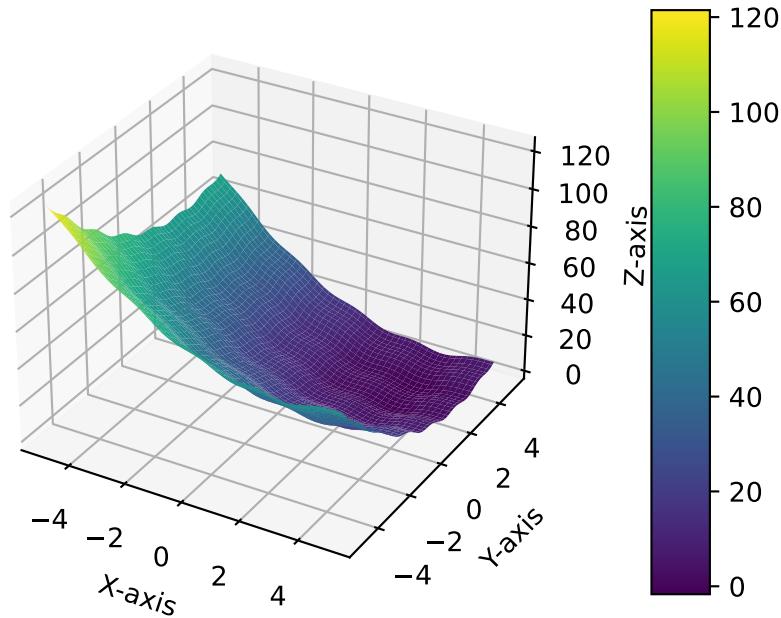


Figure 1.6: 3D surface plot of an exemplary objective function.

Exemplary illustration of the Nelder-Mead algorithm

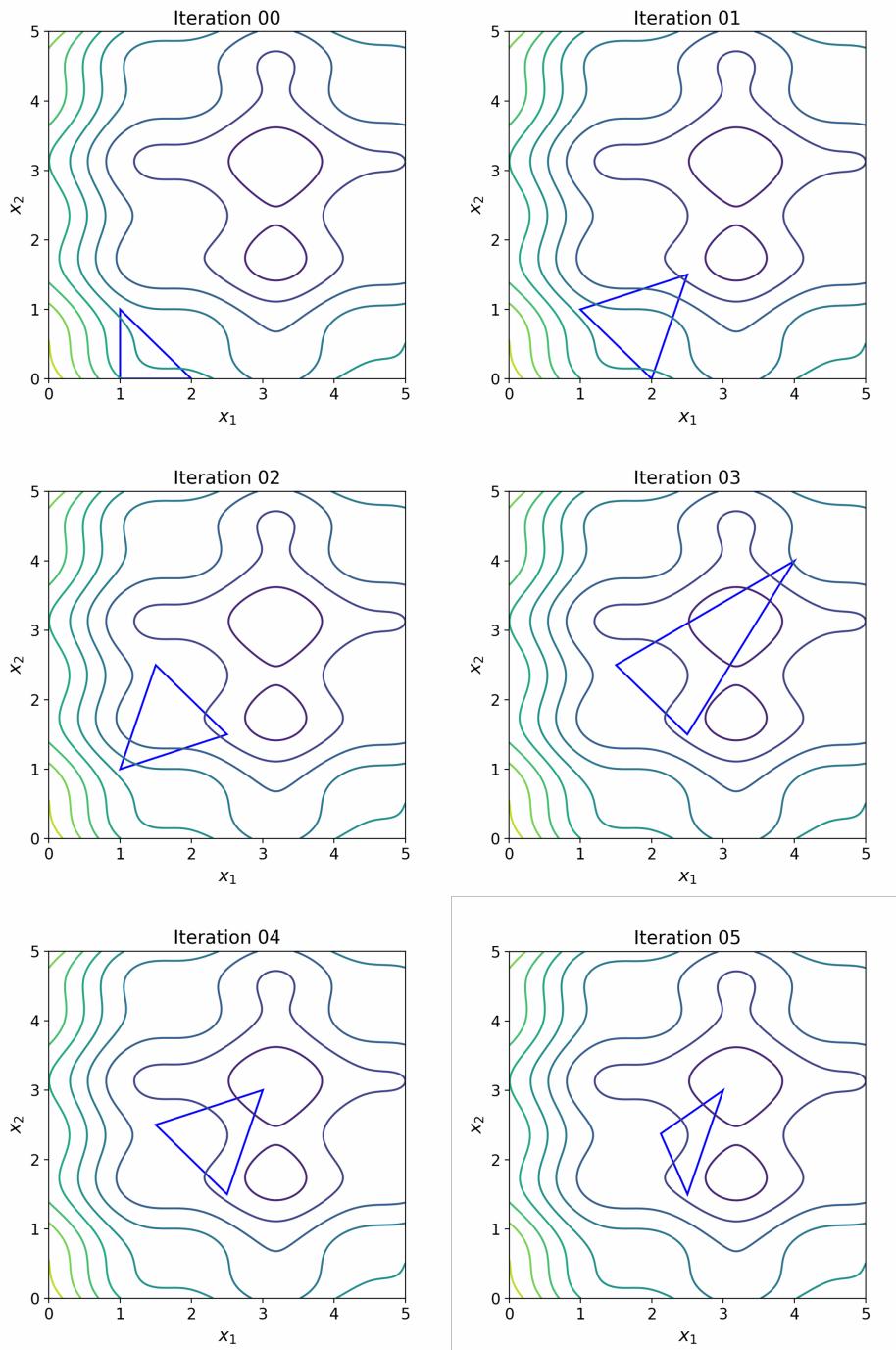


Figure 1.7: Exemplary illustration of the Nelder-Mead algorithm.

Exemplary Illustration of Particle Swarm Optimization

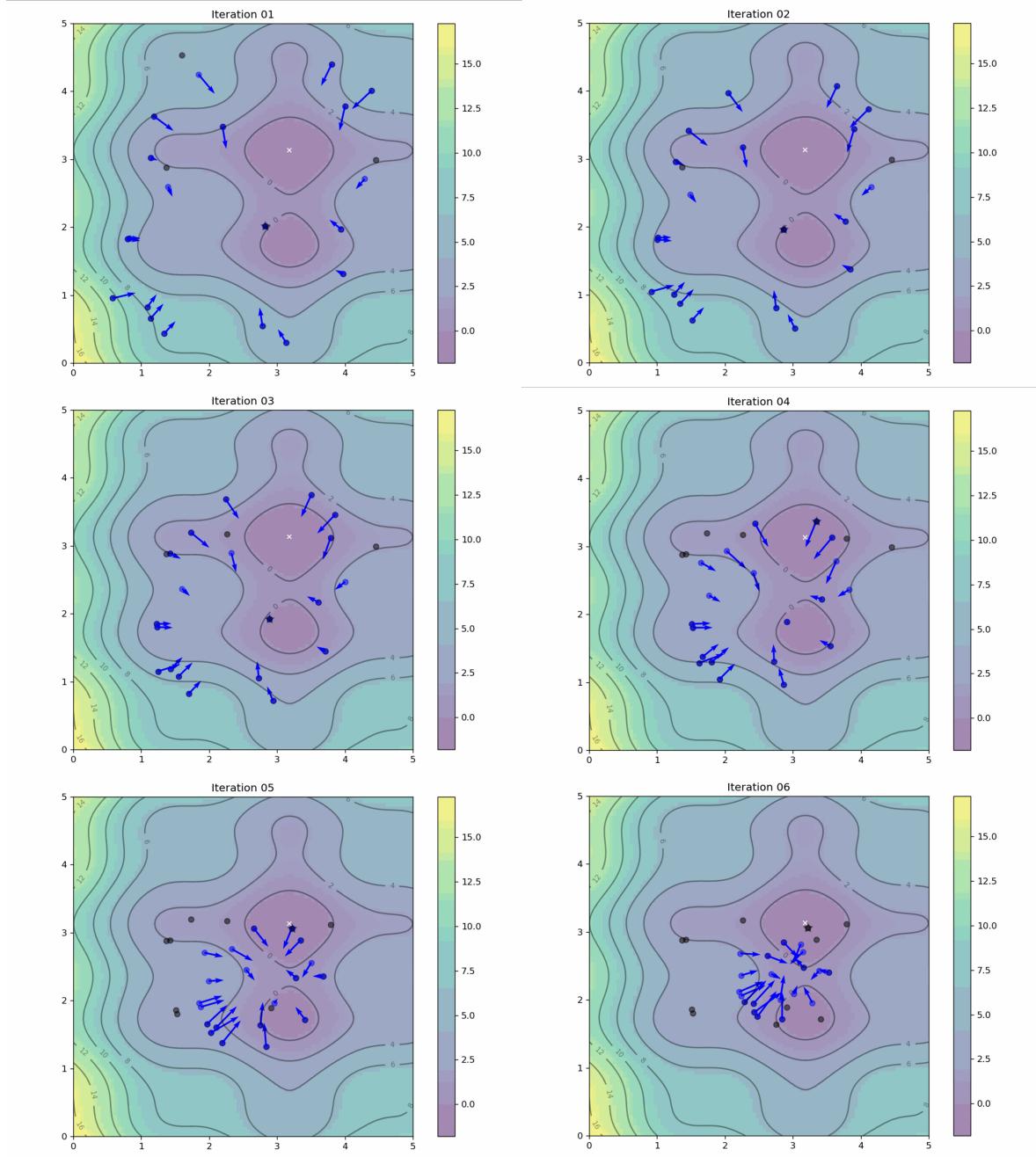


Figure 1.8: Exemplary illustration of Particle Swarm Optimization

At the start of the algorithm a “swarm” of particles is generated randomly and evaluated regarding the function value. As can be seen, one particle in the first iteration is able to find a function value close to the optimum and thus the rest of the particles adjust their velocity vector and start diffusing towards the current so-called gbest. As can be seen, all velocity vectors have different lengths which is due to the calculation method which depends on the particles value and a random factor.

During the diffusion of particles, another particle encounters a new gbest in iteration 4. However, as the particle swarm exhibits velocity vectors that are highly focused on the previous gbest as determined by the weight w , particles take a few iterations until their trajectory turns and they diffuse towards the new gbest.

1.2.1 Temperature Profiles

In contrast to the experimental investigation of crystallization processes, the theoretical investigations are less confined by experimental limitations. One such limitation of experimental work is given in the choice of temperature profile which is especially important on the QCFBC where the temperature profile needs to be and can be maintained even when advancing the container to the next module.

As investigated before, the advancement of modules on the apparatus can cause temperature jumps of up to 22.5 K. Additionally, experimental profiles need to be adjusted with regards to the cooling capacity of the apparatus.

Since the underlying model is developed based on previous experimental work, the emphasis with regards to temperature profiles on the apparatus is put on previously established profiles. These include a linear, a progressive and an oscillating profile. A plot of the respective temperature profile for a starting temperature of $\vartheta_{\text{start}}=59^{\circ}\text{C}$, an ending temperature of $\vartheta_{\text{end}}=20^{\circ}\text{C}$ and a process time of 7200 s with 600 s of holding time at the start and end of the process given in Figure 1.9.

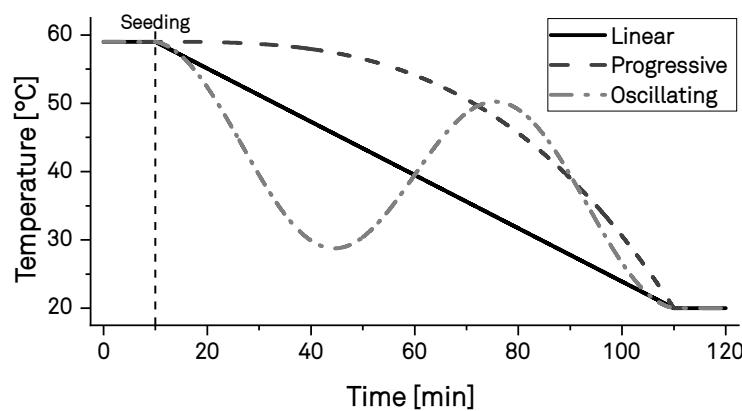


Figure 1.9: Temperature profiles investigated on the QCFBC including the linear, progressive and oscillating profile at $\vartheta_{\text{start}}=59^{\circ}\text{C}$, $\vartheta_{\text{end}}=20^{\circ}\text{C}$, $t_{\text{pre}}=t_{\text{ripe}}=600\text{ s}$ and $t_{\text{process}}=7200\text{ s}$.

In the given plot, each temperature profile comes with a holding time at the start and at the end of the profile. The holding time at the start of the profile is owed to the cooling of mother liquor when transferred to the container. Hence, experimental theses have assigned a preheating buffer time. The holding time at the end of the process is referred to as ripening

time and is implemented in order to promote a depletion of supersaturation which is set to increase the overall yield.

Linear Profile

As can be seen in Figure 1.9, the linear temperature profile can be characterized by its' simplicity and has therefore been used as a benchmark profile in previous experimental work. While the profile itself is easily characterized and implemented with constant temperature jumps in between modules apart from the first module if a preheating time is applied, in terms of crystallization, the linear temperature profile can prove to be unfavorable as a chance of secondary nucleation and low growth rates can occur. The linear temperature profile is defined by Equation 1.25 and characterized by the starting time ϑ_{start} , the ending time ϑ_{end} , the preheating time t_{pre} and the ripening time t_{ripe} .

$$T_l(t) = \begin{cases} T_{\text{start}} & t < t_{\text{pre}} \\ T_{\text{start}} - (T_{\text{start}} - T_{\text{end}}) \cdot \left(\frac{t - t_{\text{pre}}}{t_{\text{cryst}} - t_{\text{pre}} - t_{\text{ripe}}} \right)^3 & t_{\text{pre}} \leq t < t_{\text{cryst}} - t_{\text{ripe}} \\ T_{\text{end}} & t \geq t_{\text{cryst}} - t_{\text{ripe}} \end{cases} \quad (1.25)$$

Progressive Profile

In contrast to the linear profile, the progressive (or cubic) temperature profile starts with a low cooling rate that is gradually increased over the course of the process. This way, a near constant supersaturation is set to occur which reduces the occurrence of secondary nucleation. Due to the resulting reduced surface area of crystals, a reduction in crystallization yield is to be expected. Nevertheless, the reduced occurrence of secondary nucleation makes the progressive profile the benchmark profile in this work with a calculation according to Equation 1.26.

$$T_p(t) = \begin{cases} T_{\text{start}} & t < t_{\text{pre}} \\ T_{\text{start}} - (T_{\text{start}} - T_{\text{end}}) \cdot \left(\frac{t - t_{\text{pre}}}{t_{\text{cryst}} - t_{\text{pre}} - t_{\text{ripe}}} \right)^3 & t_{\text{pre}} \leq t < t_{\text{cryst}} - t_{\text{ripe}} \\ T_{\text{end}} & t \geq t_{\text{cryst}} - t_{\text{ripe}} \end{cases} \quad (1.26)$$

Oscillating Profile

Last but not least, the oscillating temperature profile has been investigated experimentally to obtain a narrower CSD by heating the suspension and thereby promoting the dissolution of fine crystals which form due to secondary nucleation in the container. Since the

oscillating profile depends not only on preheating time, ripening time as well as start and ending temperature, but much rather also on four constants $A - D$ as can be seen in Equation 1.27, the profile needs to be fitted for variation of the cycle time. Since the underlying crystallization model does not feature a dissolution kinetic, the oscillating profile will play a subordinate role in this work as the model is unable to fully capture the profiles' kinetics.

$$T_0(t) = \begin{cases} T_{\text{start}} & t < t_{\text{pre}} \\ A \cdot \cos \left(B \cdot \frac{t-t_{\text{pre}}}{t_{\text{cryst}}-t_{\text{pre}}-t_{\text{ripe}}} \right) - C \cdot \left(\frac{t-t_{\text{pre}}}{t_{\text{cryst}}-t_{\text{pre}}-t_{\text{ripe}}} \right) + D & t_{\text{pre}} \leq t < t_{\text{cryst}} - t_{\text{ripe}} \\ T_{\text{end}} & t \geq t_{\text{cryst}} - t_{\text{ripe}} \end{cases} \quad (1.27)$$

1.3 Solid-Liquid Separation Model

The solid-liquid separation model used for simulating the filter belt crystallizer is split into two sub-models. As long as the cake saturation exceeds a value of $S = 1$ a filtration model based on the *Darcy*-equation is employed. After dropping below the saturation threshold the model is switched and instead a deliquoring model, based on physical basics presented in ?? is employed. Hereafter, the relevant equations and calculations performed in the model will be presented.

1.3.1 Filtration Model

The filtration model implemented in DWSIM consists of different substeps which are performed subsequently in the calculation. As a first step, the model reads inputs from the preceding input file which are then used to estimate filter cake parameters. Utilizing the calculation results the main calculation is started and outputs are generated. The calculation and governing equations are given in the following.

1.3.1.1 Porosity Calculation

The first step in the filtration model is the estimation of the filter cake porosity. The porosity is required on the one hand to calculate the height of the filter cake and the resulting volume of the filter cake which is especially of importance in the drying model used in the last process step, as well as - on the other hand to determine fundamental cake properties used in the deliquoring calculation.

Due to the randomness being a factor in the distribution of particles in real filter cakes combined with non-idealities in mixing in the crystallization vessel, it is important to note

that the calculation merely estimates a cake porosity. A more accurate model based prediction could be made by geometrically discretizing the vessel in the crystallization process and applying knowledge gained from already existing computational fluid dynamic studies which could provide more accurate results with regards to cake properties.

