Original software publication

Dyssol—An open-source flowsheet simulation framework for particulate materials

Vasyl Skorych *, Maksym Dosta, Stefan Heinrich

Institute of Solids Process Engineering and Particle Technology, Hamburg University of Technology, Denickestrasse 15, 21073 Hamburg, Germany

ARTICLE INFO

ABSTRACT

Dyssol is a modelling framework for the dynamic flowsheet simulation of processes designed for handling of particulate materials. Main distinctive features of this software are the comprehensive description of multidimensionally distributed particulate materials, the application of transformation matrices and the use of sequential-modular simulation approach. This cross-platform system can be easily extended with new models, applied for calculation of large datasets and coupled to the external programme packages.

Code metadata

Current code version	v0.10.0.2
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_208
Code Ocean compute capsule	-
Legal Code Licence	BSD-3-Clause
Code versioning system used	git
Software code languages, tools, and services used	C++, Qt, HDF5, zlib, SUNDIALS, KISS FFT, Inno Setup.
Compilation requirements, operating environments & dependencies	Windows, Visual Studio 14 2015, Qt 5.11.3 msvc2015 / Qt 5. 11.3 msvc2015_64, Qt Visual Studio Tools for Visual Studio 2015, Git
If available Link to developer documentation/manual	https://github.com/FlowsheetSimulation/Dyssol-open/tree/master/Documentation
Support email for questions	vasyl.skorych@tuhh.de

Software metadata

Current software version	v0.10.0.2
Permanent link to executables of this version	https://github.com/FlowsheetSimulation/Dyssol-open/releases/latest
Legal Software Licence	BSD-3-Clause
Computing platforms/Operating Systems	Microsoft Windows
Installation requirements & dependencies	Visual C++ Redistributable for Visual Studio 2015
If available, link to user manual–if formally published include a reference to the publication in the reference list	https://github.com/FlowsheetSimulation/Dyssol-open/tree/master/Documentation
Support email for questions	vasyl.skorych@tuhh.de

1. Motivation and significance

Complex production processes are used for the intermediate handling or for the final manufacturing of the numerous

---

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

products, from the processing of the iron ore and ending with the production of pharmaceutical compounds. They often consist of several transformation steps occurring in different apparatuses, which are connected into a single chain by material or energy streams. These complex structures are referred to as flowsheets. Flowsheet simulation frameworks can be effectively used to describe the behaviour of such processes, optimize them, develop control strategies or to design new products.

In flowsheet simulations, each apparatus or transformation step is described using separate units that are connected by material streams, whereby the output of one unit can be connected to the input of another. From a computational point of view, material streams are a general construct that describes the flows of mass, energy or information and is applied to transfer them between units. Depending on the input streams and the current state of the unit, the transformations occurring in the unit are calculated, which leads to changes in the output streams. Such flexible and extendable approach makes it easy to define various process configurations, modify process parameters, and numerically investigate their behaviour.

There are numerous commercial flowsheet simulation tools like AspenPlus [1], gPROMS Formulated Products [2], ChemCad [3], etc. However, most of these programmes have been developed with a focus on liquid–vapour systems with simplified description of the solid phase. As a result, a comprehensive numerical description of the particulate material cannot be effectively performed in them. Due to various aspects, such as the description of the solid phase by interdependent multidimensional distributed parameters, modelling of solids processes requires specific treatment [4–6].

In this contribution, we present a new open-source simulation framework Dyssol, which has been especially developed with a focus on modelling solids processes. This framework has been developed based on our previous research [7–9], in the scope of the DynSimFP project [10]—a priority research programme financed by German Research Foundation (DFG) from 2013 until 2020.

2. Software description

2.1. Methods

A sequential-modular approach [11,12] is used to perform calculations in Dyssol. Compared to the equation-oriented (simultaneous) approach, modular simulation has a set of advantages that play a decisive role, especially for treatment of solids processes. The most important is that each single unit can be solved separately using its own calculation strategy. This feature plays a crucial role in the case of particulate materials, where single process steps are often described with complex mathematical models of heterogeneous nature. For example, coupling of several units described with multidimensional population balance models (PBM) into one equation set can already make the application of the equation-oriented approach unfeasible.

