Advanced Powder Technology 30 (2019) 555–564

Contents lists available at ScienceDirect

Advanced Powder Technology

journal homepage: www.elsevier.com/locate/apt

Original Research Paper

Investigation of an FFT-based solver applied to dynamic flowsheet simulation of agglomeration processes ☆

Vasyl Skorych [a,*], Maksym Dosta [a], Ernst-Ulrich Hartge [a], Stefan Heinrich [a], Robin Ahrens [b], Sabine Le Borne [b]

[a] Hamburg University of Technology, Institute of Solids Process Engineering and Particle Technology, Denickestrasse 15, 21073 Hamburg, Germany
[b] Hamburg University of Technology, Institute of Mathematics, Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany

ARTICLE INFO

Article history:
Received 8 October 2018
Received in revised form 27 November 2018
Accepted 6 December 2018
Available online 11 December 2018

Keywords:
Population balance equation
Fast Fourier transformation
Adaptive cross approximation
Dynamic flowsheet simulation
Continuous agglomeration

ABSTRACT

The growth of particles due to agglomeration is often mathematically described by population balance equations. The numerical evaluation of these equations and applying new methods to their solution is an area of increasing interest. In this contribution, a new approach for solving the agglomeration population balance model based on a separable approximation of the agglomeration kernel and a fast Fourier transformation is investigated. Its applicability within a dynamic flowsheet simulation of continuous agglomeration processes with complex structures is analysed. A simulation framework Dyssol is used to study the new method and compare it to the well-known fixed pivot technique. Studies have shown that the new approach can provide a more efficient solution if certain constraints on the number of classes and on the separation rank of the agglomeration kernel are met.

1. Introduction

Particulate processes with a change in particle size can be found in a large number of industries including mineral processing, agriculture, pharmaceutics, food, chemistry, etc. Among such fundamental processes, one can distinguish particle formation (via wetting and nucleation), consolidation and growth, attrition and breakage [1]. Most of these processes usually occur simultaneously, but in some cases, one of them dominates the others. This contribution considers the pure agglomeration process and its dynamics with respect to the flowsheet simulation.

Agglomeration by definition is the growth process where primary particles are sticking together, forming larger porous secondary bodies, where the original particles can still be distinguished [2]. During this process, small initial particles are tailored to reach the properties required for final products.

Population balance models (PBM) are widely used to describe the population dynamics of particulate processes in various areas, such as fluidized bed spray granulation [3,4], fluidized bed melt agglomeration [5], crystallization [6,7], aerosol and droplet coalescence [8,9], polymerization [10], coagulation of colloids [11], etc.

Industrial scale production processes, involving agglomeration, often operate in a continuous mode and have complex structures consisting of several linked apparatuses, connected by material and information flows. Moreover, in order to optimize the process by reducing energy and raw material consumption, they can contain one or several recycle streams. Because of this complexity, it is usually difficult to investigate the influence of various parameters on the process behaviour. In that case, it is common to use flowsheet simulation [12] to study those processes. On the one hand, the methods applied in such simulation environments must ensure a fairly accurate representation of real processes. On the other hand, they should provide the necessary efficiency of calculations. Thus, inadequate computational load can be a limiting factor for applying a particular method or the entire approach. The solution of the population balance equations with agglomeration terms is one of such limiting factors, being a very computationally intensive and time-consuming operation.

In this contribution, a new approach for solving the PBM of the agglomeration process based on a separable approximation and a fast Fourier transformation (FFT) is investigated. This method, originally introduced and analysed in [13] and [14], was augmented with the adaptive cross approximation (ACA) approach to obtain a separable representation of the agglomeration kernel functions.

☆ Open Access for this article was sponsored by the Society of Powder Technology, Japan, through the KAKENHI Grant Number 18HP2009/ Grant-in-Aid for Publication of Scientific Research Results, Japan Society for the Promotion of Science, 2018.

* Corresponding author.
E-mail address: vasyl.skorych@tuhh.de (V. Skorych).

https://doi.org/10.1016/j.apt.2018.12.007
0921-8831/

The dynamic flowsheet simulation framework Dyssol [15] was used in this contribution to study the new method of solving the agglomeration integrals applied to existing continuous production processes.

2. Continuous fluidized bed agglomeration

A particle formation during the agglomeration process is usually introduced in terms of population balance models. This approach describes a temporal change of the particle number distribution in relation to the selected particle properties. A continuous fluidized bed agglomeration process with a homogeneous distribution of particles can be described by the following one-dimensional PBM [16,17]:

$$\frac{\partial n(v,t)}{\partial t} = B_{agg}(n,v,t) - D_{agg}(n,v,t) + \dot{n}_{in}(t) - \dot{n}_{out}(t), \tag{1}$$

$$B_{agg}(n,v,t) = \frac{1}{2}\beta_0 \int_0^v \beta(u, v-u) n(u,t) n(v-u,t) du, \tag{2}$$

$$D_{agg}(n,v,t) = \beta_0 n(v,t) \int_0^\infty \beta(u,v) n(u,t) du, \tag{3}$$

where v and u are volumes of agglomerating particles; $n(v,t)$ is the number density function; $\dot{n}_{in}(t)$ and $\dot{n}_{out}(t)$ are the number density distributions of inlet and outlet streams correspondingly; $B_{agg}(n,v,t)$ and $D_{agg}(n,v,t)$ are the birth and death rates of particles with volume v caused due to agglomeration; β_0 is the agglomeration rate constant, dependent on operating conditions but independent from particle sizes; $\beta(v,u)$ is the agglomeration kernel describing the agglomeration frequency between particles of volumes v and u, which produce a new particle with the size $(v+u)$. In this paper the following kernels are considered:

$$\beta_B(v,u) = \left(v^{\frac{1}{3}} + u^{\frac{1}{3}}\right)\left(v^{-\frac{1}{3}} + u^{-\frac{1}{3}}\right), \quad \text{Brownian motion (continuum)} \tag{4}$$

$$\beta_P(v,u) = \frac{(v+u)^{0.71}}{(vu)^{0.062}}. \quad \text{Peglow kernel} \tag{5}$$

The Brownian kernel [11,18] was chosen as an example of an analytically separable kernel, which is a requirement for the FFT-based method (see chapter 3). The Peglow kernel was empirically developed by Kapur [19], while its adjustable parameters were calculated by Peglow [17]. This kernel is not analytically separable but can be approximated very accurately by a separable expansion.