The applied estimation of the cake porosity is based on a method which is used to determine information on the heat capacity of porous bulk materials. Beginning with the particle size distribution which - in this case - is taken from the crystallization model output an auxiliary parameter ξ is calculated [34]:

$$\xi = \sqrt{\frac{\sum_i \frac{\Delta Q_i}{d_i^2}}{\left(\sum_i \frac{\Delta Q_i}{d_i^2}\right)^2} - 1} \quad (1.28)$$

ξ then serves as input parameter to Equation 1.29 which estimates the porosity of a polydisperse bulk material ε_{PD} . Since information on polydisperse porosities require information on the properties of monodisperse bulk of the same material, the corresponding monodisperse porosity ε_{MD} serves as a second input parameter. Since no information on monodisperse bulks is given here, an estimation is made. An infinite, unordered, monodisperse bulk generally has a porosity within the range $0.36 \leq \varepsilon \leq 0.42$. If a bulk is tapped and a certain degree of compaction has been applied to the bulk this reduces the range of valid porosities to $0.36 \leq \varepsilon \leq 0.38$. Assuming that the cake filtration applies a level of compaction to the filter cake, a value of $\varepsilon_{MD} = 0.36$ has been assumed in the porosity calculation.

$$\varepsilon_{PD} = \varepsilon_{MD} \cdot (-0.112 \cdot \xi^3 + 0.017 \cdot \xi^2 - 0.259 \cdot \xi + 1) \text{ with } \varepsilon_{MD} \approx 0.36 \quad (1.29)$$

1.3.1.2 Model Equations and Assumptions

After the porosity estimation is performed, the calculation of the main loop is started. The height of the filter cake is used to calculate the height of the filter cake according to Equation 1.30.

$$h_{cake} = \frac{m_{cr}}{(1 - \varepsilon) \cdot \rho_{cr} \cdot A_{filt}} \quad (1.30)$$

While in reality the filter cake is formed over the course of the filtration process, this equation assumes the formation of a filter cake in the time used to shift modules on the apparatus. However, the used solution to the *Darcy*-equation does not take the change in cake resistance over the course of the filtration into account and rather assumes a constant value here [35]. Hence, using Equation 1.30 provides a valid approach.

As for the solution to the *Darcy*-equation the well-established approach using a constant

Δp is used. Since this approach has already been introduced in ?? further explanations will not be given in the following. As an additional output to the model, the filtration script also calculates the filtration speed v_{filt} which is defined as the ratio of filtrate flow rate \dot{V}_{filtr} and filtration area A_{filt} as given in Equation 1.31. [35]

$$v_{\text{filt}} = \frac{\Delta p}{\sqrt{(\eta\beta)^2 + (2\Delta p\eta\alpha\chi) \cdot t}} \quad (1.31)$$

Further outputs include the residual moisture of the filter cake, the cake saturation as well as properties such as the suspension viscosity and the suspension density. For the calculation of the suspension density, Equation 1.32 is used. Here, it is assumed that the density of the sucrose in solution equals that of the unsolved sucrose.

$$\eta_{\text{sus}} = \left(\frac{w_{\text{cr,final}}}{\rho_{\text{cr}}} + \frac{1 - w_{\text{cr,final}}}{\rho_{\text{solv}}} \right)^{-1} \quad (1.32)$$

An idealized assumption in the filtration model is the absence of crystallization kinetics. Since the filter cake is saturated at the beginning of the filtration process and a supersaturation is still going to be present, in reality crystal growth would still take place. Furthermore, if the filter cake is formed and a remainder of liquid is still present, a large number of particle interactions and thus agglomeration is set to take place.

1.3.1.3 Parameter Determination

The theoretical description of filtration processes requires knowledge from experiments since a theoretical description of the parameters such as the filter medium resistance β_{FM} , cake resistance α_{cake} and constant χ_{cake} is not easily possible. As discussed in ?? experimental procedures for the determination of these parameters exist and are standardized, however past theses have not employed these methods. Since this work relies on previous experimental data to propose a theoretical model, an estimation method based on available data has been employed in the filtration model.

From data collected previously [28] a number of filter curves with data collected in 30 s intervals are known. Assuming, that the filtration process is not finished after the first 30 s this implies one known point $(V_{\text{known}}, \frac{t}{V_{\text{known}}})$ on the filter curve. Since this does not

suffice for the formulation of a linear equation five data points are additionally introduced to generate a family of curves as shown in Equation 1.33 to Equation 1.37.

$$f_1(V) = \underbrace{\frac{1}{V_{\text{known}} - 0}}_{m_1} \cdot V + \underbrace{\left(\frac{t}{V_{\text{known}}} - 1 \right)}_{n_1} \quad (1.33)$$

$$f_2(V) = \underbrace{\frac{(t/V)_{\text{known}} - 1}{V_{\text{known}} - 0}}_{m_2} \cdot V + \underbrace{1}_{n_2} \quad (1.34)$$

$$f_3(V) = \underbrace{\frac{(t/V)_{\text{known}} - n_3}{V_{\text{known}} - 0}}_{m_3} \cdot V + \underbrace{\frac{n_1 + n_2}{2}}_{n_3} \quad (1.35)$$

$$f_4(V) = \underbrace{\frac{(t/V)_{\text{known}} - n_4}{V_{\text{known}} - 0}}_{m_4} \cdot V + \underbrace{\frac{n_1 + n_3}{2}}_{n_4} \quad (1.36)$$

$$f_5(V) = \underbrace{\frac{(t/V)_{\text{known}} - n_5}{V_{\text{known}} - 0}}_{m_5} \cdot V + \underbrace{\frac{n_2 + n_3}{2}}_{n_5} \quad (1.37)$$

These equations were generated as follows. All equations assume a linear relationship. Equation 1.33 is generated using the lowest possible y-Intercept with $\left(\frac{t}{V_{\text{known}}} - 1 \right)$. On the other hand, Equation 1.34 represents the steepest possible curve, by defining the lowest possible y-Intercept to $n_2 = 1$. These two extrema define the maximum possible confidence interval in which the filtration needs to take place.

By averaging both y-intercepts, n_1 and n_2 a more realistic approach for the filter curve (Equation 1.35) is generated. Since the calculation of the deliquoring model only takes one course of filtration into account, this more realistic approach is used later on. Equation 1.36 and Equation 1.37 are then generated by averaging n_1 and n_3 and n_2 and n_3 respectively. This second average generates a second - more defined confidence interval which is set up to provide a second, more realistic uncertainty quantification.

1.3.2 Deliquoring Model

In contrast to the filtration model, the deliquoring model now needs to take the gas phase into account. As outlined in ??, theoretical models for the description exists. As such, for this work it has been chosen to use the model given by Wakeman and Tarleton [36].

The overall model relies on the calculation of various parameters by more or less empirical calculations. However, as outlined before, empirical models represent the standard in deliquoring modeling. The starting point of the Wakeman-model is the determination of the threshold pressure p_b which is calculated as Equation 1.38. The capillary pressure is

derived from capillary pressure curves. In order to reduce saturation, a minimum pressure needs to be applied which is defined as threshold pressure. This hysteresis between saturation and pressure is visible in capillary pressure curves.

$$p_b = \sum_i \Delta Q_3 \cdot \frac{4.6(1 - \varepsilon_{av})\sigma}{\varepsilon_{av} d_{p,i}} \quad (1.38)$$

In order to accomodate the underlying polydisperse bulk, a modification of the equation proposed by Wakeman is performed. Instead of relying on the empirical calculation of a mean particle size \bar{x} , the model equation has been extended using an additive hypothesis taking the calculated Q_3 -distribution into account. The above equation requires knowledge about the surface tension of the liquid as well as knowledge about the average porosity ε . The relation used to calculate the surface tension is derived from literature data [37]. While Wakeman provides an equation for the average porosity, here, the porosity determined in the filtration model has been utilized.

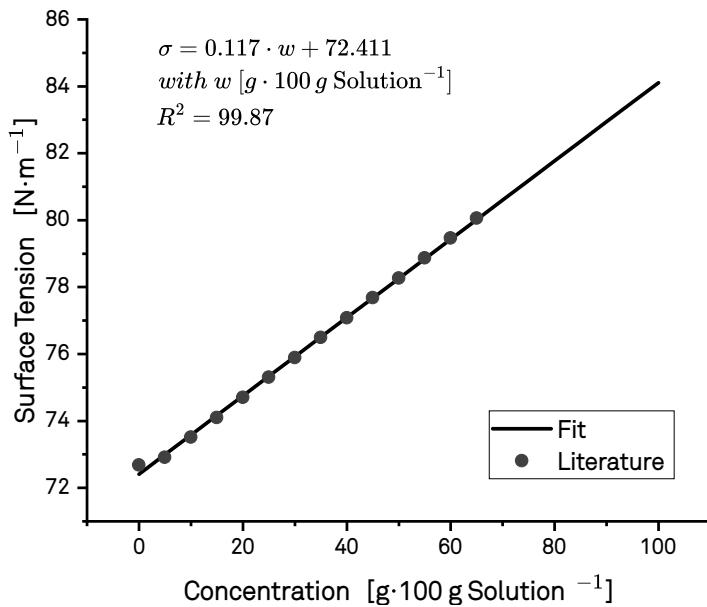


Figure 1.10: Determined correlation for the determination of surface tension σ at different concentrations of water-sucrose solutions at 21 °C. Literature data was taken from [37] and the function was determined using Microsoft Excel.

The correlations used for calculation later on are based on the capillary number N_{cap} which is a dimensionless number defined as the ratio of dewatering forces to the retaining surface tension inside the filter cake. As seen above, the equation for the capillary number is extended by the previously introduced additive hypothesis to form:

$$N_{cap} = \sum_i \Delta Q_3 \cdot \frac{\varepsilon_{av}^3 d_{p,i}^2 (\rho_l g h_{cake} + \Delta p)}{(1 - \varepsilon_{av})^2 h_{cake} \sigma} \quad (1.39)$$

As introduced in ?? each filter cake has a theoretical limit S_∞ which cannot be exceeded when deliquoring a filter cake. Based on the capillary number it is possible to make use of a correlation for the determination of the irreducible saturation which has been correlated as Equation 1.40.

$$S_\infty = 0.155 \cdot (1 + 0.031 \cdot N_{\text{cap}}^{-0.49}) \quad (1.40)$$

The calculation of *Wakeman* relies on a second dimensionless number which is given in the dimensionless deliquoring time θ . The dimensionless time is continuously updated for each time step in the main calculation loop and is defined as Equation 1.41.

$$\theta = \frac{k_{\text{av}} p_b}{\eta \varepsilon_{\text{av}} (1 - S_\infty) L^2} t_d \quad (1.41)$$

As can be seen in the equation, the calculation of the dimensionless deliquoring time uses another unknown parameter k_{av} , the cake permeability which is calculated according to Equation 1.42.

$$k_{\text{av}} = \frac{1}{\alpha_{\text{av},\text{FC}} \rho_s (1 - \varepsilon_{\text{av}})} \quad (1.42)$$

This equation relies on an unknown parameter - the mean filter cake resistance $\alpha_{\text{av},\text{FC}}$. While the cake resistance is contained within the constants determined using the previously introduced *Darcy*-equation relations, it is only calculable as the product $\alpha_{\text{FC}} \chi_{\text{FC}}$. Therefore an approach from literature has been employed. As can be seen in Equation 1.43 the parameter is determined based on a relation derived from the *Kozeny-Carman* equation. [38]

$$\alpha_{\text{av},\text{FC}} = 180 \frac{1 - \varepsilon}{\varepsilon^3} \frac{1}{\phi^2 d_p^2 \rho_s} \quad (1.43)$$

Similarly to Equation 1.38, Equation 1.43 has been extended by an additive hypothesis to account for the underlying Q_3 -distribution:

$$\alpha_{\text{av},\text{FC}} = \sum_i \Delta Q_{3,i} \cdot 180 \frac{1 - \varepsilon}{\varepsilon^3} \frac{1}{\phi^2 d_{p,i}^2 \rho_s} \quad (1.44)$$

In order to fully calculate the deliquoring process according to this specific method, the reduced saturation S_R is introduced. This parameter is calculated via empirical equations given in Equation 1.45 based on the magnitude of the product of dimensionless deliquoring

time and the dimensionless pressure which is defined as the ratio of ambient pressure to threshold pressure.

$$S_R = \begin{cases} \frac{1}{1 + 1.08(\theta p/p_b)^{0.88}} & 0.096 \leq \frac{\theta p}{p_b} \leq 1.915 \\ \frac{1}{1 + 1.46(\theta p/p_b)^{0.48}} & 1.915 \leq \frac{\theta p}{p_b} \leq 204 \end{cases} \quad (1.45)$$

Using the well established approach that the change in saturation which is denoted as reduced saturation here is defined as $\frac{S - S_\infty}{1 - S_\infty}$ it is now possible to perform the final step in this deliquoring calculation method and returns the saturation S of the filter cake as:

$$S = S_R(1 - S_\infty) + S_\infty \quad (1.46)$$

For better visualization of the calculation process in the deliquoring model by Wakeman, the schematic overview of the order of calculated quantities is depicted in Figure 1.11.

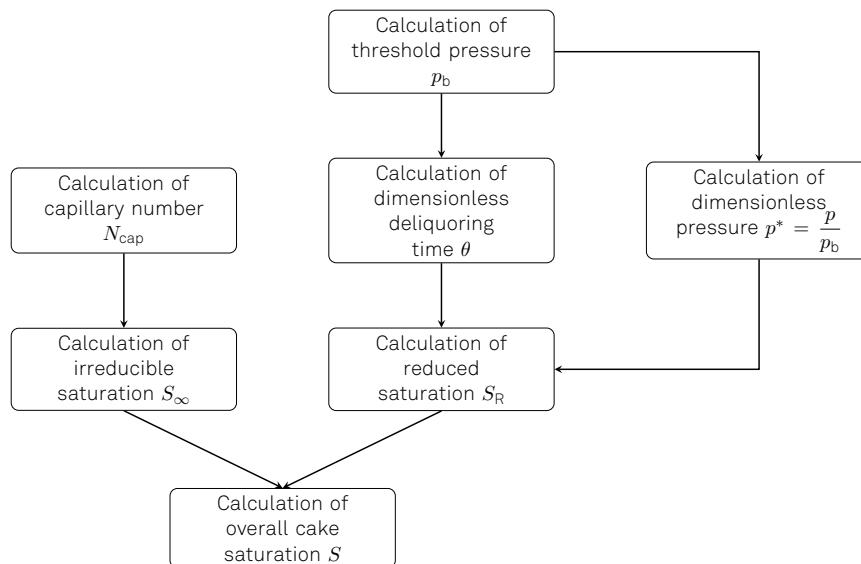


Figure 1.11: Schematic flowchart of the overall deliquoring model.

1.3.3 Filter Cake Washing Model

While the differential equation obtained by the dispersion model which was briefly introduced in ?? is possible and - given initial and boundary conditions - can be solved, this work does not implement the differential equation itself, but much rather on solutions compiled in design spaces in order to simplify calculations and reduce overall computational cost. In order to obtain a suitable correlation between washing ratio W and the ratio X^* which describes the concentration of solute in wash effluent. [36]

As introduced previously, the dispersion number $\frac{vL}{D_L}$ is of importance in the calculation and defines the used relation. Therefore, the first step in calculation is to determine the dispersion number as shown in Equation 1.47.

$$D_n = \frac{vL}{D_L} = \frac{\rho vx}{\eta} \frac{\eta}{\rho D} \frac{L}{x} \frac{D}{D_L} = \text{Re} \cdot \text{Sc} \frac{L}{x} \frac{D}{D_L} \quad (1.47)$$

The dispersion number is dependent on the dimensionless *Reynolds* number Re which is used to characterize the flow of wash liquor through the filter cake and the the *Schmidt* number Sc which describes the ratio of molecular diffusivity of momentum to that of mass. The last part of Equation 1.47, the ratio of the molecular diffusion coefficient to the axial dispersion coefficient $\frac{D}{D_L}$ depends on the tortuosity of the filter cake. In order to properly characterize this ratio, three empirical correlations as given in Equation 1.48 exist. [36]

$$\frac{D}{D_L} = \begin{cases} \frac{1}{\sqrt{2}} & \text{Re} \cdot \text{Sc} < 1 \\ \frac{1}{\sqrt{2}} + 1.75 \text{Re} \cdot \text{Sc} & \text{Re} \cdot \text{Sc} \geq 1 \& H_{FC} \geq 0.1 \text{ m} \\ \frac{1}{\sqrt{2}} + 55.5(\text{Re} \cdot \text{Sc})^{0.96} & \text{Re} \cdot \text{Sc} \geq 1 \& H_{FC} < 0.1 \text{ m} \end{cases} \quad (1.48)$$

The now obtained dispersion number allows to choose a suitable function $X^* = f(\mathcal{W})$. An exemplary overview for a dispersion number of 1 is given in Equation 1.49. The remaining underlying functions for dispersion numbers in the range 0.01-10000 used in the model are given in the chapter 1.5.

$$X^* = \begin{cases} 1 & 0 \leq \mathcal{W} < 0.1 \\ 1.1433 - 1.9882\mathcal{W} + 3.5187\mathcal{W}^2 - 2.3065\mathcal{W}^3 + 0.5687\mathcal{W}^4 & 0.1 \leq \mathcal{W} < 1.7 \\ 0.3515\mathcal{W}^{-1.4654} & \mathcal{W} \geq 1.7 \end{cases} \quad (1.49)$$

A second, arguably more important measure is the fraction of “contaminant” removed from the filtercake F which is calculated as the integral of the ratio X^* [36]:

$$F = \int_0^{\mathcal{W}} \frac{X - X_{\text{feed}}}{X_0 - X_{\text{feed}}} \quad (1.50)$$

This equation has been solved and compiled as a design space by *Wakeman* and *Tarleton* and implemented into the model. The relationships for dispersion number and washing ratio are given in the appendix. According to *Wakeman* and *Tarleton* a correction of the washing ratio for predeliquored filter cakes according to Equation 1.51 needs to be performed since the washing liquid needs to fill up the desaturated pores in order to perform

a washing effect [36]. However, according to literature this correction is unsubstantiated [39]. Existing studies show a contrasting effect of predeliquoring on washing performance. Thus, the correction step is not performed in the model equation.

$$\mathcal{W}_{\text{corr}} = \mathcal{W}_{S=1} + 15.1(1 - S) \exp(-1.56X^*) - 7.4(1 - S^2) \exp(-1.72X^*) \quad (1.51)$$

Experimental investigations of the process of washing on the filter belt crystallizer have been performed previously where it was determined that the filter cake washing is capable to reduce the loading of filter cake to a value of $X_{\text{cake}}^* = 0.2$ [28]. In order to account for the experimental results, the fractional removal F has been scaled accordingly. Since *Lapainis* also observed product losses of less than 1% for a filter cake washing with ethanol, this phenomenon is neglected in the calculation.