To perform data transfer between separate models and to reach convergence, the waveform relaxation approach is applied. This method can be effectively used for modular simulations on the scale of the entire flowsheet and on the scale of individual process units [13]. Furthermore, to improve overall efficiency, several convergence methods (direct substitution, Wegstein's [14] and Steffensen's [15] approaches) and several extrapolation methods (nearest neighbour, linear and cubic spline) are implemented [16].

In Dyssol, granular materials are described by discretized multidimensional interdependent distributed parameters. Each dimension corresponds to a specific particle property (size, shape,

etc.) and is divided into a set of intervals according to the discretization grid. A material is defined as a set of values, where each one describes the mass fraction of particles with specific combination of properties. An important aspect here is the correct and efficient handling of these large datasets. In Dyssol, all material streams are described with a sparse data format implemented based on the hierarchical tree topology [16]. This significantly reduces the size of stored data and the amount of calculations. To describe changes in particle distributions, transformation matrices are applied. They allow to avoid the loss of information about the secondary distributions and significantly extend the applicability of units [9].

The reader is referred to our previous works to obtain more detailed information about the simulation algorithm and analysis of its efficiency [8,16], the importance and treatment of distributed parameters [9], and the validation of the modelling system [8].

2.2. Software architecture

The Dyssol framework was developed using an object-oriented approach. Main software components can be divided into modules, which are schematically shown in Fig. 1.

The core of the system is the *Flowsheet* and *Simulator* modules. The first one describes the structure of the simulated process and all the parameters of the included models. The latter one is responsible for the execution of the main computational algorithm and for the data transfer between units. *Flowsheet* contains a set of *Units,* material *Streams* and information on the topology of the simulated process. All units have access to the library of built-in *Equations* solvers.

The most important of the *Auxiliary modules* are the following. *Material database* that organizes access to materials and their properties and allows manipulating them using graphical user interface (GUI). Constant, temperature- and pressure-dependent parameters are supported, as well as properties describing the interaction of two different materials, such as interface tension. For more details, see [16]. The *Models manager* module acts as an interface between the simulation system and units and solvers from external libraries. Finally, the *Data storage* subsystem organizes writing and reading of data on a hard drive and a runtime data caching.

Models of solvers and units are organized as shared libraries (*.dll or *.so). They can be developed separately from Dyssol and dynamically linked at runtime. To simplify the creation of new models, the simulation system is equipped with a *Models creator* tool. It includes templates of new modules, examples of their implementation, and provides access to the programming interfaces (*API*) necessary to develop a fully-functional units or solvers.

The software provides a simplified *Graphical user interface* to configure units and streams connections and to perform initial results analysis. Moreover, there exist *Command-line interface*, which can be used to run simulations in batch mode. More detailed description of each module of the simulation system can be found in [8] and [16].

2.3. Implementational aspects

Dyssol is implemented with C++ programming language using several external libraries and tools. The graphical user interface is based on the Qt library and Qt software framework [17]. The storage subsystem uses the hierarchical data format HDF5 and the HDF5 library [18]. This ensures portability of the simulation framework and allows the use of external packages to analyse and edit Dyssol files. Built-in equation solvers are implemented using
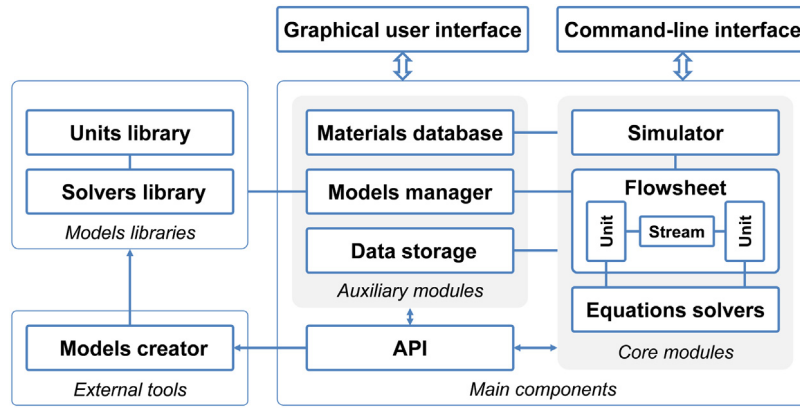
Fig. 1. Simplified representation of the architecture of the simulation system Dyssol.

the IDA [19] and KINSOL [20] solvers from the SUNDIALS [21] package. The KISS FFT library [22] is utilized to perform a fast Fourier transform, needed in some units and solvers. The Dyssol installation package for the Windows platform is assembled using the Inno Setup [23] utility. The *Models creator* tool is based on a preconfigured solution for Visual Studio 2015, which includes projects with required API libraries and templates of units.