$n(v,0)$ denotes the initial holdup in the vessel. It is assumed that the bed mass in the apparatus remains constant, therefore, the mass flow of the outlet ($\dot{m}_{out}$) in the developed model depends on the mass flow in the input stream ($\dot{m}_{in}$) as:

$$\dot{m}_{out}(t) = \dot{m}_{in}(t). \tag{6}$$

3. Application of FFT to solve the PBE for agglomeration

In population balance equations the birth rate integral (2) is numerically very expensive to compute. It has been previously shown in [20] that the calculation of the agglomeration integral takes up to 85% of the overall computational time, having $O(N^2)$ complexity (where N is the number of discrete classes). For the fast evaluation of the birth rate integral (2), a new approach, previously introduced in [13] and [14], is used. It reduces the original $O\left(N^2\right)$ to almost linear $O(N \log N)$ complexity. The prerequisites for such a fast evaluation are a separable kernel representation and a fast Fourier transformation. The fast Fourier transformation requires uniform grids to discretize the property coordinate v. A function $\beta(v,u)$ is called separable with separation rank M if it can be expressed in the form

$$\beta(v,u) = \sum_{i=1}^{M} a_i(v) b_i(u), \tag{7}$$

for functions a_i and b_i depending only on a single variable.

The agglomeration birth (2) and death (3) terms can then be computed as sums of M terms of the form

$$\begin{aligned} B_{agg}(n,v,t) &= \frac{1}{2}\int_0^v a_i(v-u) n(u,t) n(v-u,t) b_i(u) du \\ &= \frac{1}{2}\int_0^v \psi_i(v-u,t)\varphi_i(u,t) du, \end{aligned} \tag{8}$$

$$\begin{aligned} D_{agg}(n,v,t) &= a_i(v) n(v,t) \int_0^{1-v} b_i(u) n(u,t) du \\ &= \psi_i(v,t) \int_0^\infty \varphi_i(u,t) du, \end{aligned} \tag{9}$$

where functions $\psi_i := a_i n$ and $\varphi_i := b_i n$ have been introduced to simplify notation. The factors $a_i(v)$ and $b_i(u)$ of the separable approximation of the kernel β as well as density distribution function n, and hence ψ and φ, are approximated through piecewise constant functions on a uniform grid.

Among the agglomeration kernels presented above in Eqs. (4), (5), the Brownian motion kernel β_B is separable. The Brownian kernel has separation rank 3 in view of

$$\beta_B(v,u) = 2 + v^{\frac{1}{3}}u^{-\frac{1}{3}} + v^{-\frac{1}{3}}u^{\frac{1}{3}} = \sum_{i=1}^{3} a_i(v) b_i(u) \tag{10}$$

with $a_1(v) = \sqrt{2}$, $a_2(v) = v^{\frac{1}{3}}$, $a_3(v) = v^{-\frac{1}{3}}$ and $b_1(u) = \sqrt{2}$, $b_2(u) = u^{-\frac{1}{3}}$, $b_3(u) = u^{\frac{1}{3}}$.

The Peglow kernel function $\beta_P(v,u)$ is not separable, but may be approximated by a rank-M separable function

$$\beta_P^{(M)}(v,u) \approx \sum_{i=1}^{M} a_i(v) b_i(u). \tag{11}$$

There exist several analytic approaches in the literature to derive such separable approximations, including (Chebyshev) interpolation, adaptive or hybrid cross approximation (ACA, HCA) algorithms, sinc approximation or approximation by exponential sums ([21–24]). In [13] the kernel separation has been accomplished using Chebyshev polynomials. In this paper, to derive a separable approximation, the adaptive cross approximation approach is introduced. This method is new with respect to its application in this context. The separable cross approximation for a continuous, bivariate function $\beta(v,u)$ is constructed recursively by setting

$$E_0(v,u) = \beta(v,u) \tag{12}$$

and defining error functions

$$E_i(v,u) = E_{i-1}(v,u) - \frac{E_{i-1}(x_i,u) E_{i-1}(v,x_i)}{E_{i-1}(x_i,x_i)} \tag{13}$$

with fixed interpolation points $x_i \in U_{cheb}$, $1 \le i \le M$ for a rank M-approximation. Scaled/shifted Chebyshev nodes

$$U_{cheb} := \left\{ \left(\frac{1}{2} + \frac{1}{2}\cos\frac{(2k-1)\pi}{2M}\right) | 1 \le k \le M \right\}. \tag{14}$$

were chosen as these interpolation points.

The separable cross approximation is then defined as

$$\beta^{(M)}(v,u) = \sum_{i=1}^{M} \frac{E_{i-1}(x_i,u)E_{i-1}(v,x_i)}{E_{i-1}(x_i,x_i)}. \tag{15}$$

In view of (15) the source agglomeration integral can be then approximated as

$$\begin{aligned}
&\frac{1}{2}\int_0^v \beta^{(M)}(v-u,u)n(v-u)n(u)\,du \\
&= \sum_{i=1}^{M}\frac{1}{2}\int_0^v \frac{E_{i-1}(x_i,u)E_{i-1}(v-u,x_i)}{E_{i-1}(x_i,x_i)}n(v-u)n(u)\,du \\
&= \sum_{i=1}^{M}\frac{1}{2}\int_0^v \frac{E_{i-1}(v-u,x_i)n(v-u)}{E_{i-1}(x_i,x_i)}E_{i-1}(x_i,u)n(u)\,du \\
&= \sum_{i=1}^{M}\frac{1}{2}\int_0^v \varphi_i(v-u)\psi_i(u)\,du.
\end{aligned} \tag{16}$$

The separable kernel approximation turns the birth rate agglomeration integral $\int_0^v \varphi_i(v-u)\psi_i(u)du$ into a convolution form $\varphi_i * \psi_i$, which after piecewise constant discretisation, can be computed efficiently [25] through

$$\varphi_i * \psi_i = IFFT(FFT(\varphi_i) \odot FFT(\psi_i)). \tag{17}$$

This evaluation uses the (inverse) Fast Fourier transformation (FFT/IFFT) and the elementwise product ($\odot$) and is based on the convolution-theorem. For details, see [13] and [14].

The calculation of the (inverse) fast Fourier transform is possible for any number of discrete classes N, but it is both the fastest and the simplest if N is equal to a power of 2. Therefore, the chosen grid is extended to the next larger power of 2, to profit from this higher speed and simplicity in implementation.

The proposed new method was compared with the well-known fixed pivot method [26] that calculates the summations of the discretized equations (2) and (3) in a straightforward manner in $O\left(N^2\right)$ complexity.

4. Dynamic flowsheet simulation in Dyssol

Dynamic flowsheet simulation is intended for the numerical study of transient processes occurring in all apparatuses of a production chain. When applying this simulation approach, the process is represented with a set of units, which describe apparatuses, their parts or sub-processes, and a set of material streams, intended to transfer information about material parameters between them.

In this work the simulation framework Dyssol [15] is used to perform test studies. It is being developed within the priority program of the German Research Foundation (DFG) SPP 1679 "Dynamic simulation of interconnected solids processes DYNSIM-FP", which focuses on investigations in the area of dynamics of solids processes [27].

The simulation system Dyssol implements a sequential-modular approach [28] that makes it possible to treat each individual model on a flowsheet separately, using its own numerical methods and computational algorithms. This significantly increases the flexibility of the system and simplifies the development and implementation of new models [3]. The most significant drawback of this method is the more complex computation of flowsheets containing recycle streams. Before the calculation begins, such streams must be initialized with some values [29] in order to break the cycle. After that, all units within the recycle loop are calculated iteratively until convergence.