Model correlations for the concentration of solute in the wash effluent [36]

$$\begin{aligned}
 D_n = 0.01 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.1 \\ 0.0524W^{-0.5311} & W \geq 0.1 \end{cases} \\
 D_n = 0.05 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.1 \\ 0.1036W^{-0.5848} & W \geq 0.1 \end{cases} \\
 D_n = 0.1 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.1 \\ 0.1364W^{-0.6037} & W \geq 0.1 \end{cases} \\
 D_n = 0.5 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.1 \\ 1.0754 - 2.609W + 3.5187W^2 - 2.3065W^3 + 0.5687W^4 & 0.1 \leq W < 1.6 \\ 0.2481W^{-0.9950} & W \geq 1.6 \end{cases} \\
 D_n = 1 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.1 \\ 1.1433 - 1.9882W + 1.8761W^2 - 0.9206W^3 + 0.1798W^4 & 0.1 \leq W < 1.7 \\ 0.3515W^{-1.4654} & W \geq 1.7 \end{cases} \\
 D_n = 5 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.1 \\ 1.0583 - 0.0795W - 1.7285W^2 + 1.241W^3 - 0.2603W^4 & 0.1 \leq W < 2 \\ 2.1739 \cdot \exp(-1.7383W) & 2 \leq W < 4.2 \\ 0 & W \geq 4.2 \end{cases} \\
 D_n = 10 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.2 \\ 0.6630 + 2.3569W - 4.9493W^2 + 2.9684W^3 - 0.5826W^4 & 0.2 \leq W < 1.7 \\ 11.5698 \cdot \exp(-2.9575W) & 1.7 \leq W < 3.3 \\ 0 & W \geq 3.3 \end{cases} \\
 D_n = 50 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.5 \\ -5.3263 + 26.272W - 37.56W^2 + 21.47W^3 - 4.3437W^4 & 0.5 \leq W < 1.4 \\ 44.08 \cdot \exp(-4.667W) & 1.4 \leq W < 2.4 \\ 0 & W \geq 2.4 \end{cases} \\
 D_n = 100 \quad X^* &= \begin{cases} 1 & 0 \leq W < 0.6 \\ -10.01 + 39.21W - 46.46W^2 + 19.9W^3 - 2.122W^4 & 0.5 \leq W < 1.1 \\ 0.5789W^{-8.0948} & 1.1 \leq W < 2.2 \\ 0 & W \geq 2.2 \end{cases}
 \end{aligned}$$

$$\begin{aligned}
D_n = 500 \quad X^* &= \begin{cases} 1 & 0 \leq \mathcal{W} < 0.8 \\ -16.77 + 53.01\mathcal{W} - 49.87\mathcal{W}^2 + 14.23\mathcal{W}^3 & 0.8 \leq \mathcal{W} < 1.1 \\ 0.3095\mathcal{W}^{-7.5097} & 1.1 \leq \mathcal{W} < 2.1 \\ 0 & \mathcal{W} \geq 2.1 \end{cases} \\
D_n = 1000 \quad X^* &= \begin{cases} 1 & 0 \leq \mathcal{W} < 0.8 \\ 1.0583 + 0.0795\mathcal{W} - 1.7285\mathcal{W}^2 + 1.241\mathcal{W}^3 & 0.8 \leq \mathcal{W} < 1.1 \\ 94.13 \exp(6.9979\mathcal{W}) & 1.1 \leq \mathcal{W} < 1.8 \\ 0 & \mathcal{W} \geq 1.8 \end{cases} \\
D_n = 10000 \quad X^* &= \begin{cases} 1 & 0 \leq \mathcal{W} < 0.8 \\ -9.7886 + 26.83\mathcal{W} - 17.72\mathcal{W}^2 + 1.3069\mathcal{W}^3 & 0.8 \leq \mathcal{W} < 1.1 \\ 0.0427\mathcal{W}^{-7.9662} & 1.1 \leq \mathcal{W} < 1.4 \\ 0 & \mathcal{W} \geq 1.4 \end{cases}
\end{aligned}$$

Model correlations for Dispersion numbers, Washing Ratio and Fractional Removal [36]

Table 1.1: Empirical model correlations used for the filter cake washing model [36].

\mathcal{W}	Dispersion number											
	0.01	0.05	0.1	0.5	1	5	10	50	100	500	1000	10000
0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0.059	0.068	0.075	0.093	0.098	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.2	0.074	0.1	0.118	0.169	0.187	0.2	0.2	0.2	0.2	0.2	0.2	0.2
0.3	0.085	0.123	0.15	0.229	0.262	0.298	0.3	0.3	0.3	0.3	0.3	0.3
0.4	0.094	0.143	0.176	0.28	0.327	0.391	0.399	0.4	0.4	0.4	0.4	0.4
0.5	0.102	0.159	0.199	0.323	0.382	0.476	0.496	0.5	0.5	0.5	0.5	0.5
0.6	0.11	0.175	0.22	0.361	0.43	0.552	0.586	0.6	0.6	0.6	0.6	0.6
0.7	0.116	0.188	0.238	0.395	0.472	0.618	0.668	0.698	0.7	0.7	0.7	0.7
0.8	0.122	0.201	0.255	0.426	0.51	0.675	0.793	0.79	0.797	0.8	0.8	0.8
0.9	0.128	0.213	0.27	0.454	0.544	0.723	0.799	0.868	0.882	0.897	0.898	0.898
1	0.134	0.224	0.285	0.479	0.575	0.764	0.85	0.929	0.947	0.972	0.975	0.977
1.1	0.139	0.234	0.299	0.502	0.602	0.8	0.891	0.971	0.988	1	1	1
1.2	0.144	0.244	0.311	0.524	0.628	0.83	0.923	0.998	1	1	1	1
1.3	0.149	0.253	0.323	0.544	0.651	0.854	0.949	1	1	1	1	1
1.4	0.153	0.262	0.335	0.562	0.672	0.875	0.969	1	1	1	1	1
1.5	0.158	0.27	0.346	0.58	0.691	0.892	0.984	1	1	1	1	1
1.6	0.162	0.279	0.356	0.596	0.709	0.907	0.996	1	1	1	1	1
1.7	0.166	0.286	0.366	0.611	0.725	0.919	1	1	1	1	1	1
1.8	0.169	0.294	0.376	0.626	0.741	0.93	1	1	1	1	1	1
1.9	0.173	0.301	0.385	0.639	0.755	0.938	1	1	1	1	1	1
2	0.177	0.308	0.394	0.652	0.768	0.946	1	1	1	1	1	1
2.1	0.18	0.315	0.403	0.664	0.781	0.953	1	1	1	1	1	1
2.2	0.184	0.322	0.412	0.676	0.792	0.959	1	1	1	1	1	1
2.3	0.187	0.328	0.42	0.687	0.803	0.963	1	1	1	1	1	1
2.4	0.19	0.334	0.428	0.698	0.813	0.967	1	1	1	1	1	1
2.5	0.193	0.34	0.435	0.708	0.823	0.971	1	1	1	1	1	1
2.6	0.196	0.346	0.443	0.718	0.832	0.974	1	1	1	1	1	1
2.7	0.199	0.352	0.45	0.727	0.84	0.976	1	1	1	1	1	1
2.8	0.202	0.357	0.458	0.736	0.848	0.978	1	1	1	1	1	1
2.9	0.205	0.363	0.465	0.744	0.856	0.98	1	1	1	1	1	1
3	0.208	0.368	0.472	0.752	0.863	0.982	1	1	1	1	1	1
3.1	0.211	0.374	0.478	0.76	0.87	0.983	1	1	1	1	1	1
3.2	0.214	0.379	0.485	0.768	0.876	0.984	1	1	1	1	1	1
3.3	0.217	0.384	0.492	0.776	0.882	0.985	1	1	1	1	1	1
3.4	0.22	0.389	0.498	0.783	0.888	0.986	1	1	1	1	1	1
3.5	0.222	0.394	0.505	0.79	0.894	0.986	1	1	1	1	1	1
3.6	0.225	0.399	0.511	0.797	0.9	0.987	1	1	1	1	1	1
3.7	0.228	0.404	0.517	0.804	0.905	0.987	1	1	1	1	1	1
3.8	0.231	0.409	0.524	0.811	0.91	0.988	1	1	1	1	1	1
3.9	0.233	0.413	0.53	0.818	0.915	0.988	1	1	1	1	1	1
4	0.236	0.418	0.536	0.824	0.92	0.988	1	1	1	1	1	1

1.3.4 Overall Model Inputs and Outputs

All three models have been introduced as scripts for custom unit operations into DWSIM. Since the unit operations build on one another to constitute a unified model, the output from the crystallization model is read by the filtration model. The output used in the subsequent script includes the crystal size distribution, the crystal mass and the resulting remaining liquid mass.

Additionally, the model parameters for filter cake resistance and filter medium resistance are given from an external script. These can either be the previously introduced known data for $\frac{t}{V}$ and V or, in the case of a known filter curve, the slope and y-intercept of the filter curve in which case no uncertainty estimation is made.

Additionally, the model relies on general inputs of the apparatus and substances. General apparatus configuration includes cycle times, operating pressures and filtration time, which denotes the time preceding the washing model. The substance data is partially read from the DWSIM database and partially given by an input file.

The models used in the overall solid-liquid separation model also build on each other. Therefore, an exchange between saturation and moisture data as well as previous calculations like the height of the filter cake, mean diameter, cake resistance and cake porosity is made. Schematically, an overview on model inputs and outputs as well as data passed between models is given in Figure 1.12.

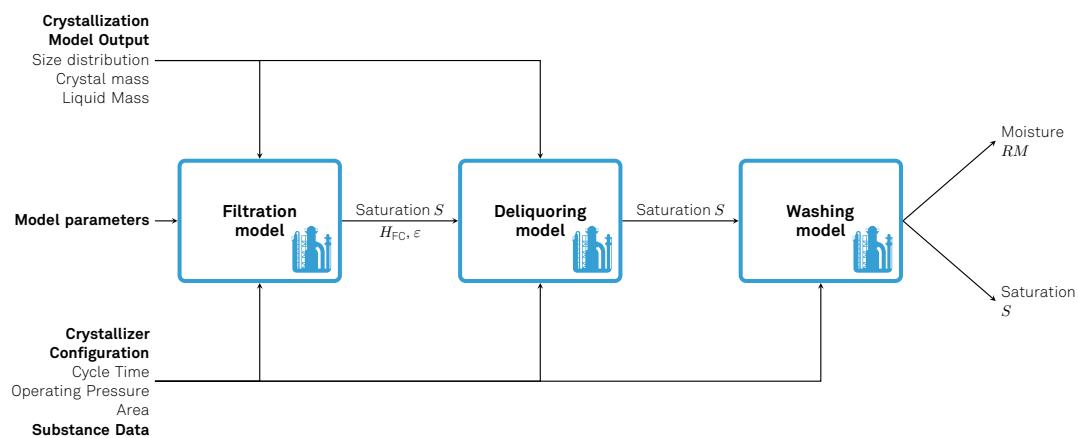


Figure 1.12: Schematic overview of the most important input and output parameters passed to and from the solid-liquid separation model. Parameters passed between the unit operations filtration, deliquorizing and filter cake washing are included.

1.4 Drying Model

The last unit operation in the process chain on the filter belt crystallizer is given in convective drying of the filter cake. Convective drying and especially through-air drying are a process which is of especially high interest in the food and the textile industry, due to which a large number of different modeling approaches exist [40–43]. However, these models oftentimes contain secondary kinetics, e.g. shrinking which are not or only partially applicable to the drying of filter cakes. Therefore, a drying model based on physical equations and the behavior of fluid flow through porous media is established. The following chapter describes the model equations, the calculation process and the calculation of necessary physical parameters.

1.4.1 Model Equations

Since convective drying is a coupled process of heat transfer and mass transfer, the overall model needs to account for four main phenomena which are given in the desaturation of the filter cake, the saturation of the drying air, heat transfer to the filter cake and the overall displacement of cold air from the filter cake and the subsequent replacement with preheated dry air. Hence, a physical model needs to provide four model equations which each account for one of these respective phenomena.

Cake loading

As it has been established in ??, an overall convective drying process consists of three main steps which are challenging to account for when modeling the process. However, the baseline of a drying process is always given in the convective drying rate \dot{m}_D . In order to refrain from setting up estimation procedures which are not guaranteed to deliver adequate results, the model used in this work extends the overall convective drying rate by an efficiency function $\Lambda(w_{\text{cake}})$ to the equation seen in Equation 1.52.

$$\dot{m}_D = \frac{\beta A}{R_s T} (p_i^{\text{LV}} - \varphi \cdot p_i^{\text{LV}}) \cdot \Lambda(w_{\text{cake}}) \quad (1.52)$$

While this function is purely empirical, it is fitted to experimental data collected previously [28]. The calculation of the fitting function is further described later on in section 1.4.2. The calculation of the filter cake loading further assumes, completely dry air, i.e. $\varphi = 0$ which reduces dependencies of the baseline drying rate to that of temperature and saturated vapor pressure p_i^{LV} . In this case, the saturated vapor pressure of air was retrieved from the DWSIM database where it is calculated using Lee-Kesler method [8].

This simplification combined with the retrieval of data from DWSIM only leaves the mass transfer coefficient β unknown. Here, a literature correlation proposed by *Daizo and Motoyuki* has been utilized which has been derived from investigation of porous beds of fine particles. [44]

By applying the dimensionless *Sherwood* number Sh_p which denotes the convective mass transfer rate to the diffusion rate and is given in Equation 1.53 and correlating it to the dimensionless *Péclet* number Pe_p which denotes the ratio of advective transport rate to diffusive transport rate, using an approach based on heat transfer phenomena in porous

beds, they were able to come up with a calculation method for β which requires knowledge of few parameters.

$$\text{Sh}_p = \frac{\beta D_p}{\mathcal{D}} = \frac{\phi_s}{6(1-\varepsilon)\zeta} \cdot \text{Pe}_p \quad (1.53)$$

$$\text{Pe}_p = \frac{D_p c_p G_0}{k_f} = \frac{D_p v_0}{\mathcal{D}} \quad (1.54)$$

As can be seen in Equation 1.55, the calculation requires the shape factor ϕ , the gas velocity v_0 , the porosity ε of the bed and a tortuosity factor ζ . Since knowledge about the tortuosity of the bed is not given, in this case it has been assumed to be $\zeta = 1$.

$$\beta = \frac{v_0}{ad_p \zeta} = \frac{\phi_s v_0}{6(1-\varepsilon)} \quad (1.55)$$

Based on the now known mass transfer factor β , the differential equation used in the calculation loop has been derived. In order to do so, a balancing equation for the unstationary change in the amount of liquid in the filter cake has been set up. Using the definition of the cake saturation and that of the porosity this yields the final differential equation as given in Equation 1.56. Here, the convective drying rate has been converted to a volumetric mass transfer rate which is related to the pure volume of crystals in the filter cake V_{cryst} . The full application of the balancing equation and the resulting derivation of the differential equation is given in section 1.7.1.1.

$$\frac{dw}{dt} = \frac{1}{\rho_{\text{cryst}}} \dot{D} \text{ with } \dot{D} = \frac{\dot{m}_{\text{D}}}{V_{\text{cryst}}} \left[\frac{\text{kg}}{\text{m}^3 \cdot \text{s}} \right] \quad (1.56)$$

As evident from the equation, a uniform filter cake with no change in loading over the filter cake height is assumed. This is done, since filter cakes generated in the crystallizer are generally small within the range of a few millimeters in height. While in reality, spatial deviations are set to occur, their relative magnitude can be assumed to be negligible.

Drying Air Loading

While the filter cake is assumed to have both a uniform loading and temperature, the differential equations describing air loading and temperature are set up in a way that both a temporal and spatial change over the filter cake is described. By setting up partial differential equations (PDE) it is possible to gain more information regarding the process itself, as well as to have a tool for assessing how physically meaningful the overall results are. The PDE which accounts for the change in loading of the drying air over time and the overall length of the filter cake in the z -direction is based on the adapted equation of continuity for

porous media [45–47]. Assuming that a change in mass occurs and no gradient in both the x - and y -direction occur and applying the equation only to one singular component i yields the PDE as given in Equation 1.57. Once again, the overall derivation is given in chapter 1.5.

$$\dot{D}_i = \rho \varepsilon (1 - S) \frac{\partial w_i}{\partial t} + \rho v_z \frac{\partial w_i}{\partial z} \quad (1.57)$$

Cake Temperature

The basis for calculation of the cake temperature is given in a general heat balance around the filter cake. By considering the relevant terms of heating of the filter cake, evaporation of the liquid and consideration of heat losses through the wall, it is possible to set up a general differential equation for the change in cake temperature over time:

$$c_p m \frac{dT_{\text{cake}}}{dt} = \dot{Q}_{\text{conv}} - \dot{Q}_{\text{loss}} - \dot{Q}_{\text{evap}} \quad (1.58)$$

Since the cake itself consists of three different components in the crystals, the liquid and the drying gas, the heat capacity c_p of the entire filter cake is defined as following:

$$c_p m = V_{\text{cake}} \left(\underbrace{\rho_s c_{p,s} (1 - \varepsilon)}_{\text{crystals}} + \underbrace{\rho_l c_{p,l} \varepsilon S}_{\text{liquid component}} + \underbrace{\rho_g c_{p,g} \varepsilon (1 - S)}_{\text{drying gas}} \right) \quad (1.59)$$

Assuming that the drying gas exerts little influence on the overall heat capacity it is neglected in this equation. Furthermore, since it is assumed that heat losses through the walls are negligible due to the relatively small quantifiable area coupled with a comparatively high gas flow rate, this term is neglected.

Substituting in the terms for convective heat transfer and evaporation yields the overall differential equation for the change in cake temperature over time which is given in Equation 1.60. As before, a more detailed overview of the terms and the derivation of the overall balancing equation is given in chapter 1.5.

$$\frac{1}{1 - \varepsilon} (\rho_s c_{p,s} (1 - \varepsilon) + \rho_l c_{p,l} \varepsilon S) \frac{dT_{\text{cake}}}{dt} = \frac{\alpha A}{V_{\text{cryst}}} \Delta T_{\text{cake}} - \dot{D} \Delta h^{LV} \quad (1.60)$$

Similar to the cake loading, it is assumed that the cake temperature is uniform over the entire filter cake without the occurrence of gradients. Furthermore, similar to the differential equation for filter cake loading, the differential equation features a transfer coefficient; in this case the heat transfer coefficient α . The calculation of α is performed using correlations involving the dimensionless *Nusselt* number. A more detailed explanation is given in section 1.4.1.1.