The main version of the programme is developed for Windows, where the graphical and command-line versions of the user interface are available. Microsoft Visual C++ 14.0 is used to compile and build Dyssol for Windows. For Linux, only the command-line version exists, which is build using g++-7 compiler.

2.4. Software functionalities

Dyssol allows to investigate the dynamics of complex production processes applying extended description of granular materials with multidimensional distributed parameters and their advanced treatment with the help of transformation matrices. Solid, liquid and vapour phases are described by a set of overall (mass flow, temperature, pressure), concentrated (phase fraction) and distributed (compound fraction) parameters. For the solid phase, it is possible to extend the description with additional dependent distributed parameters, such as porosity, moisture content or form factor.

Available standardized interfaces for implementing new models and solvers allow one to expand the system with new modules. The current installation package (version v0.10) includes a set of the following steady-state and dynamic models: agglomerator [24], bunker, crushers (e.g. [9,25]), cyclone [26], granulator [27], heat exchanger, mixer, screens [28–31], splitter, time delay. In addition to the solvers from the SUNDIALS package, Dyssol contains three solvers of agglomeration population balance equations: cell average [32], fixed pivot [33] and FFT-based [24].

It is worth mentioning that the current version has some limitations compared to other commercially available products. Therefore, in the next releases of Dyssol, the following challenges are to be addressed:

- Improving GUI for visual configuration of flowsheets and extended results analysis.
- Extension of the materials database by connecting to existing databanks and incorporating tools for thermodynamic calculations.
- Adding the ability to handle multi-component particles.
- Extension of the model library with new relevant units.
- Simplification of the way of expanding the system with new models.

Table 1
Main model parameters used for milling process.

	Inlet 1	Inlet 2
Mass flow	0.1 kg/s	until 4 h: 0 kg/s
		after 4 h: 0.3 kg/s
μ_{size}	200 mm^3	350 mm^3
σ_{size}	20 mm^3	30 mm^3
$\mu_{moisture}$	0.2	0.4
$\sigma_{moisture}$	0.04	0.02
	Screen 1	Screen 2
Separation diameter	268.1 mm^3	143.8 mm^3
Separation sharpness	until 2 h: 10	until 2 h: 10
	after 2 h: 3.5	after 2 h: 3.5
	Mill 1	Mill 2
Hold-up mass	200 kg	100 kg
x_{lim}	520 mm^3	610 mm^3
k	9.5	5.5
ϕ	0.5	0.9
ψ	4.1	4.5
ϑ	2.5	3.2

- Improving performance by introducing advanced parallelization techniques.

3. Illustrative example

To illustrate the usage of Dyssol for complex processes, we analyse the dynamics of the sieve-mill circuit. The flowsheet structure is shown in Fig. 2 (left) and main model parameters are listed in Table 1. Screens are calculated according to the steady-state model of Molerus and Hoffmann [28]. Mills are dynamic units, developed in [9], which apply Austin & Luckie [34] selection $S(x)$ and Austin [35] breakage $B(x, y)$ functions in form

$$S(x) = \left(\frac{x}{x_{lim}}\right)^k, \tag{1}$$

$$B(x, y) = \frac{\phi \cdot \psi}{y} \cdot \left(\frac{x}{y}\right)^{\psi-2} + (1 - b1) \cdot \frac{\vartheta}{y} \cdot \left(\frac{x}{y}\right)^{\vartheta-2}, \tag{2}$$

where x and y are volumes of particles, x_{lim} is the maximum particle size, and k, ϕ, ψ and ϑ are model parameters. Both units implement the approach with transformation matrices [9].