Finding a convergent solution on a large time interval is a very complicated and time-consuming operation. To improve calculation performance in this case, Dyssol implements a modified waveform relaxation method (WRM) [30,31]. Applied to the flowsheet simulation, it implies splitting the entire simulation time into smaller intervals [31], called time windows. All units within the recycle loop are iteratively calculated for each such short interval, so that convergence on it can be reached much faster. The size of the time window changes dynamically: if the convergence is not reached prior to the specified upper iterations limit, the current time window decreases and calculations continue; if the convergence is achieved within the number of iterations below the lower limit, the size of the time window for the next time interval increases with a given rate constant. Instead of using the results of the previous iteration to initialize recycle streams, convergence methods are applied to calculate initial values for each iteration [32]. It additionally reduces the number of iterations required to achieve convergence. The criterion of convergence for the time interval $[T;T+T_{WIN}]$ can be expressed as

$$\forall t \in [T;T+T_{WIN}], \left|Y_i(t) - Y_{ref,i}(t)\right| < |Y_i(t)| \cdot R_{tol} + A_{tol}, \tag{18}$$

where $Y_i(t)$ is the calculated value at the time point t on the iteration i; $Y_{ref,i}(t)$ is the reference value calculated by a convergence method at the time point t on the iteration i; R_{tol}, A_{tol} are relative and absolute tolerances accordingly. For example, applying the Wegstein convergence method [33], the reference values for the next iteration $i+1$ can be calculated, using the obtained results from the current ($Y_i(t)$) and two previous ($Y_{i-1}(t)$ and $Y_{i-2}(t)$) iterations as:

$$\begin{aligned}
Y_{ref,i+1}(t) &= k \cdot Y_{i-1}(t) + (1-k)\cdot Y_i(t), \\
k &= \tfrac{s}{s-1}, \\
s &= \tfrac{Y_i(t)-Y_{i-1}(t)}{Y_{i-1}(t)-Y_{i-2}(t)}.
\end{aligned} \tag{19}$$

To solve the differential-algebraic equations (DAE), Dyssol applies the IDA (Implicit Differential-Algebraic) solver [34] from the SUNDIALS package [35], which uses the variable-order, variable-coefficient backward differentiation formula (BDF) in fixed-leading-coefficient form. Applying this method to solve the initial-value problem of form

$$F(t,y,\dot{y}) = 0, \quad y(t_0) = y_0, \quad \dot{y}(t_0) = \dot{y}_0, \tag{20}$$

one obtains a nonlinear algebraic system

$$G(y_n) \equiv F\left(t_n, y_n, h_n^{-1}\sum_{i=0}^{q}\alpha_{n,i}y_{n-i}\right) = 0, \tag{21}$$

which must be solved for each time step. Here t is an independent value (time); y_0 and $\dot{y}_0$ are given initial values; y_n is the computed approximation of $y(t_n)$; q is the method order in the range from 1 to 5; α is a coefficient determined by the method order and the history of time steps, chosen so that to maximize q; h is the time step size, such that

$$h_n = t_n - t_{n-1} \tag{22}$$

The time step h_n and the method order q are chosen dynamically to minimize local truncation error according to specified tolerances.

The solution of the system (21) is performed by using a modified Newton algorithm, where the Jacobian J is calculated not every time step, but only when it is necessary for convergence, e.g. when a non-fatal convergence failure occurs. Applying the Newton iteration results in a linear system for Newton correction of the form:

$$J\left[y_{n(m+1)} - y_{n(m)}\right] = -G\left(y_{n(m)}\right), \tag{23}$$

where $y_{n(m)}$ is the m-th approximation to y_n and J is the approximation of the system's Jacobian:

$$J = \frac{\partial G}{\partial y} = \frac{\partial F}{\partial y} + \frac{\alpha_{n,0}}{h_n}\frac{\partial F}{\partial \dot{y}}. \tag{24}$$

To solve the linear system for the Newton corrections (23), the direct linear solver for dense matrices is used. If convergence fails with the current J, the time step is reduced so that $h_n = h_n/4$, and the integration step is redone.

In order to represent a continuous process in the modelling environment, it is discretized with a predetermined accuracy using so called time points. Each such point describes the state of a model at a specific time, being actually a snapshot of the current process conditions. Since the simulation system employs the sequential-modular approach, each model of a flowsheet describes its parameters, as well as the parameters of the output material, for its own set of time points (Fig. 1).

If there are no recycle loops in the flowsheet structure, the algorithm for solving the agglomeration model is straightforward. For the entire simulation interval $[T_{START}; T_{END}]$, the DAE solver is called to calculate the population balance Eq. (1). The DAE solver, in turn, calls the agglomeration solver to calculate both birth (2) and death (3) terms of Eq. (1) and then starts integrating PBM. The equation is evaluated several times to obtain a solution for a single time point, meaning that the agglomeration solver is called multiple times within one invocation of the DAE solver. When each next time point is calculated with a specified accuracy (18), it is written to the output material stream. Thus, the internal step of the DAE solver h (22) directly affects the number of generated time points, and therefore also affects the simulation time of the entire circuit. When the calculation of the current model on the interval $[T_{START}; T_{END}]$ is completed, all the resulting time points are transferred to the next model in the flowsheet structure as an input parameter, and this model is calculated.

Existence of recycle loops slightly complicates the computational algorithm, due to the application of the waveform relaxation method. All units within the recycle loop are calculated at each small time interval $[T; T + T_{WIN}]$ iteratively. Consequently, to calculate each time window, the above procedure will be repeated several times, accordingly increasing the number of calls to the DAE solver and, in turn, to the agglomeration solver.

During this work the simulation system Dyssol has been extended by several models, which have been implemented using provided program interfaces.

5. Reference flowsheets of the continuous fluidized bed agglomeration process

Fig. 2 shows the flowsheet of a simple fluidized bed agglomeration process with continuous external material feed.

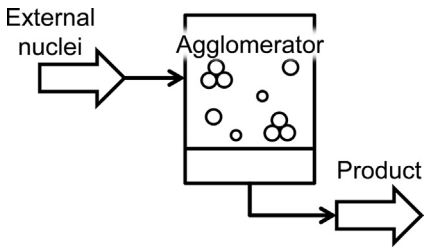

Fig. 2. Flowsheet structure 1 of a continuous agglomeration process.

The Agglomerator unit is described according to (1) and (6) and can utilize both the fixed pivot method as well as the FFT-based approach to calculate the birth (2) and death (3) rate integrals for the Brownian (4) or Peglow (5) kernel.

For test studies with Flowsheet 1, the parameters from Table 1 were used, where the particle sizes are given by the normal distribution of the mass-related density function

$$f(x_i) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right), \tag{25}$$

where x_i is the particle diameter; μ and σ are the mean value and the standard deviation of the normal distribution, respectively.

The agglomeration rate factors β_0 were chosen in such a way that both agglomeration kernels produce results of the same order.