Drying Air Temperature

Last but not least the balancing equation for the temperature of the drying air also assumes a change in loading in the z -direction over the filter cake. To account for this, the PDE is once again based on the equation of continuity for porous media. However, this time the variation for heat transfer according to Equation 1.61 is employed.

$$V_{\text{cake}} \frac{\partial(T \rho_g c_{p,g} \varepsilon(1 - S))}{\partial t} + V_{\text{cake}} \nabla(\vec{u} T \rho_g c_{p,g}) = -\alpha A \Delta T_{\text{cake}} - \dot{m}_{\text{D}} \Delta h^{LV} \quad (1.61)$$

Here, the change in temperature over time and height of the filter cake is correlated to the loss terms of heat from the air which include heat transfer to the filter cake as well as the necessary heat required to evaporate the liquid from the filter cake. By once again neglecting the change in temperature in the x - and y -direction the final PDE can be obtained as:

$$\frac{V_{\text{cake}}}{V_{\text{cryst}}} \rho_g c_{p,g} \varepsilon(1 - S) \frac{\partial T}{\partial t} + \frac{V_{\text{cake}}}{V_{\text{cryst}}} \rho_g c_{p,g} u_z \frac{\partial T}{\partial z} = -\frac{\alpha A}{V_{\text{cryst}}} \Delta T_{\text{cake}} - \dot{D} \Delta h^{LV} \quad (1.62)$$

1.4.1.1 Nusselt Correlations

In cases in which the heat transfer coefficient α in convective heat transfer is unknown, correlations towards the determination usually involve the dimensionless *Nusselt-number* which correlates convective heat transfer to conductive heat transfer [48]. In the case of heat transfer in porous bulk media similar to the present filter cake, it is important to adjust the determination of the *Nusselt* number to account for the porosity of the material. Hence, the significant length L used in the determination of the general *Nusselt* number is replaced by an equivalent diameter d_k which is calculated as given in Equation 1.63 to form Equation 1.64.

$$d_k = \sqrt{\frac{A_p}{\pi}} \quad (1.63)$$

$$\text{Nu} = \frac{\alpha d_k}{\lambda} \quad (1.64)$$

Common relations involve the inclusion of other dimensionless number such as the *Particle Reynolds* number Re_ψ as given in Equation 1.65 which correlates inertial to viscous forces and the *Prandtl* number Pr which is defined as the ratio of kinematic viscosity ν and heat transfer coefficient α as shown in Equation 1.66.

$$\text{Re}_\psi = \frac{v_{\text{free}} d_k}{\nu \varepsilon} \quad (1.65)$$

$$\text{Pr} = \frac{\nu}{a} \quad (1.66)$$

Combining the two dimensionless number yields the *Peclét* number according to Equation 1.67. The *Peclét* number Pe denotes the ratio of advective to diffusive transfer rate and is defined as the product of *Reynolds* and *Prandtl* number.

$$Pe = \frac{v_{\text{free}} d_k}{a \varepsilon} = \frac{d_k w_{\text{free}} \rho c_p}{\lambda \varepsilon} = Re \cdot Pr \quad (1.67)$$

In the nondimensional description of flow in porous bulks, the *Peclét* number generally represents the starting point. The model implemented in DWSIM is capable of applying multiple *Nusselt* correlations depending on the scale of the *Peclét* number. In the following, only a singular correlation for the case that $Pe = Re \cdot Pr < 500$ is outlined in more detail. A comprehensive overview of all implemented correlations is given in section 1.4.1.1.

Since heat transfer from fluid to a porous bulk bed is enhanced compared to just that of a singular sphere, the given correlations assumes, that this improvement can be implemented by the determination of the *Nusselt* number of a single sphere combined with a form factor f_a according to Equation 1.69. It is important to note, that the form factor deviates from that seen in crystallization and solid-liquid separation modeling. While experimentally determined parameters for f_a exist for structured packings, the description for bulks of spheres is subject to further research. Hence, this work assumes the correlation for a bulk of monodisperse spheres to account for f_a . [49]

$$Nu = f_a \cdot Nu_{\text{single sphere}} = (1 + 1.5 \cdot (1 - \varepsilon)) \cdot Nu_{\text{single sphere}} \quad (1.68)$$

Here, the single sphere *Nusselt* number is constituted of a laminar as well as a turbulent term which is due to general turbulence theory. In the vicinity of a wall - or in this case that of a particle - a viscous sublayer is formed. Here, turbulent momentum transfers are negligible since velocity fluctuations are blocked by the wall [45, 50]. Thus, it can be assumed that transport phenomena are strictly laminar [51].

However, with increasing distance from the wall the viscous, laminar sublayer is abandoned and a transient buffer layer occurs. Here, laminar phenomena from the laminar layer as well as turbulent phenomena from the purely turbulent region interfere with each other [45, 50]. Due to the comparatively greater size of the transient layer, heat transfer occurs predominantly compared to the viscous sublayer [51]. Therefore, the overall *Nusselt* model necessitates the combination of a laminar and a turbulent term as shown in Equation 1.69.

$$Nu_{\text{single sphere}} = 2 + \sqrt{Nu_{\text{lam}}^2 + Nu_{\text{turb}}^2} \quad (1.69)$$

The calculation of the laminar and turbulent *Nusselt* numbers is performed via empirical correlations involving the *Reynolds* and *Prandtl* number which ensures comparability among a range of systems. [49]

$$\text{Nu}_{\text{lam}} = 0.664 \cdot \sqrt{\text{Re}_\psi} \cdot \sqrt[3]{\text{Pr}} \quad (1.70)$$

$$\text{Nu}_{\text{turb}} = \frac{0.037 \text{Re}_\psi^{0.8} \text{Pr}}{1 + 2.443 \text{Re}_\psi^{-0.1} \cdot (\text{Pr}^{2/3} - 1)} \quad (1.71)$$

After calculating all the required singular components of the overall *Nusselt* number, all parameters except for the heat transfer coefficient α are known. By rearranging Equation 1.64 it is therefore possible to obtain values for α .

Further Nusselt Correlations

If $1 < \text{Pe} < 100$ [44]:

$$\text{Nu}_p = \frac{\phi_s(1 - \psi)}{6(1 - \varepsilon)\zeta} \cdot \frac{D_p c_p G_0}{k_f} \quad (1.72)$$

$$\psi = \frac{\vartheta_{\text{out}} - \vartheta_{\text{cake}}}{\vartheta_{\text{in}} - \vartheta_{\text{cake}}} = \frac{c_{\text{out}} - c_{\text{cake}}^*}{c_{\text{in}} - c_{\text{cake}}^*} \quad (1.73)$$

Assuming that no significant change in temperature or change in concentration of the drying air occurs over the length of the filter cake occurs; i.e. $(1 - \psi) \approx 1$:

$$\text{Nu}_p = \frac{h_p D_p}{k_f} = \frac{\phi}{6(1 - \varepsilon)\zeta} \cdot \text{Pe}_p \quad (1.74)$$

If $100 \leq \text{Pe} < 500$ [52]:

$$\text{Nu} = (1 + 1.5(1 - \varepsilon)) \cdot \text{Nu}_{\text{single sphere}} \quad (1.75)$$

$$\text{Nu}_{\text{single sphere}} = 2 + F \cdot \frac{\sqrt{\frac{\text{Pe}}{\sqrt[6]{\text{Pr}}}}}{\varepsilon} \quad (1.76)$$

If the flow is laminar i.e. $\text{Re} \leq 10^3$ [51]:

$$F = 0.664 \quad (1.77)$$

If the flow is turbulent [52]:

$$F = 0.664 \cdot \sqrt{1 + \left(\frac{0.0557 \cdot \left(\frac{\text{Re}}{\varepsilon} \right)^{0.3} \cdot \text{Pr}^{2/3}}{1 + 2.44(\text{Pr}^{2/3} - 1) \left(\frac{\text{Re}}{\varepsilon} \right)^{-0.1}} \right)^2} \quad (1.78)$$

1.4.1.2 Discretization

While the differential equations for cake temperature and cake loading are solved by the integrated *Runge-Kutta* algorithm of 4th order in DWSIM, the PDE for air temperature and air loading need to be discretized in order to employ the solver. Therefore, the filter cake has been geometrically discretized into n equidistant finite intervals as shown schematically in Figure 1.13. While the flow rate of air through the filter cake is assumed to remain constant over the entire length, both the loading and the temperature of the air change over length.

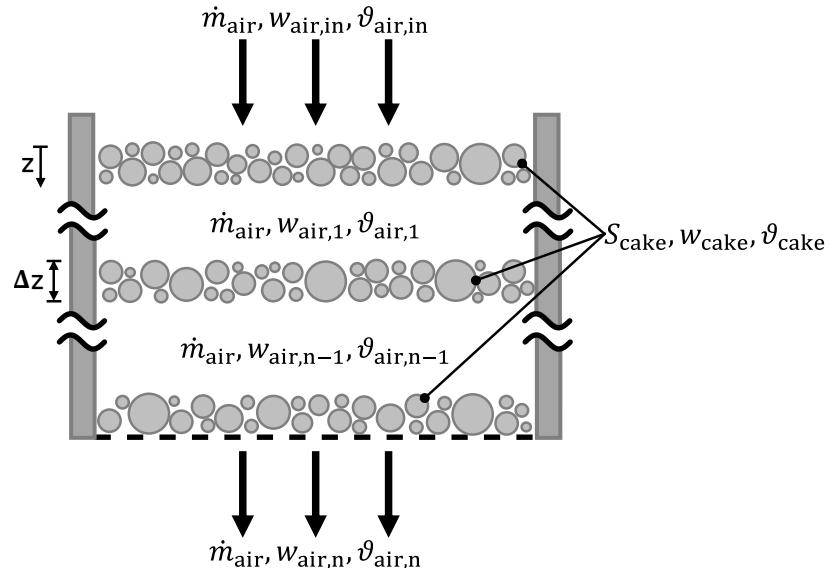


Figure 1.13: Schematic depiction of the discretized filter cake.

Since the filter cake features a small height of only few milimeters to centimeters, the *forward Euler method* is used to discretize the filter cake [53]. As shown in Equation 1.79 and Equation 1.80, the differentials $\frac{dw}{dz}$ and $\frac{dT}{dz}$ respectively are assumed to be equal to the first order finite difference. Albeit erroneous, the overall small scale of the systems allows for this simplification while leaving only a negligible truncation error $\mathcal{O}(\Delta z)$.

$$\frac{dw}{dz} \approx \frac{w^{i+1} - w^i}{\Delta z} + \mathcal{O}(\Delta z) \quad (1.79)$$

$$\frac{dT}{dz} \approx \frac{T^{i+1} - T^i}{\Delta z} + \mathcal{O}(\Delta z) \quad (1.80)$$

The implementation of this simple integration scheme transforms the PDE to a set of n ordinary differential equations which are subsequently solved by the internal DWSIM solver.

1.4.2 Efficiency Function

As introduced in section 1.4.1 the model equation for the convective drying rate has been extended by an empirical efficiency function $\Lambda(w_{\text{cake}})$ which accounts for shortcomings of the model in the second and third drying section. The course of this function has been divided into three general sections as shown in Equation 1.81.

$$\Lambda(w_{\text{cake}}) = \begin{cases} 1 & w_{\text{cake}} \geq w_{\text{cake}}^{1 \rightarrow 2} \\ c_1 \cdot w_{\text{cake}}^2 + c_2 & w_{\text{cake}}^{2 \rightarrow 3} < w_{\text{cake}} < w_{\text{cake}}^{1 \rightarrow 2} \\ 0 & w_{\text{cake}} \leq w_{\text{cake}}^{2 \rightarrow 3} \end{cases} \quad (1.81)$$

It is assumed that the overall drying rate in the first drying interval equals the aforementioned convective drying rate. Hence, in the first interval Λ assumes a value of one. In contrast, the drying rate in the third drying interval is assumed to be negligible since w_{cake} decreases asymptotically until the theoretical minimum loading as determined by the hygroscopic residual moisture $RM_{\text{hygroscopic}}$ is reached.

The drying rate in the second interval ideally assumes a parabolic shape. Therefore an equation of the form $a \cdot x^2 + b$ is utilized here. In order to fit the equation as well as the threshold loadings $w_{\text{cake}}^{2 \rightarrow 3}$ and $w_{\text{cake}}^{1 \rightarrow 2}$ that denote the transitions between drying sections, experimental data is used. While $w_{\text{cake}}^{2 \rightarrow 3}$ can be determined from the available data, $w_{\text{cake}}^{1 \rightarrow 2}$ is determined by calculating the intersect between the fitted curve and the baseline given in the convective drying rate. Finally, the obtained correlations are normalized to the baseline which yields the overall efficiency function as:

$$\Lambda(w_{\text{cake}}) = \begin{cases} 1 & w_{\text{cake}} \geq 0.23287 \\ 18.544 \cdot w_{\text{cake}}^2 - 0.00562 & 0.0174 < w_{\text{cake}} < 0.23287 \\ 0 & w_{\text{cake}} \leq 0.0174 \end{cases} \quad (1.82)$$

1.4.3 Overall Model Inputs and Outputs

Similar to the previous unit operations in the process chain, the drying model has been implemented in DWSIM as a custom unit operation. The inputs of the models can be classified into four categories which include substance data, crystallizer configuration, model parameters and output from previous unit operations. Schematically, this is shown in Figure 1.14.

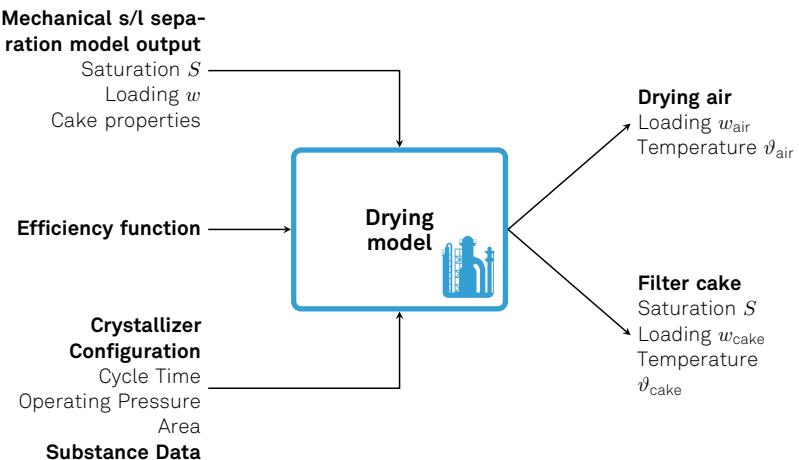


Figure 1.14: Schematic overview of the most important input and output parameters passed to and from the drying model.

The substance data is mainly called from the internal DWSIM database. The only exception to this rule are the density of the crystals and the heat capacity of the crystals. The crystallizer configuration constitutes information on operating conditions including the cycle time, the drying time, drying temperatures, pressure of the drying gas and the filtration area.

As discussed in section 1.4.2, the corresponding model equation is calculated beforehand and then passed to the drying model. Last but not least, the drying model calls data calculated in previous unit operations. This includes starting saturation, the loading of the cake as well as cake properties such as porosity and cake height. By applying the underlying model equations as discussed before, the systems outputs the course of cake loading, saturation and temperature as well as air loading and air temperature over time.

1.5 System Architecture and Graphical User Interface

In order to fully utilize the scripts developed around the previously introduced model equations, a graphical user interface (GUI) was developed using the PyQt5 Python package [54]. The resulting GUI can be classified into the input window which allows the user to enter process data, custom crystal size distribution, allows access to the fitting scripts while displaying a live-output of the results of the fitting scripts as well as customization of the temperature profile. A screenshot of the main window is given in Figure 1.15 while images of the remaining functions of the window are given in chapter 1.5.

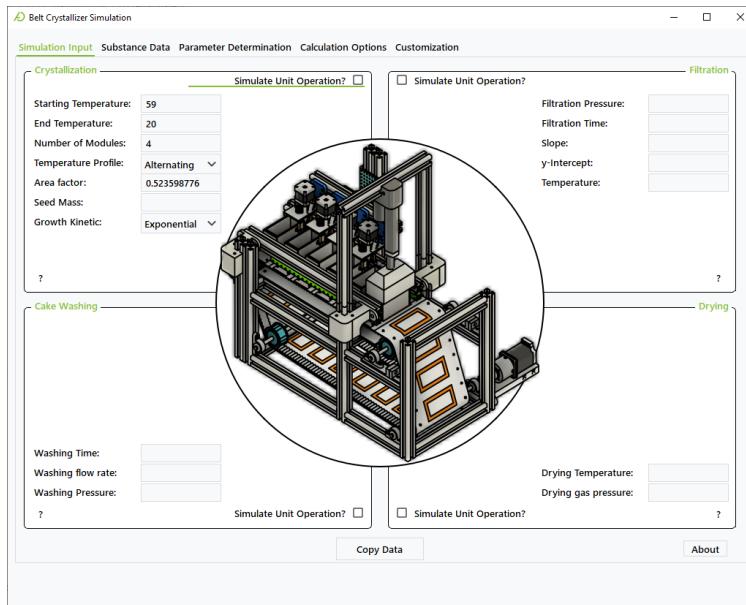


Figure 1.15: Screenshot of the main window of the programmed GUI. Here, the user can enter data for the respective unit operations. Via the tabs at the top of the window additional functionalities of the GUI can be accessed.

Furthermore, the GUI is able to send requests to WolframAlpha via the WolframAlpha Simple API_{pr} [55]. Although WolframAlpha is not all-encompassing with regards to the data needed for the models, this integration offers an additional, simple-to-use opportunity to gain data for the crystallized compound which either cannot or should not be called from the DWSIM database as discussed later on.

The second component of the GUI is a window displaying results of the simulation. After a simulation started from the main window of the GUI is finished, the result window is called automatically and display output data from the respective unit operation.

All in all, this leads to the overall system architecture shown in Figure 1.16. The main window as well as the results window can be classified as the frontend with which the user interacts. When a simulation is started, the main window reads the user inputs and passes them to a JavaScript Object Notation (JSON) file. Simultaneously, for each simulated unit operation is called which accesses the DWSIM API_{pr} and starts a simulation via the Automation3 class [5]. As soon as the simulation is finished, the data is written to an Excel file and the Python script for starting the simulation is finished. This in turn causes the main window to start the script for the results window which opens and displays data.