Solid material is described by two interdependent distributions: 200 classes of particle volumes ranging from 0 to 500 mm^3 and moisture content on interval 0 to 50% divided into 50 classes. All distributed parameters are described with Gaussian function with mean value μ and standard deviation σ. The first inlet continuously supplies small dry particles during the whole process. The second inlet is switched on 4 h after the start of the
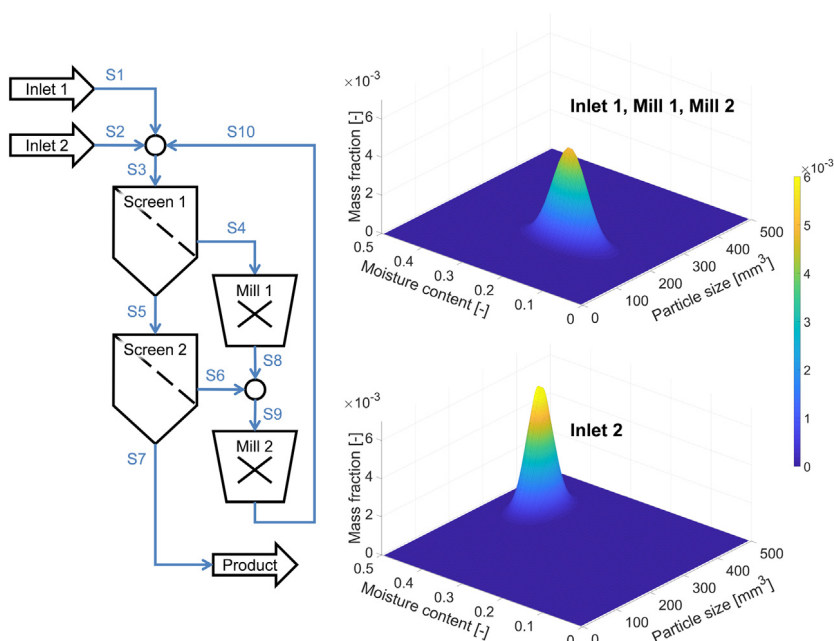
Fig. 2. Flowsheet of the milling process (left) and initial material distributions in the inlets and the mills' holdups (right).

simulation and adds large wet particles to the system. Initial holdup material in both mills has the same distribution as Inlet 1. Additional dynamics in the process arise due to a change in the operating mode of screens, in which the separation sharpness changes after 2 h of process time. During the simulation, it is assumed that there is no drying and no moisture exchange between the particles.

Fig. 3 shows the dynamic response of the process to changes in process conditions. The steady state is almost achieved after two hours. At this time point, the parameters of screens are modified and, as a consequence, the material mass flows and the particle sizes change rapidly. However, after an additional hour, a new steady state is reached. Finally, at the time point of 4 h, the second inlet begins to supply large particles to the system. As a result, parameters of the material change again, and the process stabilizes within the next hour.

The time-dependent change of particle distributions is shown in Fig. 4. The size distribution responds to the variations in the process parameters, and also changes with time due to the operation of the mills. One can see large disturbances of particle sizes at time points 0, 2 h and 4 h, when parameters of the process change rapidly, followed by a gradual stabilization. At the same time, changes in the dependent distributed parameter are caused only by the additional supply of a new material at time point 4 h. For the rest of the time, the moisture content remains constant.

The simulation results demonstrate that the approach with transformation matrices implemented in Dyssol allows to avoid the loss of information about secondary distributed parameters that are not explicitly included in the model equations. Despite the fact that the screens and mills were designed considering only the particle size distribution, the moisture content was implicitly handled and correctly calculated.

The calculation of this flowsheet with two-dimensional distribution of material parameters (total $200 \times 50 = 10000$ classes) until the first steady state takes around 1146 s. A total of 26011 time points is generated in 8 material flows, which corresponds to 44.1 ms per time point. Calculating the case but only with a one-dimensional distribution (200 classes by size), takes about 48 s.

Totally 8896 time points are generated, which is 5.4 ms per time point. All measurements were made on the computer equipped with CPU i7-7700K. More information on Dyssol performance and the effect of various parameters can be found in [8].