The following simulation parameters were used for all case studies:

Fig. 3 shows the flowsheet of a continuous fluidized bed agglomeration process with an external classification, which was used for the investigation of the dynamic behaviour of continuous agglomeration. Besides the *Agglomerator* unit itself, it includes two screens and a mill unit. New particles enter the system through the input stream *External nuclei*. Within *Agglomerator*, the growth of particles occurs due to agglomeration (1). To simplify the model, the probability of particles leaving from the outlet does not depend on their size. *Screen 1* performs separation of the oversized agglomerates from the *Agglomerator*'s outlet. The oversized material is crushed to the desired size with the help of the *Mill* apparatus. The undersized particles that are separated by *Screen 2* are combined with the milled material and sent back to *Agglomerator* by mixing with *External nuclei*. The middle fraction of two screens is used as the product.

The screen's model is described by the distribution function $G(x_i)$ defined by Plitt [36], which determines the mass fraction of the material within the size class i from the feed that leaves the screen in the coarse stream:

$$G(x_i) = 1 - \exp\left(-0.693\left(\frac{x_i}{x_{cut}}\right)^{\alpha}\right). \tag{26}$$

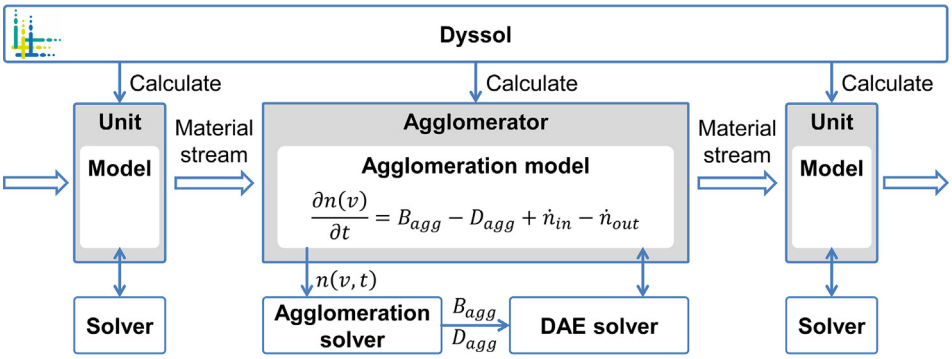

Fig. 1. Process simulation scheme in the Dyssol framework.

Table 1
Model parameters used for Flowsheet 1 and Flowsheet 2.

External nuclei	
Mass flow	10.8 kg/h
μ	3 mm
σ	0.2 mm
Agglomerator	
Holdup mass	20 kg
μ	3 mm
σ	0.1 mm
β_0	1e−10 (Brownian)/7.7e−6 (Peglow)

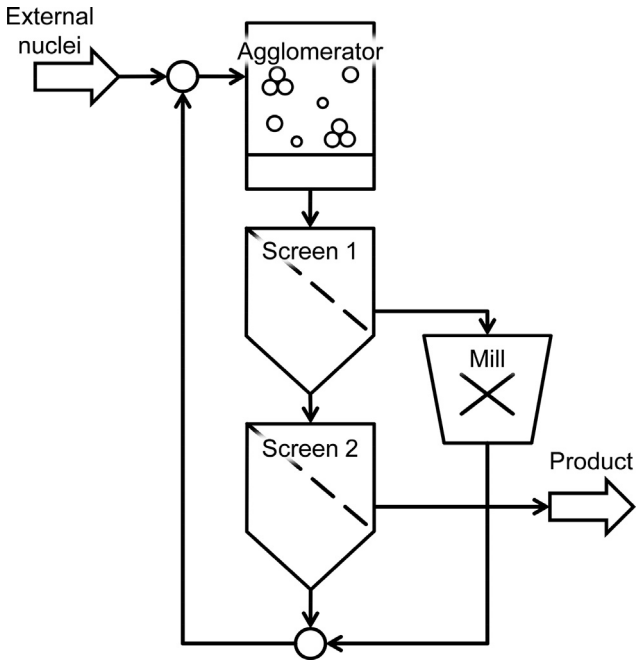

Fig. 3. Flowsheet structure 2 of a continuous agglomeration process.

Here x_i is the particle diameter; x_{cut} is the cut size of the classification model; α is the separation sharpness.

The mill's model is described by

$$w_{out,i} = \sum_{k=0}^{i} w_{in,k} S_k B_{ki} + (1 - S_i) w_{in,i}. \tag{27}$$

Here $w_{in,i}$ and $w_{out,i}$ are the mass fraction of particles within the size class i in the inlet and outlet distributions, respectively. S_k is described by the King selection function [37]

$$S_k = \begin{cases} 0, & x_k \le x_{min} \\ 1 - \left(\frac{x_{max} - x_i}{x_{max} - x_{min}}\right)^n, & x_{min} < x_k < x_{max}, \\ 1, & x_k \ge x_{max} \end{cases} \tag{28}$$
$$x_{min} = CSS \cdot \alpha_1,$$
$$x_{max} = CSS \cdot \alpha_2.$$

Here S_k is the mass fraction of particles within the size class k that will be crushed; x_k is the mean diameter of particles within the size class k; CSS is the close size setting of the crusher; α_1, α_2, n are parameters of the King selection function.

The value B_{ki} from (27) is calculated according to the Vogel breakage function [38]:

$$B_{ki} = \begin{cases} 0.5\left(\frac{x_k}{x_i}\right)^q \left(1 + \tanh\left(\frac{x_k - x'}{x'}\right)\right), & i \le k \\ 0, & i > k \end{cases}. \tag{29}$$

Here B_{ki} is the mass fraction of particles within the size class i, which get a size less than or equal to k after breakage; x' is the minimum fragment size that can be achieved by crushing; q is the parameter of the Vogel breakage function.

For test studies with Flowsheet 2, the model parameters from Table 1 and Table 3 were used.

The simulation parameters from Table 2 and Table 4 were used for case studies of Flowsheet 2.

For all case studies, the particle size is given by a distribution on a continuous volume-equidistant grid defined on the interval [0; 268] mm^3 or, in terms of particle diameter, on the interval [0; 8] mm.

6. Case studies and performance analysis

To investigate the performance improvement by using the new FFT-based method in real simulations, several case studies of the continuous agglomeration process (Fig. 2 and Fig. 3) have been conducted. Different combinations of kernels, calculation approaches and settings were examined:

1. Brownian kernel with the fixed pivot [26] method.
2. Brownian kernel with the FFT method.
3. Peglow kernel with the fixed pivot method.
4. Peglow kernel with the FFT method using separable approximation of different ranks.

The Brownian kernel was chosen as an example of an analytically separable kernel, while the Peglow kernel requires a separable approximation, and therefore different approximation ranks have been analysed.

Each solver was investigated 10 times for a different number of classes N: from 80 to 800 in increments of 80. For all performance tests, the simulation of each case study has been performed 3 times, and then the average performance characteristics were calculated.