If the fitting algorithm is called from the main window, a secondary Python script containing the respective algorithm is called. This additional script is capable of modifying the model parameters in the JSON configuration file, can initialize simulations and reads the output from the Excel file. While all the programming logic regarding the optimization algorithm is happening in a secondary script, the respective output for each iteration is printed by the

script which is subsequently read by the main window where it is plotted in a graph which plots the data and updates with each iteration.

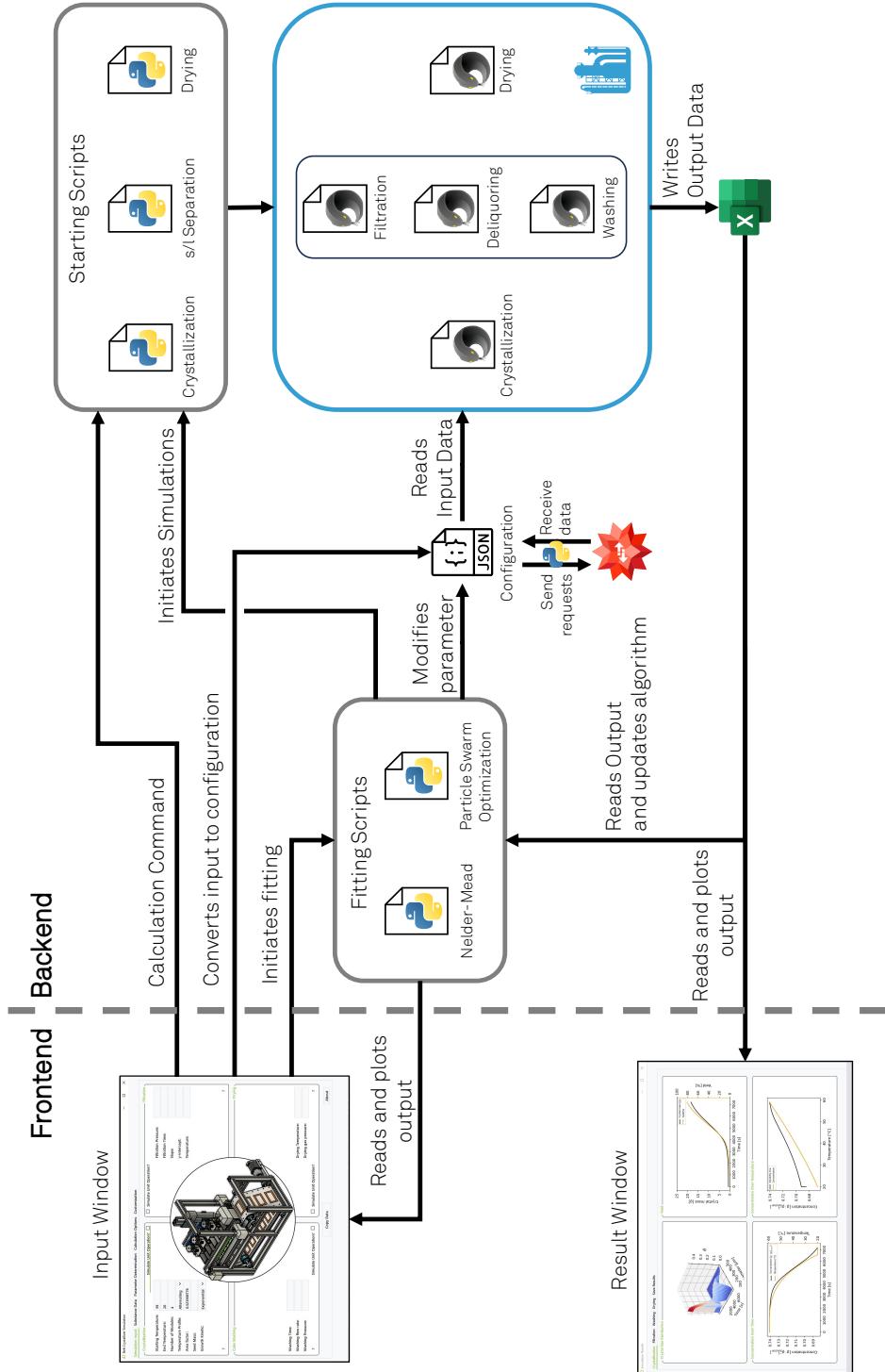


Figure 1.16: Schematic overview of the overall model structure. Including the unit operations as IronPython scripts contained in DWSIM, the scripts responsible for initiating simulations and parameter fitting scripts on the backend of the system as well as main GUI and results window on the frontend of the system.

Bibliography

- (1) Höving, S.; Oldach, B.; Kockmann, N. Cooling Crystallization with Complex Temperature Profiles on a Quasi-Continuous and Modular Plant. *Processes* **2022**, *10*, 1047, DOI: [10.3390/pr10061047](https://doi.org/10.3390/pr10061047).
- (2) Belaud, J.-P.; Pons, M., Open Software Architecture For Process Simulation: The Current Status of CAPE-OPEN Standard In *European Symposium on Computer Aided Process Engineering-12, 35th European Symposium of the Working Party on Computer Aided Process Engineering*; Computer Aided Chemical Engineering, Vol. 10; Elsevier: 2002, pp 847–852, DOI: [10.1016/S1570-7946\(02\)80169-9](https://doi.org/10.1016/S1570-7946(02)80169-9).
- (3) van Baten, J.; Pons, M. CAPE-OPEN: Interoperability in Industrial Flowsheet Simulation Software. *Chemie Ingenieur Technik* **2014**, *86*, 1052–1064, DOI: [10.1002/cite.201400009](https://doi.org/10.1002/cite.201400009).
- (4) Wagner Oliveira de Medeiros, Daniel DWSIM - Open Source Chemical Process Simulator - User Guide: Version 8.4.0, ed. by DWSIM, 2023.
- (5) Wagner Oliveira de Medeiros, Daniel DWSIM Class Library Documentation, 16.11.2023, https://dwsim.org/api_help/html/R_Project_DWSIM_Class_Library_Documentation.htm (accessed Nov. 16, 2023).
- (6) Harris, C. R. et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362, DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- (7) Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* **2020**, *17*, 261–272, DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).
- (8) Wagner Oliveira de Medeiros, Daniel DWSIM - Process Simulation, Modeling and Optimization: Technical Manual, 2016.
- (9) CO-LaN consortium Thermodynamic and Physical Properties, 2011, https://www.colan.org/wp-content/uploads/2015/05/CO_Thermo_1.1_Specification_311.pdf.
- (10) Randolph, A. D.; Larson, M. A., *Theory of particulate processes: Analysis and techniques of continuous crystallization*, 2. ed.; Academic Pr: San Diego, 1988.
- (11) *Crystallization technology handbook*, 2nd ed., rev. and expanded.; Mersmann, A., Ed.; Marcel Dekker: New York, 2001.
- (12) *Handbook of Industrial Crystallization*; Myerson, A. S., Erdemir, D., Lee, A. Y., Eds.; Cambridge University Press: 2019, DOI: [10.1017/9781139026949](https://doi.org/10.1017/9781139026949).
- (13) Koren, B., A Robust Upwind Discretization Method for Advection, Diffusion and Source Terms In *Numerical methods for advection-diffusion problems*, Vreugdenhil, C. B., Koren, B., Eds.; Notes on numerical fluid mechanics and multidisciplinary design; Vieweg: Braunschweig and Wiesbaden, 1993, pp 117–138.
- (14) Qamar, S.; Elsner, M. P.; Angelov, I. A.; Warnecke, G.; Seidel-Morgenstern, A. A comparative study of high resolution schemes for solving population balances in crystallization. *Computers & Chemical Engineering* **2006**, *30*, 1119–1131, DOI: [10.1016/j.compchemeng.2006.02.012](https://doi.org/10.1016/j.compchemeng.2006.02.012).

- (15) Hounslow, M. J.; Ryall, R. L.; Marshall, V. R. A discretized population balance for nucleation, growth, and aggregation. *AIChE Journal* **1988**, 34, 1821–1832, DOI: 10.1002/aic.690341108.
- (16) Bensouissi, A.; Roge, B.; Mathlouthi, M. Effect of conformation and water interactions of sucrose, maltitol, mannitol and xylitol on their metastable zone width and ease of nucleation. *Food Chemistry* **2010**, 122, 443–446, DOI: 10.1016/j.foodchem.2009.03.075.
- (17) De Santiago-Castillo, J. A. DotNumerics, 16.11.2023, <http://www.dotnumerics.com/> (accessed Nov. 16, 2023).
- (18) Kordylla, A.; Krawczyk, T.; Tumakaka, F.; Schembecker, G. Modeling ultrasound-induced nucleation during cooling crystallization. *Chemical Engineering Science* **2009**, 64, 1635–1642, DOI: 10.1016/j.ces.2008.12.030.
- (19) Omar, H. M.; Rohani, S. Crystal Population Balance Formulation and Solution Methods: A Review. *Crystal Growth & Design* **2017**, 17, 4028–4041, DOI: 10.1021/acs.cgd.7b00645.
- (20) Temmel, E.; Eisenschmidt, H.; Lorenz, H.; Sundmacher, K.; Seidel-Morgenstern, A. A Short-Cut Method for the Quantification of Crystallization Kinetics. 1. Method Development. *Crystal Growth & Design* **2016**, 16, 6743–6755, DOI: 10.1021/acs.cgd.6b00787.
- (21) Yokota, M.; Sato, A.; Kubota, N. A simple method for evaluating kinetic parameters in non-isothermal batch crystallization. *Chemical Engineering Science* **2000**, 55, 717–722, DOI: 10.1016/S0009-2509(99)00346-2.
- (22) Bramley, A. S.; Hounslow, M. J.; Ryall, R. L. Aggregation during Precipitation from Solution: A Method for Extracting Rates from Experimental Data. *Journal of Colloid and Interface Science* **1996**, 183, 155–165, DOI: 10.1006/jcis.1996.0530.
- (23) Faria, N.; de Azevedo, S. F.; Rocha, F. A.; Pons, M. N. Modelling agglomeration degree in sucrose crystallisation. *Chemical Engineering and Processing: Process Intensification* **2008**, 47, 1666–1677, DOI: 10.1016/j.cep.2007.09.008.
- (24) Agrawal, S. G.; Paterson, A. H. J. Secondary Nucleation: Mechanisms and Models. *Chemical Engineering Communications* **2015**, 202, 698–706, DOI: 10.1080/00986445.2014.969369.
- (25) Garside, J. Industrial crystallization from solution. *Chemical Engineering Science* **1985**, 40, 3–26, DOI: 10.1016/0009-2509(85)85043-0.
- (26) Xu, S.; Hou, Z.; Chuai, X.; Wang, Y. Overview of Secondary Nucleation: From Fundamentals to Application. *Industrial & Engineering Chemistry Research* **2020**, 59, 18335–18356, DOI: 10.1021/acs.iecr.0c03304.
- (27) Temmel, E.; Eicke, M.; Lorenz, H.; Seidel-Morgenstern, A. A Short-Cut Method for the Quantification of Crystallization Kinetics. 2. Experimental Application. *Crystal Growth & Design* **2016**, 16, 6756–6768, DOI: 10.1021/acs.cgd.6b00789.
- (28) Höving, S.; Schmidt, T.; Peters, M.; Lapainis, H.; Kockmann, N. Small-Scale Solids Production Plant with Cooling Crystallization, Washing, and Drying in a Modular, Continuous Plant. *Processes* **2023**, 11, 2457, DOI: 10.3390/pr11082457.
- (29) *Handbook of Metaheuristics*, 3rd ed. 2019; Gendreau, M., Potvin, J.-Y., Eds.; International Series in Operations Research & Management Science, Vol. 272; Springer International Publishing and Imprint: Springer: Cham, 2019.

- (30) Kennedy, J.; Eberhart, R. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE: 1995, pp 1942–1948, DOI: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- (31) Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: an overview. *Soft Computing* **2018**, *22*, 387–408, DOI: [10.1007/s00500-016-2474-6](https://doi.org/10.1007/s00500-016-2474-6).
- (32) Nelder, J. A.; Mead, R. A Simplex Method for Function Minimization. *The Computer Journal* **1965**, *7*, 308–313, DOI: [10.1093/comjnl/7.4.308](https://doi.org/10.1093/comjnl/7.4.308).
- (33) Dantzig, G. B., *Linear Programming: 1: Introduction*; SpringerLink Bücher; George B. Dantzig and Mukund N. Thapa: New York, NY, 1997, DOI: [10.1007/b97672](https://doi.org/10.1007/b97672).
- (34) Ullrich, C.; Bodmer, T.; Hübner, C.; Kempa, P. B.; Tsotsas, E.; Eschner, A.; Kasperek, G.; Ochs, F.; Spitzner, M. H., D6 Stoffwerte von Feststoffen In *VDI-Wärmeatlas*, VDI e.V., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013.
- (35) Stieß, M., *Mechanische Verfahrenstechnik*; Springer-Lehrbuch; Springer: Berlin, 1994.
- (36) Wakeman, R. J.; Tarleton, E. S.; Tarleton, E. S., *Solid/liquid separation: Principles of industrial filtration*, 1. ed.; Elsevier: Oxford, 2005.
- (37) Sucrose: *Properties and applications*, 1. ed.; Mathlouthi, M., Reiser, P., Eds.; Blackie Academic & Professional: London, 1995.
- (38) Bourcier, D.; Féraud, J. P.; Colson, D.; Mandrick, K.; Ode, D.; Brackx, E.; Puel, F. Influence of particle size and shape properties on cake resistance and compressibility during pressure filtration. *Chemical Engineering Science* **2016**, *144*, 176–187, DOI: [10.1016/j.ces.2016.01.023](https://doi.org/10.1016/j.ces.2016.01.023).
- (39) Brückner, A.; Sprott, T.; Peuker, U. A.; Hoffner, B. Durchströmungswäsche gesättigter und untersättigter Filterkuchen. *Chemie Ingenieur Technik* **2021**, *93*, 1238–1246, DOI: [10.1002/cite.202000250](https://doi.org/10.1002/cite.202000250).
- (40) Heindl, A., *Praxisbuch Bandtrocknung: Grundlagen, Anwendung, Berechnung*; Springer Vieweg: Berlin and Heidelberg, 2016.
- (41) Lee, H.; Carr, W. W.; Beckham, H. W.; Leisen, J. A model of through-air drying of tufted textile materials. *International Journal of Heat and Mass Transfer* **2002**, *45*, 357–366, DOI: [10.1016/S0017-9310\(01\)00130-2](https://doi.org/10.1016/S0017-9310(01)00130-2).
- (42) Ryan, M.; Modak, A.; Zuo, H.; Ramaswamy, S.; Worry, G. Through Air Drying. *Drying Technology* **2003**, *21*, 719–734, DOI: [10.1081/DRT-120019059](https://doi.org/10.1081/DRT-120019059).
- (43) Castro, A. M.; Mayorga, E. Y.; Moreno, F. L. Mathematical modelling of convective drying of fruits: A review. *Journal of Food Engineering* **2018**, *223*, 152–167, DOI: [10.1016/j.jfoodeng.2017.12.012](https://doi.org/10.1016/j.jfoodeng.2017.12.012).
- (44) Daizo, K.; Motoyuki, S. Particle-to-fluid heat and mass transfer in packed beds of fine particles. *International Journal of Heat and Mass Transfer* **1967**, *10*, 845–852, DOI: [10.1016/0017-9310\(67\)90064-6](https://doi.org/10.1016/0017-9310(67)90064-6).
- (45) Zierep, J.; Bühler, K., *Grundzüge der Strömungslehre: Grundlagen, Statik und Dynamik der Fluide : mit 202 Abbildungen und zahlreichen Übungen*, 11., überarbeitete und erweiterte Auflage; Lehrbuch; Springer Vieweg: Wiesbaden, 2018.
- (46) Soltanzadeh, H. The Extended Navier-Stokes Equations and Their Application to the Lay-Out of Diffusion Dryers, Dissertation, Erlangen: Friedrich Alexander Universität Erlangen Nürnberg, 2018.
- (47) Xue, L.; Guo, X.; Chen, H., *Fluid flow in porous media: Fundamentals and applications*, First edition; World Scientific: New Jersey, 2021.

- (48) Christen, D. S., *Praxiswissen der chemischen Verfahrenstechnik: Handbuch Für Chemiker Und Verfahrensingenieure*, 2nd ed.; VDI-Buch; Springer: Dordrecht, 2009, <http://gbv.eblib.com/patron/FullRecord.aspx?p=510577>.
- (49) Gnielinski, V., G9 Wärmeübertragung Partikel - Fluid in durchströmten Haufwerken In *VDI-Wärmeatlas*, VDI e.V., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013, pp 839–840.
- (50) Bailly, C.; Comte-Bellot, G., *Turbulence*, 1st ed. 2015; Experimental Fluid Mechanics; Springer International Publishing and Imprint: Springer: Cham, 2015.
- (51) Gnielinski, V. Berechnung mittlerer Wärme- und Stoffübergangskoeffizienten an laminar und turbulent überströmten Einzelkörpern mit Hilfe einer einheitlichen Gleichung. *Forschung im Ingenieurwesen* **1975**, 41, 145–153, DOI: 10.1007/BF02560793.
- (52) Martin, H. Low peclet number particle-to-fluid heat and mass transfer in packed beds. *Chemical Engineering Science* **1978**, 33, 913–919, DOI: 10.1016/0009-2509(78)85181-1.
- (53) Schiesser, W. E.; Griffiths, G. W., *A compendium of partial differential equation models: Method of lines analysis with Matlab*; Cambridge Univ. Press: Cambridge, 2009.
- (54) Riverbank Computing Limited, The Qt Company PyQt5 Reference Guide, 2022, <https://www.riverbankcomputing.com/static/Docs/PyQt5/> (accessed Nov. 27, 2023).
- (55) Wolfram Research, Inc. Wolfram|Alpha APIs: Computational Knowledge Integration, ed. by Wolfram Research, Inc., <https://products.wolframalpha.com/api> (accessed Sept. 8, 2023).
- (56) Pot, A. Industrial Sucrose Crystallization: A study on the development of crystal size distributions in continuous and batch sucrose suspension crystallizers, Dissertation, Delft: Technische Hogeschool Delft, 1983.
- (57) Herwig, H.; Moschallski, A., *Wärmeübertragung: Physikalische Grundlagen Illustrierende Beispiele Übungsaufgaben mit Musterlösungen*; Springer eBook Collection Computer Science & Engineering; Vieweg: Wiesbaden, 2006, DOI: 10.1007/978-3-8348-9070-2.
- (58) Wagner, W.; Kretzschmar, H.-J.; Span, R., D2 Stoffwerte von bedeutenden reinen Fluiden In *VDI-Wärmeatlas*, VDI e.V., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013, pp 175–136.
- (59) Kretzschmar, H.-J.; Wagner, W., *International Steam Tables: Properties of Water and Steam based on the Industrial Formulation IAPWS-IF97*, 3rd ed. 2019; Springer Vieweg: Berlin, Heidelberg, 2019, DOI: 10.1007/978-3-662-53219-5.
- (60) National Institute of Standards and Technology Entry for Water in NIST Chemistry WebBook, <https://webbook.nist.gov/cgi/cbook.cgi?Name=water&Units=SI>.
- (61) Liu, C.-T.; Lindsay, W. T. Vapor pressure of deuterated water from 106 to 300 deg. *Journal of Chemical & Engineering Data* **1970**, 15, 510–513, DOI: 10.1021/je60047a015.
- (62) Bridgeman, O. C.; Aldrich, E. W. Vapor Pressure Tables for Water. *Journal of Heat Transfer* **1964**, 86, 279–286, DOI: 10.1115/1.3687121.
- (63) *CRC Handbook of Chemistry and Physics*, 104th ed.; Rumble, J. R., Ed.; (Internet Version 2023); CRC Press/Taylor & Francis: Boca Raton, FL, 2023.