4. Impact

The main advantages and distinctive features of the Dyssol framework are:

- Dynamic simulation of complex process structures, consisting of interconnected heterogeneous units.
- Application of the sequential-modular approach that simplifies extension of the system.
- Utilization of the waveform relaxation method supplemented by convergence and extrapolation methods for dynamic calculations of process structures with recycling streams.
- Description of the solid phase with multidimensional interdependent distributed parameters.
- Proper handling of granular materials using the transformation matrix approach.
- Possibility to efficiently process interdependent distributed parameters with fine grids and a large number of classes due to dynamic data caching and special data formats.
- General applicability for various dynamic and steady state processes.
- Standardized interfaces for extending the system with new units and solvers.
- High modularity and expandability of the system due to its modular structure.
- Open-source BSD licencing.

Several research projects were already carried out using the Dyssol system. It was applied to numerically analyse the dynamics of a chemical looping combustion [36,37]. To simulate the process, dynamic models of a circulating fluidized bed, a bubbling fluidized bed and a syphon, and a steady-state cyclone model were developed. In this project, several simulation studies of the transient behaviour of the process were conducted using Dyssol.
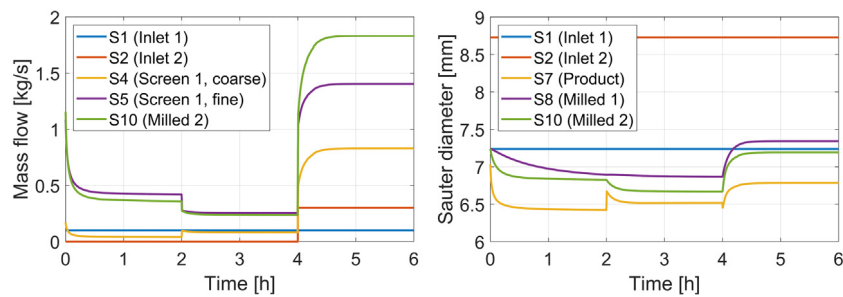
Fig. 3. Response of the milling circuit to variation of process parameters.
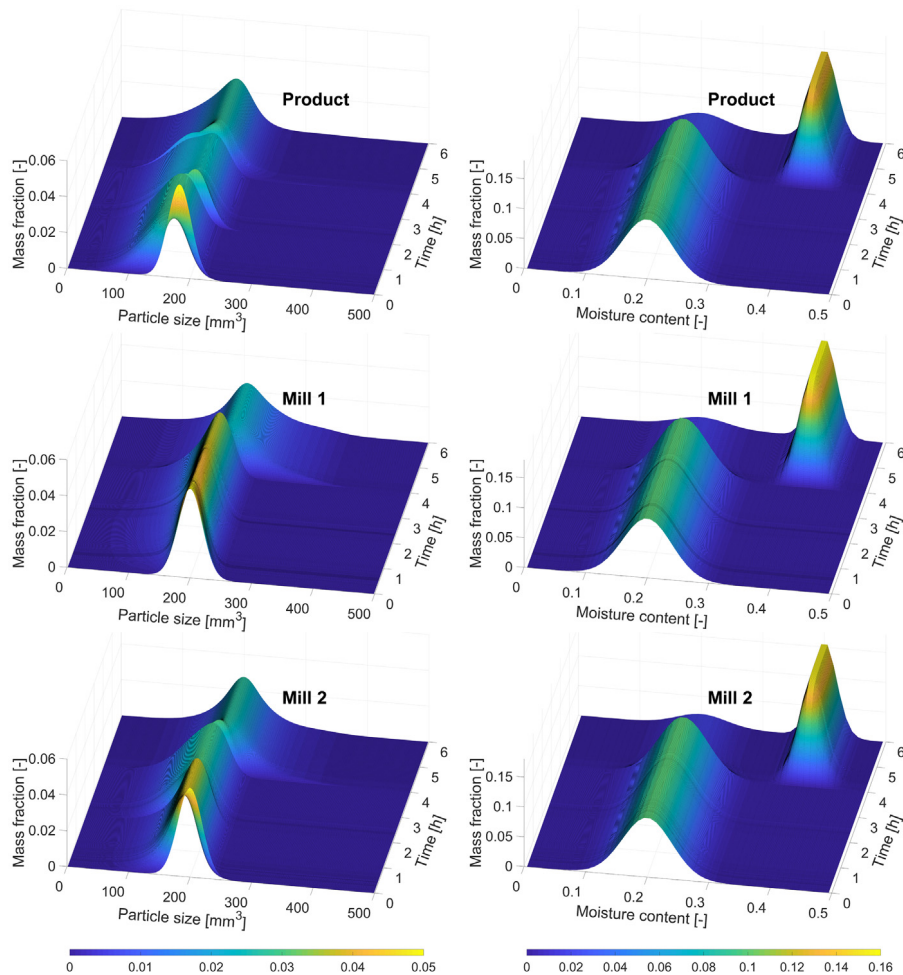

Fig. 4. Dynamic change in particle size (left) and moisture content (right) during simulation for the product and mills' holdups.