Due to the waveform relaxation method used in the Dyssol framework for iterative calculations, the computational performance of the system is very sensitive to the initial conditions, chosen for recycle streams. To calculate the consistent initial conditions, the DAE-solver solves the system (20) using the Newton iteration with a line search global strategy at the time point 0 starting from some initial guess [34]. Achieving convergence for the very first time point in this case can take a considerable time, in extreme cases comparable with the time of the entire simulation. To exclude the influence of the time required for initialization, flowsheets in all test studies were provided with consistent initial conditions. Fig. 4 and Fig. 5 show simulation results for Flowsheet 1 (Fig. 2) and Flowsheet 2 (Fig. 3), respectively. The steady-state is reached after 4 and 7 h of the process time. Due to the chosen agglomeration rate constants, both Brownian and Peglow kernels show nearly the same dynamics.

6.1. Influence of the approximation rank

In [13], it was shown that computational time increases linearly with respect to an increase of the approximation rank of analytically not separable agglomeration kernels. However, for many

Table 2
Simulation parameters used for Flowsheet 1 and Flowsheet 2.

Relative tolerance	1e−5
Absolute tolerance	1e−7
Process time	10 h

Table 3
Model parameters used for Flowsheet 2.

Screen 1	
x_{cut}	4.5 mm
α	9
Screen 2	
x_{cut}	4 mm
α	9
Mill	
CSS	3 mm
α_1	0.5
α_2	1.7
n	2
x'	3 mm
q	10

Table 4
Simulation parameters used for Flowsheet 2.

Initial time window	0.1 s
Time window change rate	1.2
WRM iterations upper limit	7
WRM iterations lower limit	3
Convergence method	Wegstein
Extrapolation method	Linear

problems, even low rank approximations result in a suitable accuracy. Therefore, for test cases with the Peglow kernel solved by the FFT-based method, the Sauter mean diameter (d_{32}) of the product was analysed at the end of the simulation (10 h of the modelling time) (Fig. 6). For both flowsheets, the solution begins to converge to a certain value as the number of classes increases. The final value of the Sauter mean diameter does not change significantly with an increase in the separation rank beyond 3. Hence, it makes sense to investigate the FFT-based method for the Peglow kernel for separation ranks $2 \leq M \leq 4$.

6.2. Performance analysis – Flowsheet 1

Primarily, a simple fluidized bed agglomeration process with a continuous external material feed was simulated using different combinations of solvers, kernels and class sizes.

While the simulation time of the fixed pivot method is strictly monotonically increasing with respect to the number of classes, the FFT-based solver shows a stepwise behaviour. It is due to the fact, that the FFT-routine uses an internal discretization with the number of classes equal to the power of two. Therefore, its effectiveness is the higher, the closer the used number of classes N to the next $2^{\lfloor \log_2 N \rfloor + 1}$ boundary. Similar behaviour is observed for almost all the following results.

As Fig. 7 shows, the use of the Peglow kernel with the FFT-based method leads to an increase in the number of generated time points in a given time interval with increasing M, meaning that the used DAE solver needs more iterations to reach convergence. This, in turn, leads to a linear increase in the number of agglomeration solver calls from about 3.5e+3 to about 3e+4 (Brownian) and 3.2e+4 (Peglow). Only for the Peglow kernel with M = 4, this value reaches 4.4e+4, leading to a noticeable increase in the number of time points (Fig. 7). Because of the improved accuracy in representing the particle size distribution (PSD) with an increase in the number of classes, the average integration step h of the DAE solver (21) also tends to increase. Therefore, there is a general tendency to reduce the number of generated time points with growing N (Fig. 7).

Linear growth of the number of agglomeration solver calls with increasing number of classes is related to the algorithm used by the applied IDA solver. The population balance Eq. (1) is calculated not

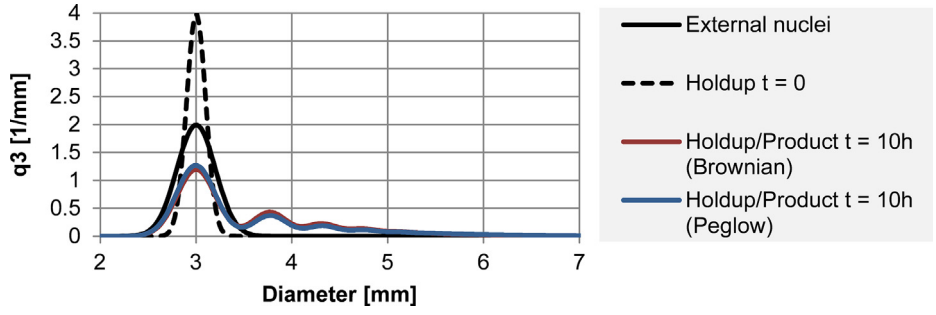

Fig. 4. Simulation results for Flowsheet 1.

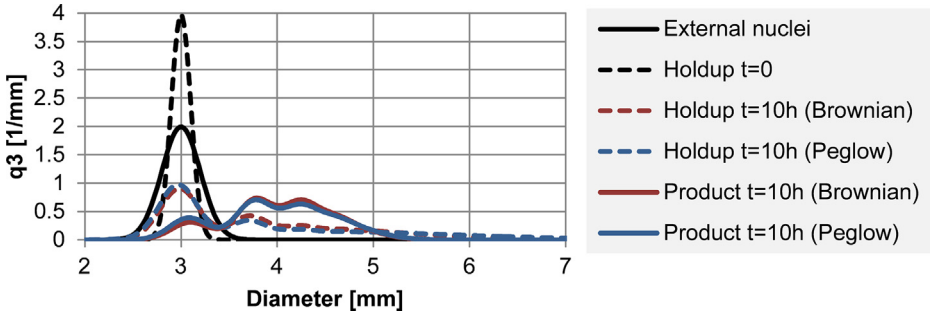

Fig. 5. Simulation results for Flowsheet 2.

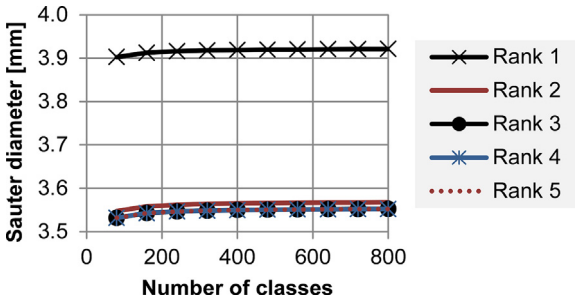

Fig. 6. Reached Sauter mean diameter for different approximation ranks of the Peglow kernel for Flowsheet 2.