Appendix

1.6 Supplementary Substance Data

Solubility Calculation [37]

$$w \left[\frac{g}{g \text{ Solution}} \right] = \frac{1}{100} \cdot 64.447 + 0.0822 \cdot \vartheta + 1.6169 \cdot 10^{-3} \cdot \vartheta^2 - 1.558 \cdot 10^{-6} \cdot \vartheta^3 - 4.63 \cdot 10^{-8} \cdot \vartheta^4 \quad (1.83)$$

Viscosity Calculation [56]

$$\log_{10}(\eta) = A + \frac{B}{\vartheta + 273.15} + \frac{C}{\vartheta + 43.15} \quad (1.84)$$

$$A = -2.038 - 13.627 \cdot y_s - 17.912 \cdot y_s^2 + 56.426 \cdot y_s^3 \quad (1.85)$$

$$B = 513.367 + 10740.329 \cdot y_s - 16781.321 \cdot y_s^2 + 14142.897 \cdot y_s^3 \quad (1.86)$$

$$C = 16.993 + 34.442 \cdot y_s + 3915.947 \cdot y_s^2 - 6839.469 \cdot y_s^3 \quad (1.87)$$

$$y_s = \frac{w}{1-w} \cdot \frac{M_{\text{water}}}{M_{\text{sucrose}}} \quad (1.88)$$

1.7 Supplementary Information: Modeling

1.7.1 Drying

1.7.1.1 Derivation of PDE used in the drying model

Filter Cake Loading

Unstationary change in the amount of liquid in the filter cake:

$$m_{\text{cr}} \frac{dw}{dt} = \varepsilon V_{\text{cake}} \rho_l \frac{S_0 - S_1}{t_1 - t_0} \quad (1.89)$$

with $w \left[\frac{\text{kg}}{\text{kg dry matter}} \right]$ and:

$$m_{\text{cr}} = (1 - \varepsilon) V_{\text{cake}} \rho_{\text{cr}} \quad (1.90)$$

$$\begin{aligned} (1 - \varepsilon) V_{\text{cake}} \rho_{\text{cr}} \frac{dw}{dt} &= \varepsilon V_{\text{cake}} \rho_l \frac{S_0 - S_1}{t_1 - t_0} \\ \Leftrightarrow (1 - \varepsilon) \rho_{\text{cr}} \frac{dw}{dt} &= \varepsilon \rho_l \frac{S_0 - S_1}{t_1 - t_0} \\ \Leftrightarrow \rho_{\text{cr}} \frac{dw}{dt} &= \frac{\varepsilon}{1 - \varepsilon} \rho_l \frac{S_0 - S_1}{t_1 - t_0} \end{aligned}$$

Using the definition of the porosity $\varepsilon = \frac{V_{\text{pores}}}{V_{\text{cake}}}$:

$$\frac{\varepsilon}{1 - \varepsilon} = \frac{\frac{V_{\text{pore}}}{V_{\text{cake}}}}{1 - \frac{V_{\text{pore}}}{V_{\text{cake}}}} = \frac{V_{\text{pore}}}{V_{\text{cake}} - V_{\text{pore}}} = \frac{V_{\text{pore}}}{V_{\text{crystal}}} \quad (1.91)$$

resulting in:

$$\rho_{\text{cr}} \frac{dw}{dt} = \frac{V_{\text{pore}}}{V_{\text{crystal}}} \rho_l \frac{S_0 - S_1}{t_1 - t_0} \quad (1.92)$$

With the cake saturation S as $S = \frac{V_l}{V_{\text{pore}}}$ under the assumption that the cake is incompressible i.e. $V_{\text{pore}} = \text{const}$:

$$\begin{aligned} \Leftrightarrow \rho_{\text{cr}} \frac{dw}{dt} &= \frac{V_{\text{pore}}}{V_{\text{cryst}}} \rho_l \frac{1}{V_{\text{pore}}} \underbrace{\frac{V_{l,0} - V_{l,1}}{t_1 - t_0}}_{\dot{V}^{L \rightarrow G}} \\ \Leftrightarrow \rho_{\text{cr}} \frac{dw}{dt} &= \frac{1}{V_{\text{cryst}}} \underbrace{\frac{m_{l,0} - m_{l,1}}{t_1 - t_0}}_{\hat{m}^{L \rightarrow gG}} \\ \frac{dw}{dt} &= \frac{1}{\rho_{\text{cryst}}} \dot{D} \text{ with } \dot{D} \left[\frac{\text{kg}}{\text{m}^3 \cdot \text{s}} \right] \end{aligned}$$

Drying Air Loading

The starting point is given in the continuity equation for porous media under the occurrence of a change in mass dm [45–47]:

$$dm = \frac{\partial(\rho\varepsilon)}{\partial t} dx dy dz dt + \left[\frac{\partial(\rho v_x)}{\partial x} + \frac{\partial(\rho v_y)}{\partial y} + \frac{\partial(\rho v_z)}{\partial z} \right] dx dy dz dt \quad (1.93)$$

Where it is assumed that the change in mass dm over a differential element $dx dy dz$ and the differential time element dt equates to the previously introduced volumetric mass change \dot{D} :

$$\dot{D} = \frac{dm}{dx dy dz dt} \quad (1.94)$$

$$\dot{D} = \frac{\partial(\rho\varepsilon)}{\partial t} + \frac{\partial(\rho v_x)}{\partial x} + \frac{\partial(\rho v_y)}{\partial y} + \frac{\partial(\rho v_z)}{\partial z} \quad (1.95)$$

Assuming that no flow in x - or y -direction occurs this simplifies to:

$$\dot{D} = \frac{\partial(\rho\varepsilon)}{\partial t} + \frac{\partial(\rho v_z)}{\partial z} \quad (1.96)$$

Taking into account the saturation of the filter cake for a single component i :

$$\dot{D}_i = \frac{\partial(\rho\varepsilon(1 - S)w_i)}{\partial t} + \frac{\partial(\rho v_z w_i)}{\partial z} \quad (1.97)$$

Assuming that: $\rho = \text{const}$, $\varepsilon = \text{const}$, $v_z = \text{const}$ and $S \approx \text{const}$ this yields the final form of the PDE with:

$$\dot{D}_i = \rho \varepsilon (1 - S) \frac{\partial w_i}{\partial t} + \rho v_z \frac{\partial w_i}{\partial z} \quad (1.98)$$

Cake Temperature

General balance for the change in temperature:

$$\frac{dT}{dt} = [\text{Inflow}] - [\text{Outflow}] + [\text{Sources}]$$

With the following sources in this case:

- heating of the filter cake
- evaporation of the liquid
- heat losses through the wall

Resulting in the following general heat balance for the filter cake [46, 47, 57]:

$$c_p m \frac{dT_{\text{cake}}}{dt} = \dot{Q}_{\text{conv}} - \dot{Q}_{\text{loss}} - \dot{Q}_{\text{evap}} \quad (1.99)$$

with $c_p m$ composed of the entries of solid, liquid and gas inside the filter cake:

$$c_p m = V_{\text{cake}} (\rho_s c_{p,s} (1 - \varepsilon) + \rho_l c_{p,l} \varepsilon S + \rho_g c_{p,g} \varepsilon (1 - S)) \quad (1.100)$$

Assuming that the overall influence of the gas on heat transfer inside the cake is neglectable compared with those of solid and liquid i.e. $c_{p,g} \ll c_{p,s}, c_{p,l}$:

$$c_p m = V_{\text{cake}} (\rho_s c_{p,s} (1 - \varepsilon) + \rho_l c_{p,l} \varepsilon S) \quad (1.101)$$

With the following expressions for the respective \dot{Q} -terms:

$$\dot{Q}_{\text{conv}} = \alpha \cdot A \cdot (T_{\text{fluid}} - T_{\text{cake}}) = \alpha \cdot A \cdot \Delta T_{\text{cake}} \quad (1.102)$$

$$\dot{Q}_{\text{loss}} = k \cdot A \cdot (T_{\text{fluid}} - T_{\infty}) = k \cdot A \cdot \Delta T_{\infty} \quad (1.103)$$

$$\dot{Q}_{\text{evap}} = \dot{m}^{L \rightarrow G} \cdot \Delta h^{LV} \quad (1.104)$$

Which results in the overall balance equation:

$$V_{\text{cake}} (\rho_s c_{p,s} (1 - \varepsilon) + \rho_l c_{p,l} \varepsilon S) \frac{dT_{\text{cake}}}{dt} = \alpha A \Delta T_{\text{cake}} - k A \Delta T_{\infty} - \dot{m}^{L \rightarrow G} \Delta h^{LV} \quad (1.105)$$

Normalizing this to the crystal volume yields:

$$\begin{aligned} \frac{V_{\text{cake}}}{V_{\text{cryst}}} (\rho_s c_{p,s} (1 - \varepsilon) + \rho_l c_{p,l} \varepsilon S) \frac{dT_{\text{cake}}}{dt} &= \frac{\alpha A}{V_{\text{cryst}}} \Delta T_{\text{cake}} - \frac{k A}{V_{\text{cryst}}} \Delta T_{\infty} - \dot{D} \Delta h^{LV} \\ \frac{1}{1 - \varepsilon} (\rho_s c_{p,s} (1 - \varepsilon) + \rho_l c_{p,l} \varepsilon S) \frac{dT_{\text{cake}}}{dt} &= \frac{\alpha A}{V_{\text{cryst}}} \Delta T_{\text{cake}} - \frac{k A}{V_{\text{cryst}}} \Delta T_{\infty} - \dot{D} \Delta h^{LV} \end{aligned}$$

Neglecting heat losses to the environment due to the low heat transfer surface coupled with the comparatively large gas velocity yields the final form of the PDE as:

$$\frac{1}{1-\varepsilon}(\rho_s c_{p,s}(1-\varepsilon) + \rho_l c_{p,l}\varepsilon S)\frac{dT_{\text{cake}}}{dt} = \frac{\alpha A}{V_{\text{cryst}}} \Delta T_{\text{cake}} - \dot{D} \Delta h^{LV} \quad (1.106)$$

Air Temperature

The starting point is given in the equation of continuity for a porous substance [45–47] combined with the two terms responsible for heat loss in the gas which are given in heat transfer to the filter cake and evaporation of the liquid from the filter cake. However, in contrast to the mass transfer equation used before, here the general form for heat transfer phenomena is assumed.

$$V_{\text{cake}} \frac{\partial(T \rho_g c_{p,g} \varepsilon (1-S))}{\partial t} + V_{\text{cake}} \nabla(\vec{u} T \rho_g c_{p,g}) = -\alpha A \Delta T_{\text{cake}} - \dot{m}_D \Delta h^{LV} \quad (1.107)$$

Assuming that $\rho = \text{const}$, $c_p = \text{const}$, $\varepsilon = \text{const}$ and $S \approx \text{const}$, changing the notation to the expanded form in x -, y - and z -direction and using the volumetric drying rate \dot{D} yields:

$$\begin{aligned} \frac{V_{\text{cake}}}{V_{\text{cryst}}} \left[\rho_g c_{p,g} \varepsilon (1-S) \frac{\partial T}{\partial t} + \rho_g c_{p,g} \left(\frac{\partial(u_x T)}{\partial x} + \frac{\partial(u_y T)}{\partial y} + \frac{\partial(u_z T)}{\partial z} \right) \right] \\ = -\frac{\alpha A}{V_{\text{cryst}}} \Delta T_{\text{cake}} - \dot{D} \Delta h^{LV} \end{aligned} \quad (1.108)$$

However, since it is assumed that the air exhibits a uniform temperature over the entire cross section in x - and y -direction this equation can be simplified to yield the final form of the PDE:

$$\frac{V_{\text{cake}}}{V_{\text{cryst}}} \rho_g c_{p,g} \varepsilon (1-S) \frac{\partial T}{\partial t} + \frac{V_{\text{cake}}}{V_{\text{cryst}}} \rho_g c_{p,g} u_z \frac{\partial T}{\partial z} = -\frac{\alpha A}{V_{\text{cryst}}} \Delta T_{\text{cake}} - \dot{D} \Delta h^{LV} \quad (1.109)$$

1.8 Supplementary Information: GUI

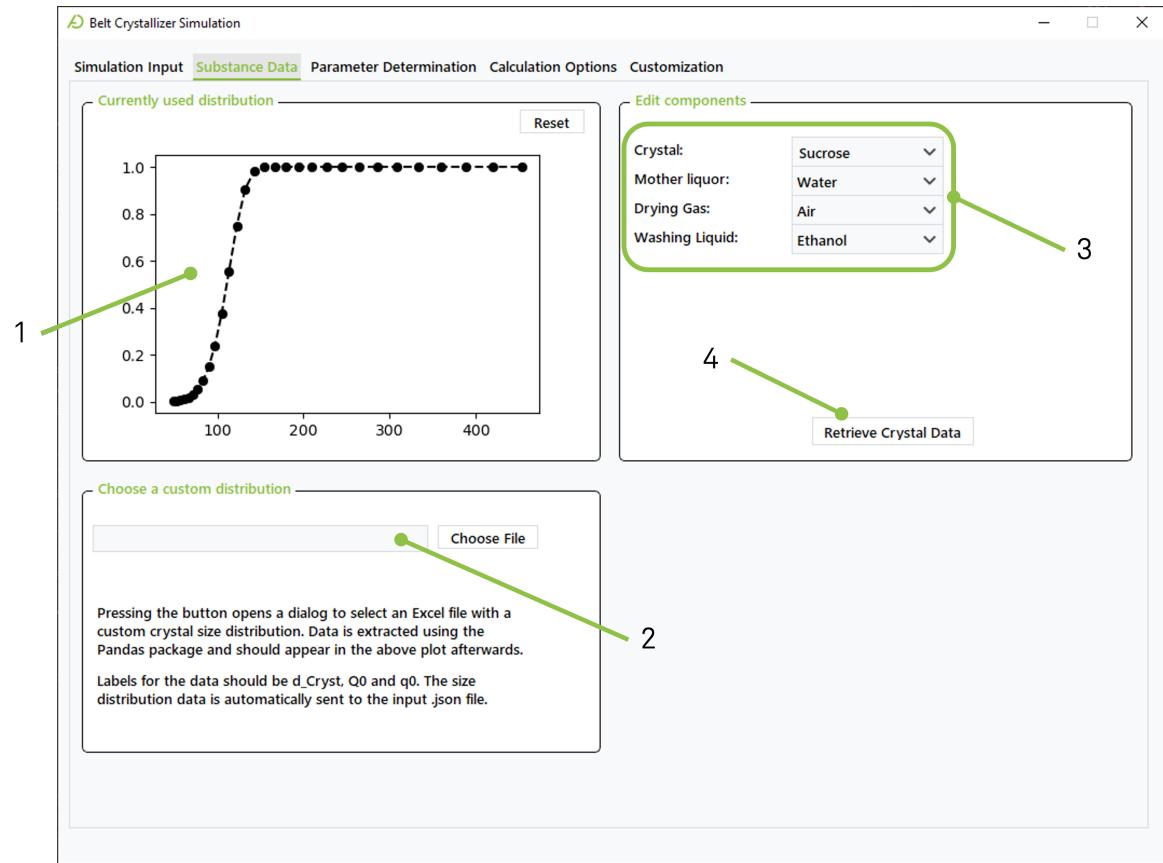


Figure 1.17: Screenshot of the tab in the GUI responsible for entering substance data. 1) displays a plot of the currently utilized Q3-distribution. 2) allows for choosing an Excel file with a custom distribution. 3) allows to customize components. By clicking on 4) a new window opens which can send requests to the WolframAlpha API_{pr} and allows user inputs for solid density, solid heat capacity and gas density.