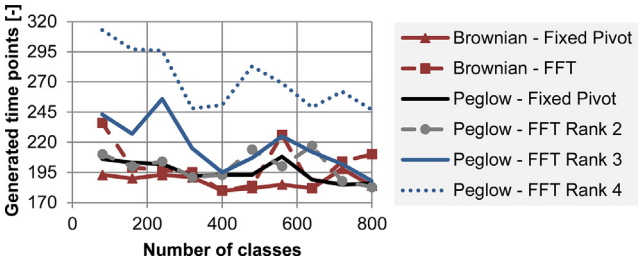

Fig. 7. Number of time points generated during the simulation of Flowsheet 1.

only to advance the system (21) in time, but also to update the system's Jacobian J (24). Each update of J requires N evaluations of the model, providing the observed tendency to a linear increase in the number of agglomeration solver calls. Also, if the approximation rank exceeds 3, nonlinear convergence failures occur in the IDA solver, which leads to additional recalculations of the Jacobian, a decrease in the time step h (22) of the solver and an increase in the number of generated time points. All this negatively affects the computational time.

The number of arising convergence failures affects not only the frequency of Jacobian recalculations, but also changes the current integration time step h and affects the method order q of the backward differentiation formula (21). Both touched parameters decrease as the approximation rank increases (especially for $M > 3$), leading to an increase in the computational load.

It should be noted that an increase in the number of agglomeration solver calls occurs in almost all cases of using the FFT-based solver, but it does not always lead to an increase in the total simulation time because of the better performance of each individual iteration of the FFT-based method as compared to the fixed pivot technique.

The simulation time of Flowsheet 1 (Fig. 2) consists of the calculation of

- the birth (2) and death (3) terms using the selected agglomeration solver;
- the population balance model (1) using the IDA solver;
- some internal operations of the Dyssol system.

To take the influence of all these factors into account, it is necessary to investigate the contributions of the given agglomeration solver and the entire Agglomerator unit to the total calculation time. Fig. 8 shows that while the whole simulation time is almost entirely spent on the calculation of the Agglomerator unit, up to 90% of it is occupied by the calculation of the birth and death terms.

For the fixed pivot technique, a constant increase in the computational time of each class is observed with increasing N (Fig. 9). The stepwise change of the effectiveness can be easily observed here for the FFT-based method. If the number of classes remains within the interval $2^{\lfloor \log_2 N \rfloor} < N < 2^{\lfloor \log_2 N \rfloor + 1}$, the efficiency of the FFT-based solver is nearly constant within the investigated region (Fig. 9).

Thus, for this case, the overall performance is largely dependent on the efficiency of the agglomeration solver. It fully correlates with Fig. 10 showing the total acceleration Q_{TOT} of the FTT-based computations with respect to the fixed pivot method, calculated as

$$Q_{TOT} = \frac{T_{TOT}^{FP} - T_{TOT}^{FFT}}{min\left(T_{TOT}^{FP}, T_{TOT}^{FFT}\right)} \cdot 100\%, \tag{30}$$

where T_{TOT}^{FP} and T_{TOT}^{FFT} are the total simulation time using the fixed pivot method and the FFT-based method, respectively.

The simulation with the Brownian kernel gains benefit from the FFT only for $N > 350$. For the Peglow kernel, the number of classes after which a performance gain can be achieved depends on the separation rank, and is equal to 200 and 620 classes for ranks 2 and 4, respectively, while rank 3 completely corresponds to the analytically separable Brownian kernel with the same rank. At the same time, the solution with rank 4 yielded significantly worse results as the FFT for all tested N. Thus, using the separation rank of up to 3 is optimal in terms of performance and accuracy in this case.

Based on this, it is possible to draw the following conclusions:

- The use of the FFT method in the case of the separable Brownian agglomeration kernel leads to more efficient calculations;
- In the case of a kernel with a separable approximation, the efficiency depends on the approximation rank;
- Efficiency decreases with increasing approximation rank;
- The FFT-based method is more efficient than the fixed pivot technique when the number of classes exceeds a certain threshold.

6.3. Performance analysis – Flowsheet 2

Evaluation of the solvers efficiency in the case of a more complex flowsheet with recycle streams (Fig. 3) is more complicated, since the simulation time consists of the calculation of

- the birth (2) and death (3) terms using the selected solver;
- the population balance model (1) using the IDA solver;
- models of the remaining units on the flowsheet (26)–(29);
- internal operations of the Dyssol system: waveform relaxation, convergence and extrapolation methods, convergence analysis, data exchange between units, etc.

In order to study the effect of all these contributions to the overall simulation time, it is necessary to separate the calculation times

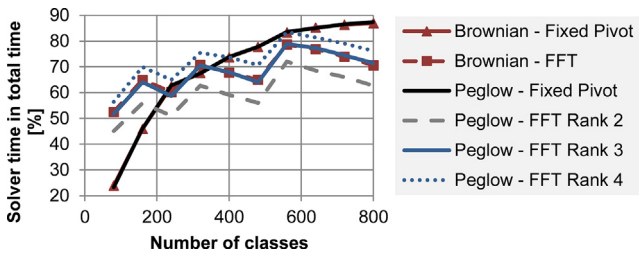

Fig. 8. Contribution of the agglomeration solver to the total simulation time for Flowsheet 1.

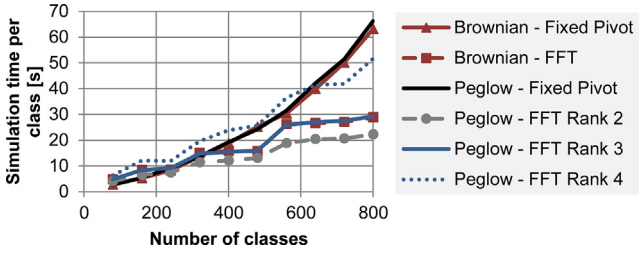

Fig. 9. Average simulation time per class for 10 h simulation time of Flowsheet 1.

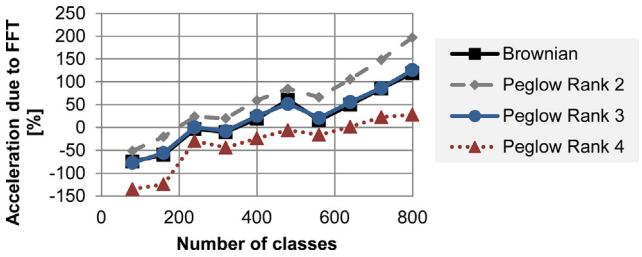

Fig. 10. Total acceleration of calculations due to the use of the FFT-based method for Flowsheet 1.

of the solver, of the *Agglomerator* unit and of the whole flowsheet. Fig. 11 shows that even though the calculation of the birth and death terms occupies a significant part of the *Agglomerator*'s operation time (Fig. 11, left), it is only a portion within the total simulation time (Fig. 11, right). Thus, its impact on the overall calculation performance is significantly reduced compared to Flowsheet 1. Despite this the FFT-based solvers with$M \leq 3$ still show better characteristics for the number of classes $N > 400$.

Because of the modular approach in combination with the WRM used in the Dyssol simulation system, all calculations within recycle loops are performed iteratively. This leads to the fact that the *Agglomerator* and hence the Eq. (1) are calculated several times for each time window. The DAE solver also has an iterative nature, so terms (2) and (3) will be evaluated several times for each call of *Agglomerator*.

The number of *Agglomerator* unit calls only slightly depends on the number of classes or the selected method and is determined by the number of time windows of the waveform relaxation method (Fig. 12). On the other hand, the number of the solver calls weakly correlates with the chosen method, but it depends largely on the number of classes, because of the recalculation of the Jacobian.

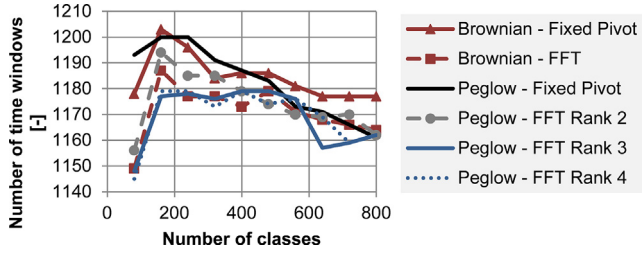

Fig. 12. Number of generated time windows of the waveform relaxation method for Flowsheet 2.

The maximum time step h of the IDA solver (21) is additionally limited by the time window of the waveform relaxation method, so that the nonlinear convergence failures are less common and the system can work with higher separation ranks M compared to Flowsheet 1.

The overall benefit from using the FFT-based method can be expressed in terms of acceleration of calculations with respect to the fixed pivot method (Fig. 13), where the total acceleration Q_{TOT} is calculated according to (30) and the solver's acceleration Q_{SOL} is defined as

$$Q_{SOL} = \frac{T_{SOL}^{FP} - T_{SOL}^{FFT}}{min\left(T_{SOL}^{FP}, T_{SOL}^{FFT}\right)} \cdot 100\%, \tag{31}$$

where T_{SOL}^{FP} and T_{SOL}^{FFT} are the simulation time spend by the agglomeration solver by using the fixed pivot method and the FFT-based method, respectively.

Although the efficiency of the agglomeration solvers per class remains the same as for Flowsheet 1, there are significant differences in computational performance. Taking into account only the agglomeration solver (Fig. 13, left), a noticeable performance gain is observed starting from about 400 classes. At the same time, the overall acceleration of calculations of the entire flowsheet reaches about 30% (Fig. 13, right) and depends on the chosen kernel and separation rank. While the Brownian kernel becomes more efficient already at 160 classes, the Peglow kernel is more effective only starting from 480 classes. At the same time, due to the use of the waveform relaxation method, the influence of the separation rank on the overall computational performance is much lower.

Taking into account that the calculation times for rank 2 and 3 are approximately equal, it seems optimal to use the separation rank 3 for problems with $N \geq 480$ in the case of applying the Peglow kernel.

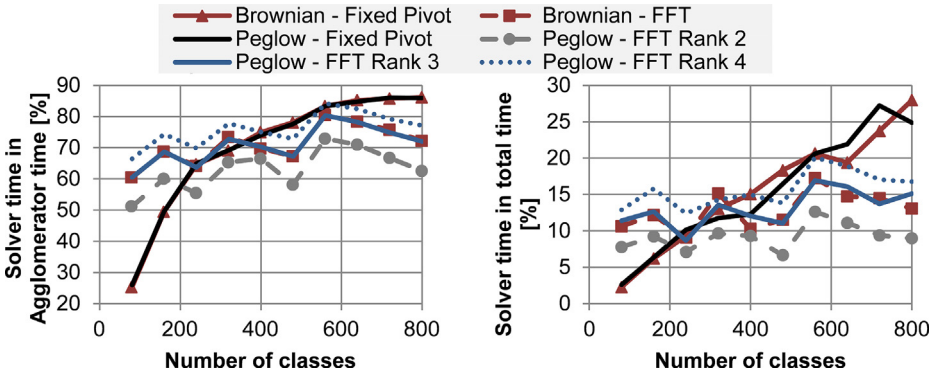

Fig. 11. Contribution of the agglomeration solver to the simulation time of the *Agglomerator* unit (left) and to the total simulation time (right) for Flowsheet 2.

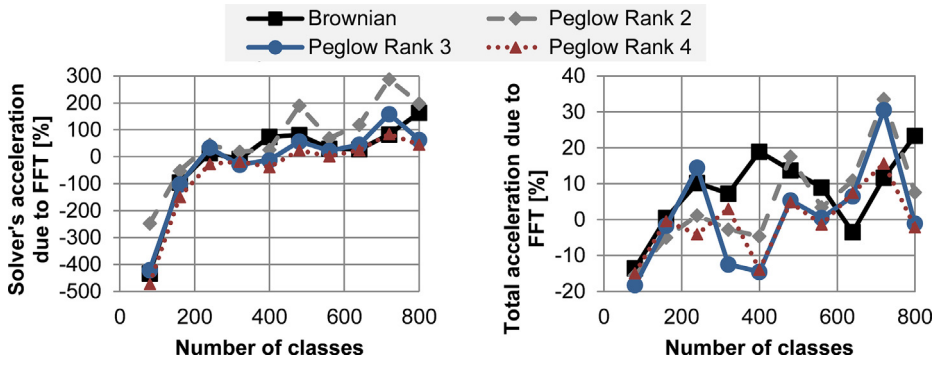

Fig. 13. Acceleration of calculations of the agglomeration terms (left) and of the whole flowsheet (right) due to the use of the FFT-based method for Flowsheet 2.

7. Conclusions

In this contribution, the applicability of the new FFT-based method for solving the population balance model of the agglomeration process for simulation of production schemes in solids processing technology was investigated. To perform the test cases, the dynamic flowsheet simulation system Dyssol was used, where the investigated solvers have been implemented. Since the method used requires a separable agglomeration kernel or its separable approximation, two agglomeration kernels were investigated and compared: the analytically separable Brownian kernel and the Peglow kernel, which was separated using the adaptive cross approximation with several ranks. This technique is new with respect to its application in this scope. The simulation performance was compared to the popular fixed pivot method.

Simulation studies show that the performance of the method depends heavily on the number of discrete classes of the particle size distribution. Taking the mutual influence of the agglomeration solver and the DAE solver on the convergence rate of the PBM solution into account, one can observe only a slight benefit when using a small number of classes.

Although the performance of the FFT-base solver can be superior to the fixed pivot method, its advantages in the dynamic flowsheet simulations can only be used under a certain set of conditions, namely:

- Use of low-level separation ranks. The use of separable approximation with high ranks can be even disadvantageous in terms of the overall simulation performance and at the same time does not provide any significant benefit in the accuracy of calculations for the tested cases.
- Relatively large number of classes N. The FFT-based solver is not suitable for small grid sizes and its efficiency and benefits increase with an increase in the number of classes.
- Number of classes is equal to $2^{\lfloor \log_2 N \rfloor + 1}$. With the given implementation, to gain the greatest benefit from the proposed solver, the number of PSD classes must be as close as possible to the next power-of-2-boundary.

When developing a generally applicable solver that can be used for a wide range of problems, it makes sense to use a combined approach: the fixed pivot method for cases with a small number of classes and the FFT-based method for those problems where it is necessary to work with a very fine discretization.