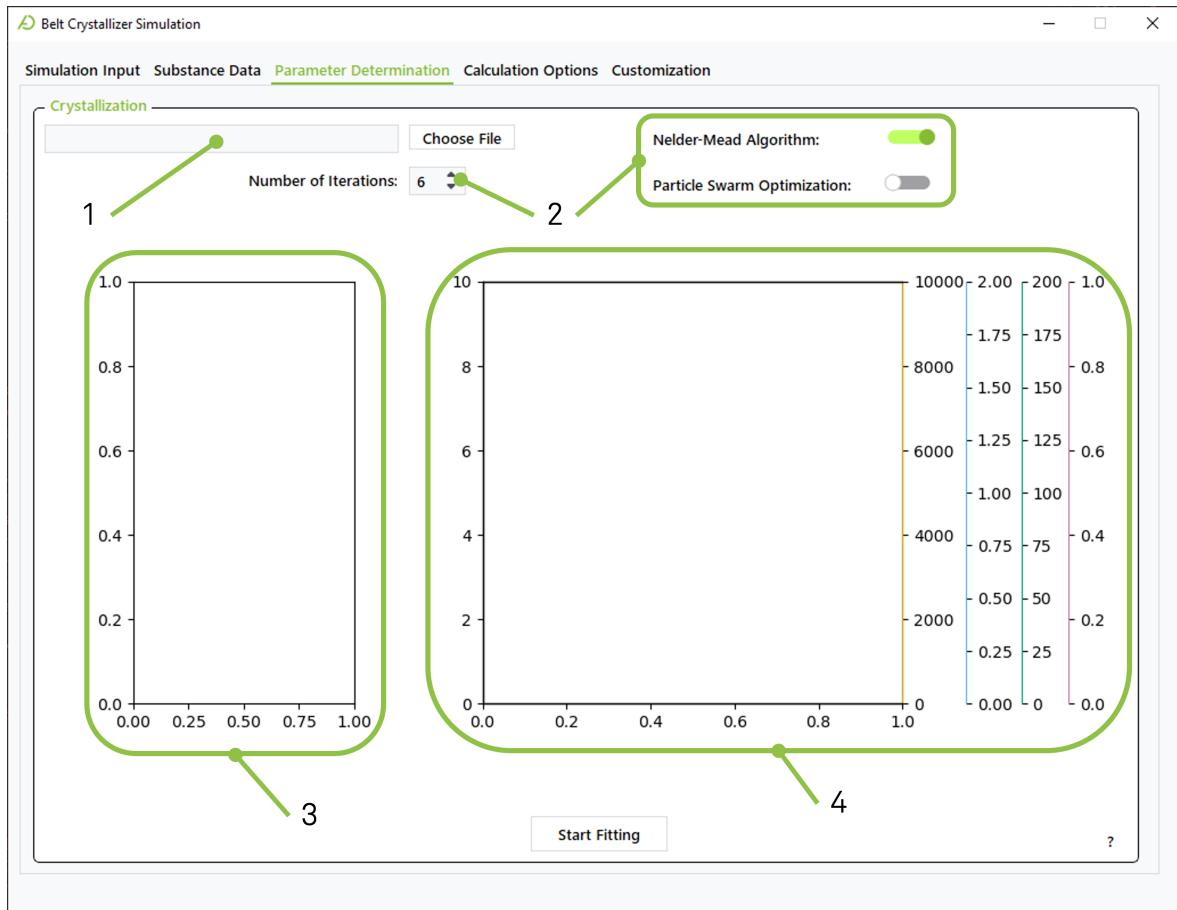


Figure 1.18: Screenshot of the tab in the GUI responsible for parameter fitting of the crystallization. 1) allows to choose a custom input file with inputs, 2) allows to choose the number of iterations and the underlying optimization algorithm, 3) displays the value of the objective function as well as the average of the objective function in real-time. 4) displays values of crystallization and nucleation parameters in real time.

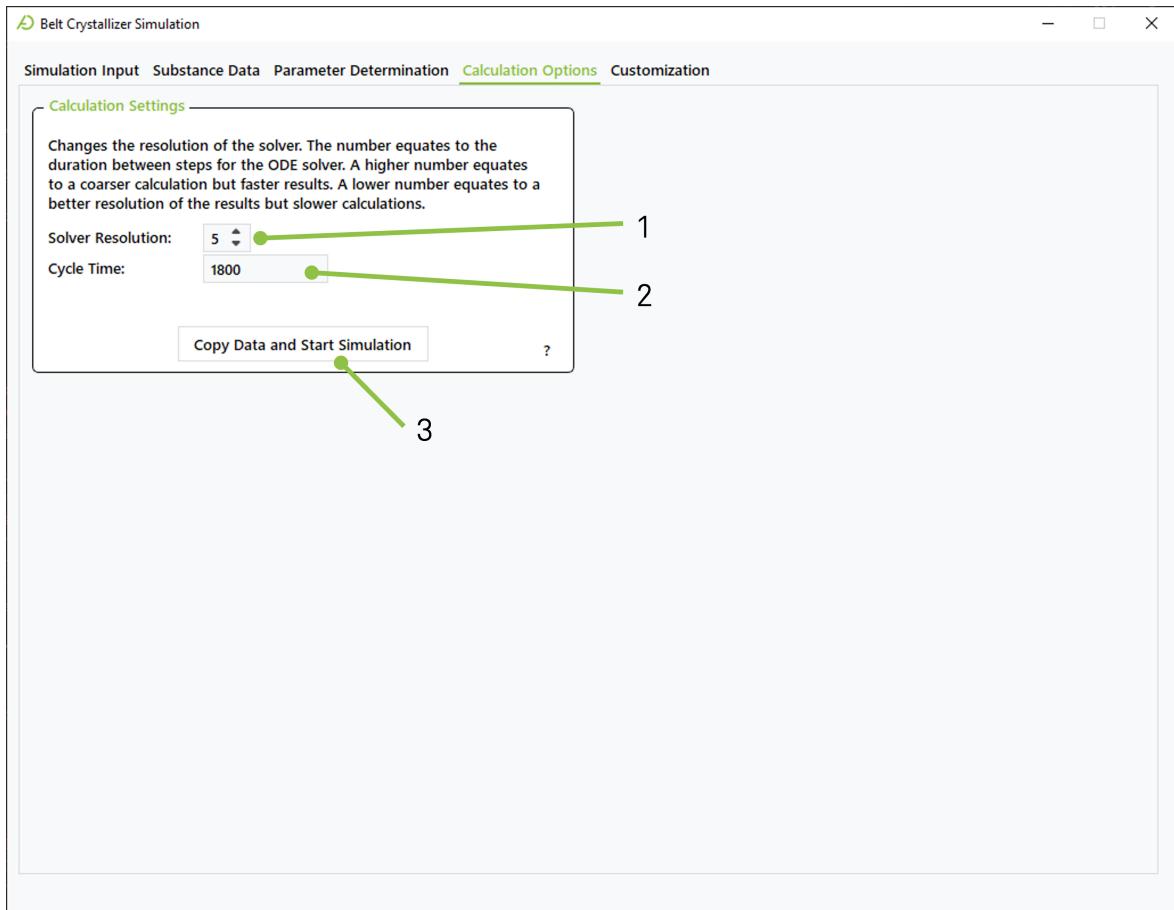


Figure 1.19: Screenshot of the tab in the GUI responsible for starting simulations. 1) allows to adjust the step width by multiples of 5. 2) allows the user to choose the cycle time. 3) copies data to the configuration JSON file and initiates simulations via external Python scripts.

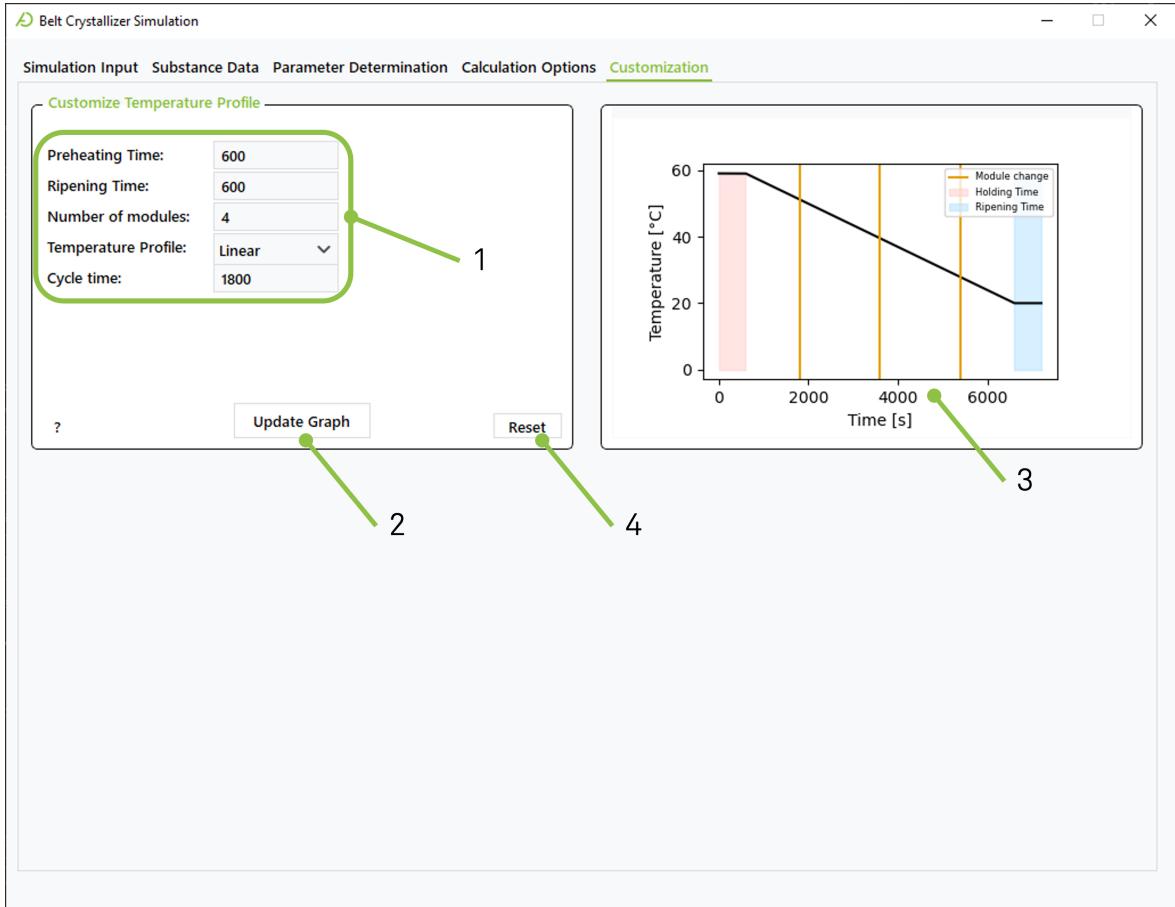


Figure 1.20: Screenshot of the tab in the GUI responsible for customizing the temperature profile. 1) allows the user to choose the preheating, ripening and cycle time as well as the number of crystallization modules and the temperature profile. 2) updates the graph shown in 3) and - in the case of the oscillating profile - calculates the required parameters A-D. 4) allows to reset the temperature profile to the one currently shown in 3). All input boxes are connected with the ones in the tabs “Simulation Input“ and “Calculation Options“ to ensure that no conflicting input is given to the configuration file.

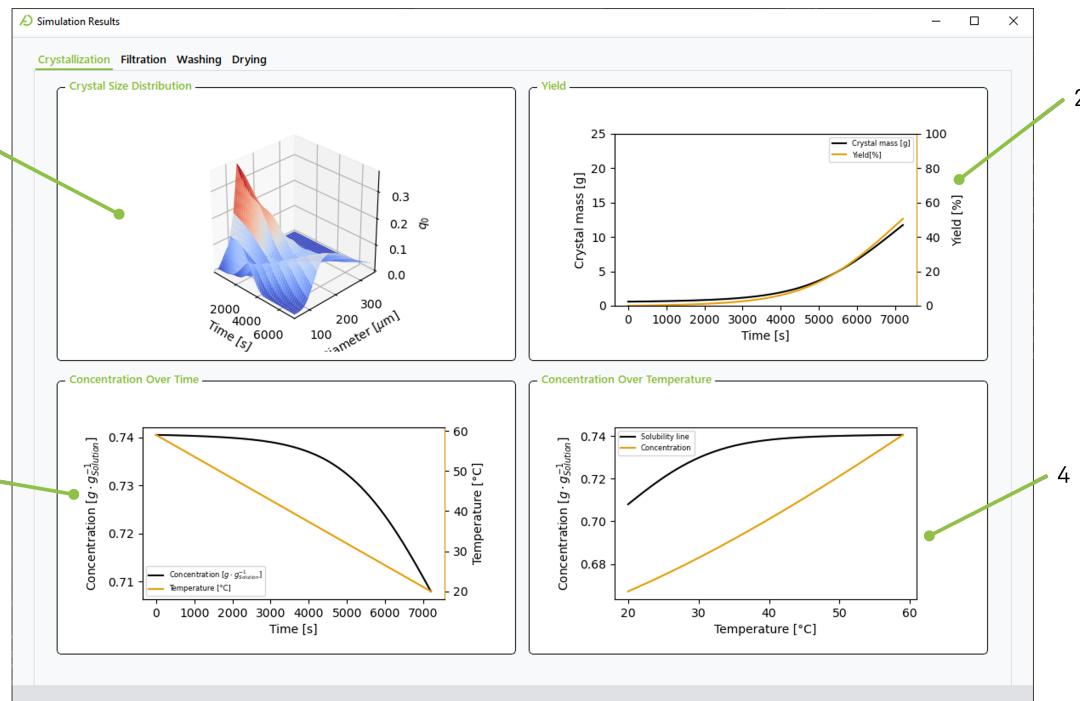


Figure 1.21: Screenshot of the tab in the GUI showing crystallization results. 1) shows the development of the CSD over time, 2) shows the development of crystal mass and yield over time. 3) shows the development of concentration and temperature over time. 4) shows the concentration and the saturation line over the temperature.

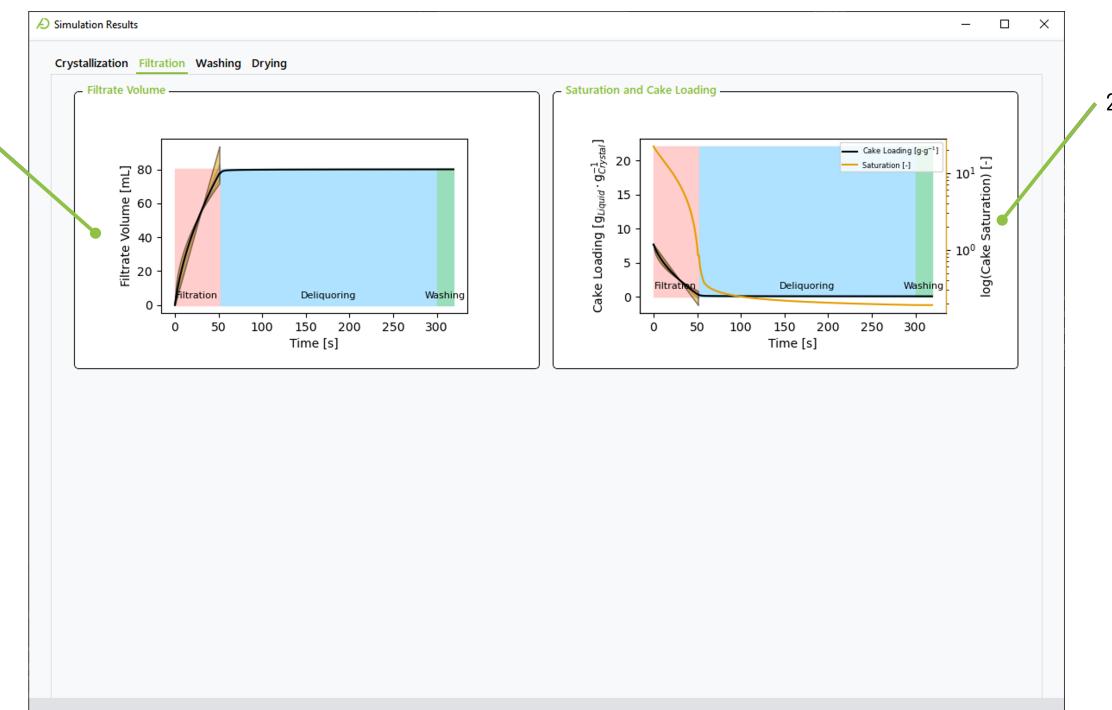


Figure 1.22: Screenshot of the tab in the GUI showing filtration results. 1) shows the cumulative volume of filtrate. 2) shows the development of saturation and cake loading over time.

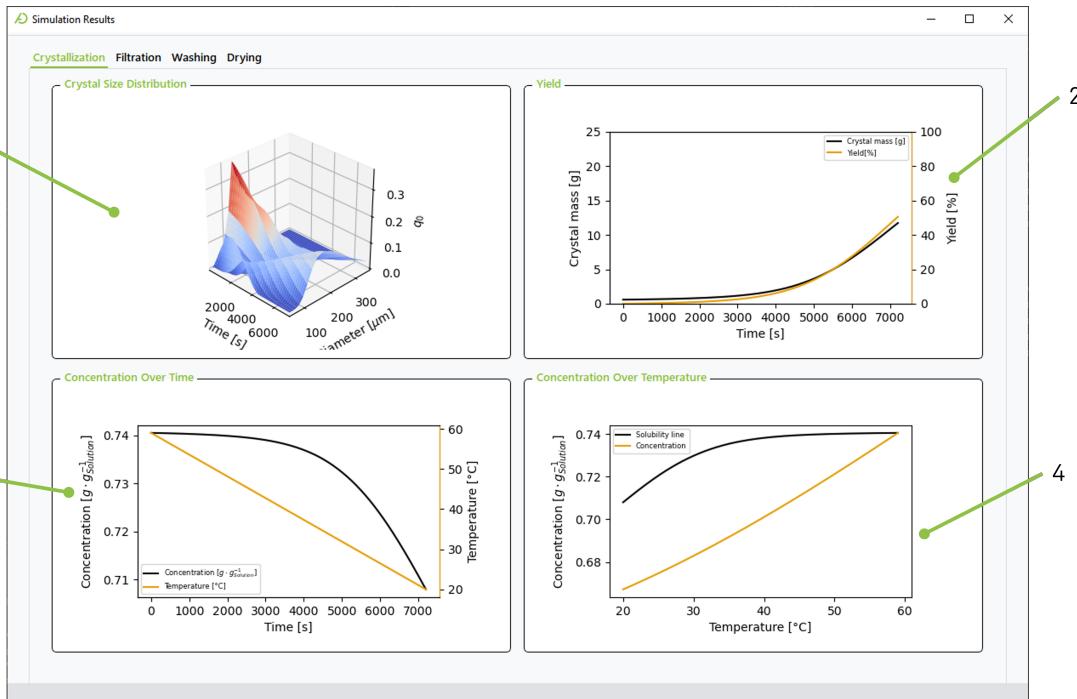


Figure 1.23: Screenshot of the tab in the GUI showing washing results. 1) shows the development of fractional removal F from the filter cake. 2) shows the development of the dimensionless loading of the wash effluent ϕ^* over time.

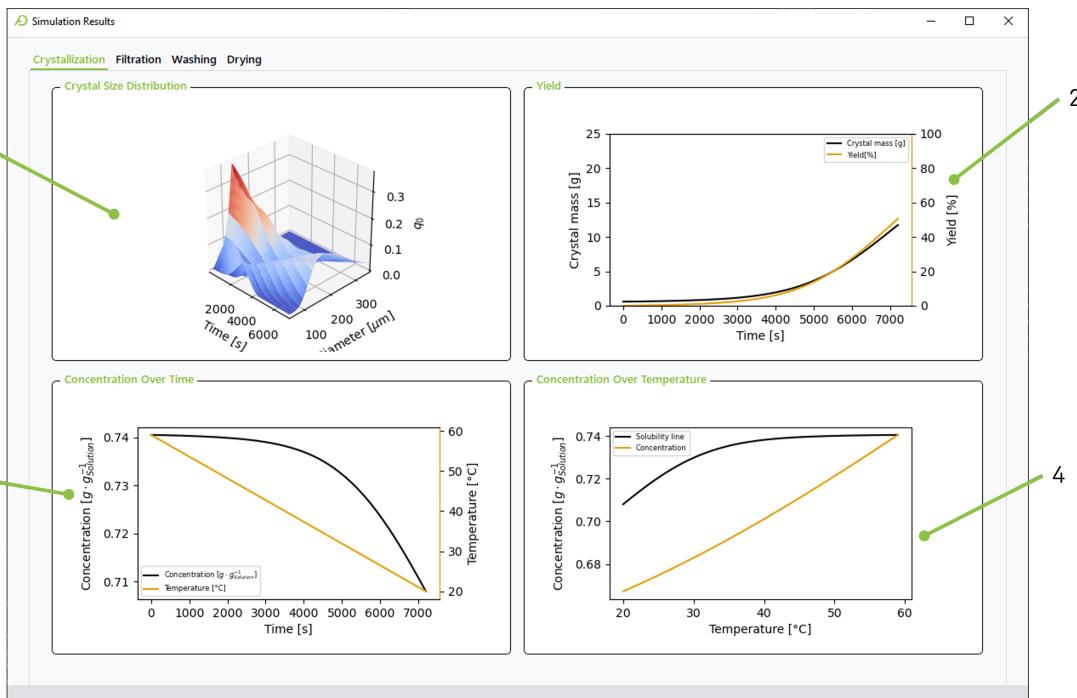


Figure 1.24: Screenshot of the tab in the GUI showing drying results. 1) shows the development of the volumetric drying rate over time. 2) shows the development of cake loading and cake temperature over time. 3) shows the development of the drying air, discretized over the filter cake over time. 4) shows the development of the drying air loading discretized over the filter cake over time.

1.9 Supplementary Information: Validation of DWSIM Database

Water - Liquid Density

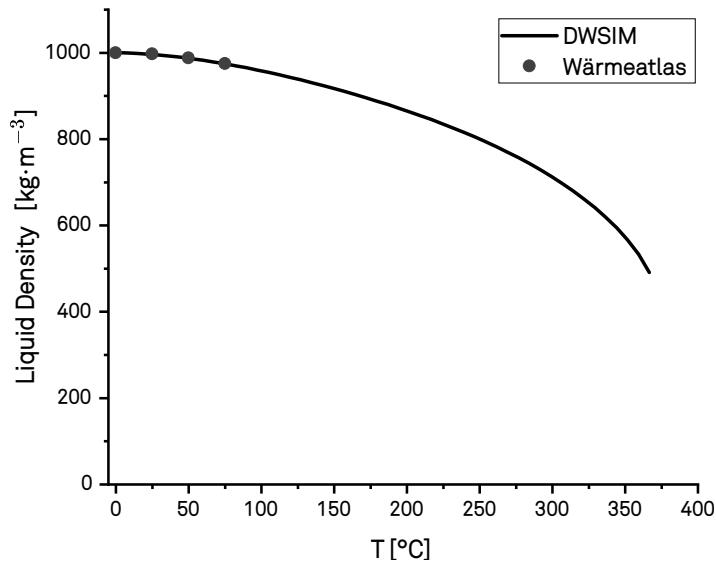


Figure 1.25: Comparision of liquid density data of water from DWSIM database with literature values according to the VDI Heat Atlas. [58]

Water - Liquid Viscosity

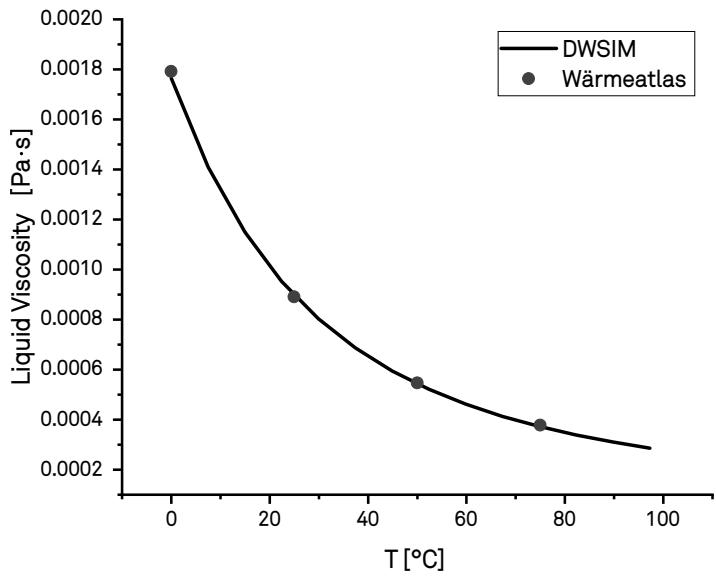


Figure 1.26: Comparision of liquid viscosity data of water from DWSIM database with literature values according to the VDI Heat Atlas. [58]

Water - Heat Capacity

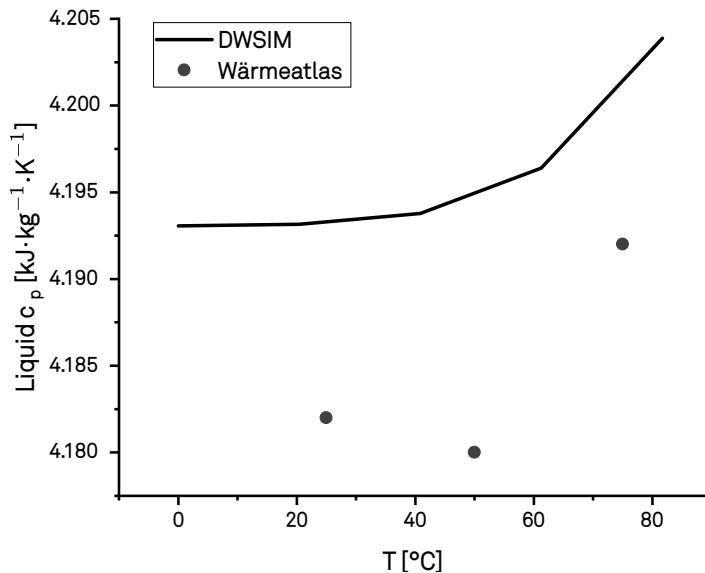


Figure 1.27: Comparision of liquid heat capacity data of water from DWSIM database with literature values according to the VDI Heat Atlas. [58]

Water - Enthalpy of Vaporization

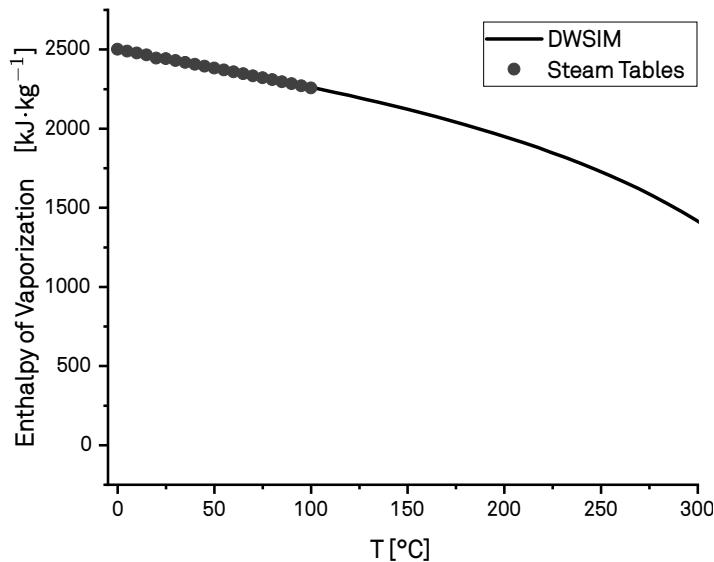


Figure 1.28: Comparision of enthalpy of vaporization data of water from DWSIM database with literature values according to steam tables. [59]

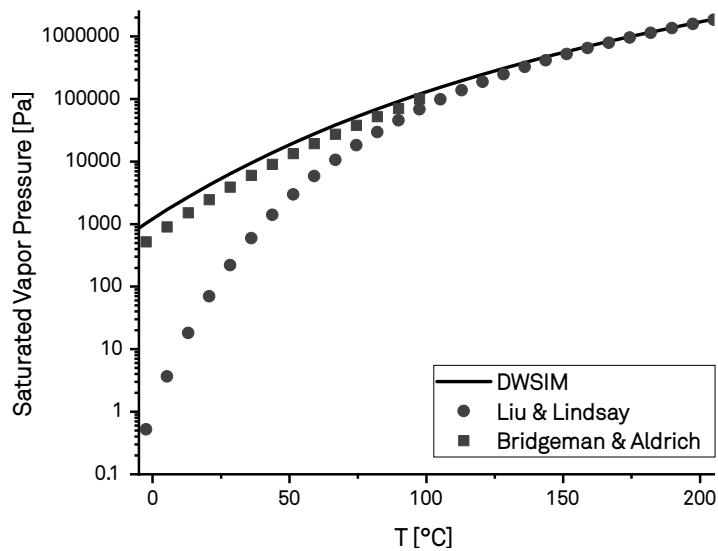
Water - Saturated Vapor Pressure

A comparison was made using the Antoine equation with parameters from [60] in different temperature ranges and regressed from different sources:

$$\log_{10} (p^{\text{LV}}) = A - \frac{B}{T + C} \quad (1.110)$$

Table 1.2: Antoine parameters used for validation of DWSIM database values [60].

Temperature Range [K]	A	B	C	Reference
379-573	3.55959	643.748	-198.043	[61]
273-303	5.40221	1838.675	-31.737	[62]
304-333	5.20389	1733.926	-39.485	[62]
334-363	5.0768	1659.793	-45.854	[62]
344-373	5.08354	1663.125	-45.622	[62]

**Figure 1.29:** Comparision of vapor pressure data of water from DWSIM database with literature values using the Antoine equation with parameters from [60] according to data from [61] and [62].

Air - Heat Capacity

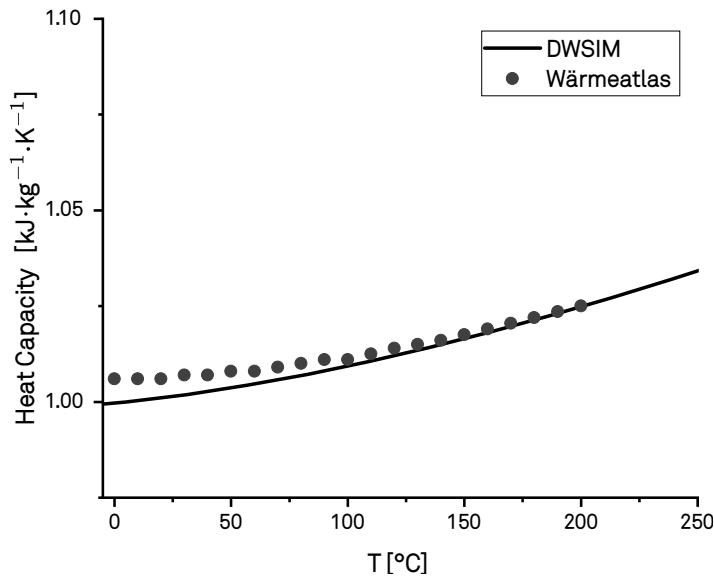


Figure 1.30: Comparision of heat capacity data of air from DWSIM database with literature values according to the VDI Heat Atlas. [58]

Air - Viscosity

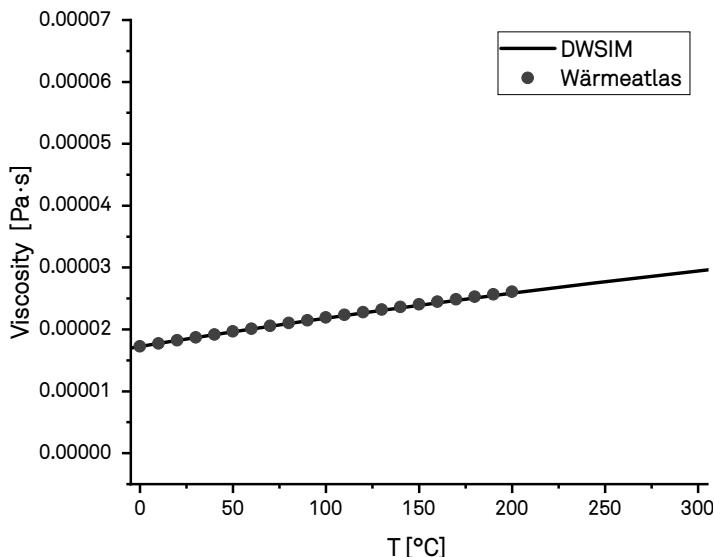


Figure 1.31: Comparision of viscosity data of air from DWSIM database with literature values according to the VDI Heat Atlas. [58]

Air - Thermal Conductivity

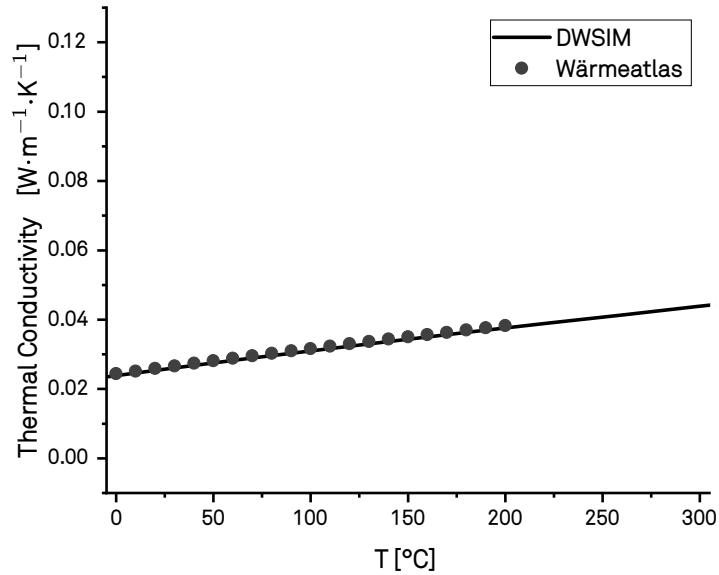


Figure 1.32: Comparision of thermal conductivity data of air from DWSIM database with literature values according to the VDI Heat Atlas. [58]

Ethanol - Density

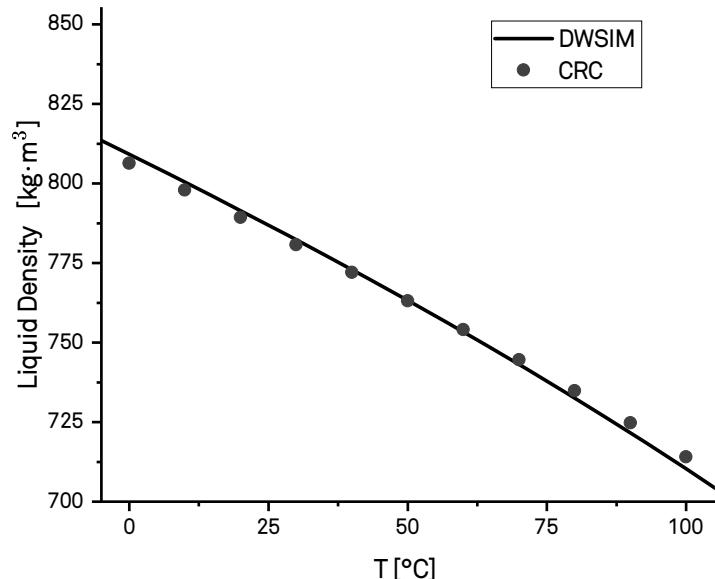


Figure 1.33: Comparision of liquid density data of ethanol from DWSIM database with literature values according to the CRC Handbook of Chemistry and Physics. [63]

Ethanol - Viscosity

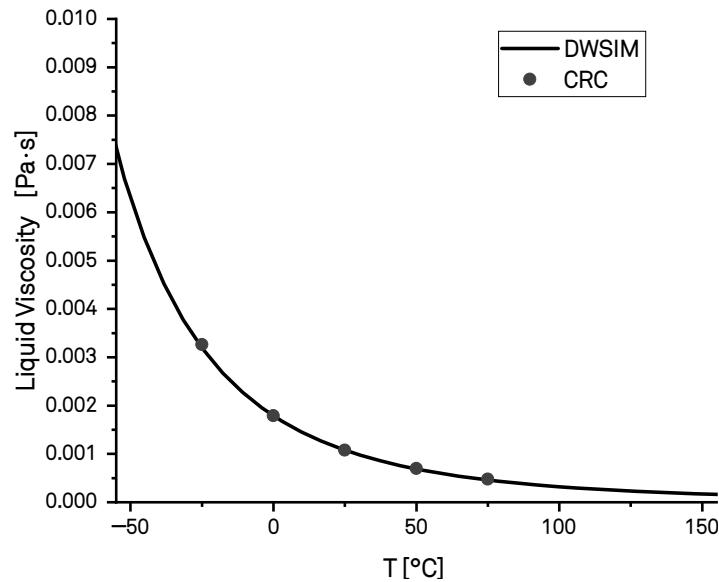


Figure 1.34: Comparision of liquid viscosity data of ethanol from DWSIM database with literature values according to the CRC Handbook of Chemistry and Physics. [63]