Acknowledgements

This work was supported by the German Research Foundation via Priority Program SPP 1679 "Dynamic simulation of interconnected solids processes" under the grants number HE 4526/14-3 and LE 3195/1-3.

References

[1] S.M. Iveson, J.D. Litster, K. Hapgood, B.J. Ennis, Nucleation, growth and breakage phenomena in agitated wet granulation processes: a review, Powder Technol. 20 (2001) 3–39.
[2] B.J. Ennis, J.D. Litster, Particle size enlargement, in: R. Perry, D. Green (Eds.), Perry's Chemical Engineers' Handbook, seventh ed.,., McGraw-Hill, New York, 1997, 20–56–20-89.
[3] M. Dosta, S. Heinrich, J. Werther, Fluidized bed spray granulation: Analysis of the system behaviour by means of dynamic flowsheet simulation, Powder Techn. 204 (2010) 71–82.
[4] C. Neugebauer, S. Palis, A. Bück, E. Diez, S. Heinrich, E. Tsotsas, A. Kienle, Influence of mill characteristics on stability of continuous layering granulation with external product classification, Comp. Aid. Chem. Eng. 38 (2016) 1275–1280.
[5] H.S. Tan, A.D. Salman, M.J. Hounslow, Kinetics of fluidised bed melt granulation V: simultaneous modelling of aggregation and breakage, Chem. Eng. Sci. 60 (2005) 3847–3866.
[6] V. Kulikov, H. Briesen, W. Marquardt, Scale integration for the coupled simulation of crystallization and fluid dynamics, Chem. Eng. Res. Des. 83 (2005) 706–717.
[7] A. Reinhold, H. Briesen, High dimensional population balances for the growth of faceted crystals: combining Monte Carlo integral estimates and the method of characteristics, Chem. Eng. Sci. 127 (2015) 220–229.
[8] F.M. Gelbard, J.H. Seinfeld, Coagulation and growth of a multicomponent aerosol, J. Colloid Interface Sci. 63 (1978) 472–479.
[9] S. Shekar, A.J. Smith, W.J. Menz, M. Sander, M. Kraft, A multidimensional population balance model to describe the aerosol synthesis of silica nanoparticles, J. Aerosol Sci. 44 (2012) 83–98.
[10] R. Dürr, A. Franz, A. Kienle, Combination of limited measurement information and multidimensional population balance models, IFAC-PapersOnLine 48 (2015) 261–266.
[11] M. Smoluchowski, Versuch einer mathematischen theorie der koagulationskinetik kolloider losungen, Z. Phys. Chem. 17 (1916) 129–168.
[12] E.U. Hartge, M. Pogodda, C. Reimers, D. Schwier, G. Gruhn, J. Werther, Flowsheet simulation of solids processes, Kona 24 (2006).
[13] S. Le Borne, L. Shahmuradyan, K. Sundmacher, Fast evaluation of univariate aggregation integrals on equidistant grids, Comput. Chem. Eng. 74 (2015) 115–127.
[14] W. Hackbusch, On the efficient evaluation of coalescence integrals in population balance models, Computing 78 (2006) 145–159.
[15] V. Skorych, M. Dosta, E.-U. Hartge, S. Heinrich, Novel system for dynamic flowsheet simulation of solids processes, Powder Techn. 314 (2017) 665–679.
[16] K.V.S. Sastry, Similarity size distribution of agglomerates during their growth by coalescence in granulation or green pelletization, Int. J. Miner. Process. 2 (1975) 187–203.
[17] M. Peglow, J. Kumar, G. Warnecke, S. Heinrich, L. Mörl, A new technique to determine rate constants for growth and agglomeration with size- and time-dependent nuclei formation, Chem. Eng. Sci. 61 (2006) 282–292.
[18] D.J. Aldous, Deterministic and stochastic models for coalescence (aggregation and coagulation): a review of the mean-field theory for probabilists, Bernoulli 5 (1999) 3–48.
[19] P.C. Kapur, Kinetics of granulation by non-random coalescence mechanism, Chem. Eng. Sci. 27 (1972) 1863–1869.
[20] F. Anker, S. Ganesan, V. John, E. Schmeyer, A comparative study of a direct discretization and an operator-splitting solver for population balance systems, Comput. Chem. Eng. 70 (2015) 95–104.
[21] M. Bebendorf, S. Rjasanow, Adaptive low-rank approximation of collocation matrices, Computing 70 (2003) 1–24.
[22] S. Börm, L. Grasedyck, Hybrid cross approximation of integral operators, Numer. Math. 101 (2005) 221–249.

[23] L. Grasedyck, Adaptive recompression of H-matrices, Computing 74 (2005) 205–223.
[24] W. Hackbusch, Hierarchical Matrices, Springer, 2015.
[25] R.N. Bracewell, The Fourier transform and its applications, d ed., McGraw-Hill, 2000.
[26] S. Kumar, D. Ramkrishna, On the solution of population balance equations by discretization – I. A fixed pivot technique, Chem. Eng. Sci. 51 (8) (1996) 1311–1332.
[27] SPP-1679: DynSim-FP, available at https://dynsim-fp.de (accessed on August 30th, 2018).
[28] M. Hillestad, T. Hertzberg, Dynamic simulation of chemical engineering systems by the sequential modular approach, Comput. Chem. Eng. 10 (1986) 377–388.
[29] G. Towler, R. Sinnott, Chemical Engineering Design. Principles, Practice and Economics of Plant and Process Design, second ed., Elsevier, 2013, pp. 223–236.
[30] E. Lelarasmee, A.E. Ruehli, A.L. Sangiovanni-Vincintelli, The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits: Theory and Applications, Diss. University of California, Berkley, 1982.
[31] M. Dosta, Dynamic Flowsheet Simulation of Solids Processes and Its Application to Fluidized Bed Spray Granulation, TU Hamburg-Harburg, 2013, Diss.
[32] A.C. Dimian, C.S. Bildea, A.A. Kiss, Integrated Design and Simulation of Chemical Processes, second ed., Elsevier, 2014, pp. 116–118.
[33] J.H. Wegstein, Accelerating convergence of iterative processes, Commun. ACM 1 (6) (1958) 9–13.
[34] A.C. Hindmarsh, R. Serban, A. Collier, User Documentation for IDA v2.6.0, 2009.
[35] SUNDIALS (SUite of nonlinear and DIfferential/ALgebraic equation solvers), available at https://computation.llnl.gov/projects/sundials (accessed on August 30th, 2018).
[36] L.R. Plitt, The analysis of solid-solid separations in classifiers, CIM Bull. 64 (708) (1971) 42–47.
[37] R.P. King, Modeling and Simulation of Mineral Processing Systems, Butterworth & Heinemann, Oxford, 2001.
[38] L. Vogel, W. Peukert, Modelling of grinding in an air classifier mill based on a fundamental material function, Kona 21 (2003) 109–120.