It was found that all developed models are generally applicable in a specific parameters range. The performed numerical calculations showed good agreement between the simulation results and experimental data.

Dyssol was also used to study the processes of particle formation during precipitation crystallization [38]. A confined impinging jet mixer model was developed by combining a newly proposed limiting time scale approach with a population balance solver. This model was investigated in a combined framework consisting of Dyssol, MATLAB, and PHREEQC [39] to study the effect of supersaturation and mixing on the precipitation processes. Based on the data obtained during the work, several versions

of more complex stirred tank reactor models for precipitation processes were developed and simulated in Dyssol.

In [24], a new approach to solving a one-dimensional PBM for agglomeration process was proposed and investigated. The new method assumes using of the separable approximation of the agglomeration kernel and the subsequent fast Fourier transform to speed up the calculations. Using Dyssol, the new method was studied and compared with the well-known fixed pivot technique, and its applicability to dynamic modelling of continuous agglomeration processes with complex structures was analysed. Simulation studies have shown that the new method can provide a more effective solution if certain restrictions are met.

In [40], a generic flowsheet model for precipitation of sparingly soluble salts was developed and implemented. Dyssol, coupled with the PHREEQC hydrochemistry software, was used as a model development environment and as a simulation tool. To verify the model, a comparison was made between the modelling and experimental precipitation of several compounds. Experimental and model studies have shown good agreement between the results for both simple unidirectional circuits and complex schemes with recirculation of material.

New approaches to formulating transformation matrices from population balance equations (PBE) for grinding and agglomeration processes were proposed in [9]. The finite volume method for spatial discretization and the second-order Runge–Kutta method to obtain the complete discretized form of the PBE were used to derive transformation laws. Applying the proposed approach, dynamic units of a mill and an agglomerator were developed. Using Dyssol, the resulting models were numerically investigated applied to an exemplary production process.

Since several years, the Dyssol software is intensively used in the educational process at Hamburg University of Technology. It is applied to study the basics of steady-state and dynamic modelling as part of the courses "Particle Technology" and "Multiscale Simulation of Granular Materials".

5. Conclusions

An open source platform for flowsheet simulations of complex dynamic processes involving particulate materials, named Dyssol, has been presented. Important features of the software are the high flexibility, customizability and extensibility of the system, which provides the widest application scope.

Thanks to the advanced description of the solid phase using multidimensional distributed parameters, as well as the proposed methods for their handling utilizing transformation matrices, this system provides ample opportunities for studying various processes involving granular materials.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

References

[1] Aspen plus. 2020, http://www.aspentech.com/en/products/engineering/aspen-plus, [Accessed 16 April 2020].

[2] PSE Products – gPROMS FormulatedProducts. 2020, http://psenterprise.com/products/gproms/formulatedproducts, [Accessed 16 April 2020].

[3] CHEMCAD – Chemical engineering simulation software by chemstations. 2020, www.chemstations.com/CHEMCAD/, [Accessed 16 April 2020].

[4] Rossiter AP, Douglas JM. Design and optimization of solids processes. Part 1. A hierarchical decision procedure for process synthesis of solids systems. Chem Eng Res Des 1986;64:175–95.

[5] Hartge EU, Pogodda M, Reimers C, Schwier D, Gruhn G, Werther J. Flowsheet simulation of solids processes. KONA 2006;24. http://dx.doi.org/10.14356/kona.2006017.

[6] Dosta M, Litster J, Heinrich S. Flowsheet simulation of solids processes: Current status and future trends. Adv Powder Technol 2020;31(3):947–53. http://dx.doi.org/10.1016/j.apt.2019.12.015.

[7] Dosta M, Heinrich S, Werther J. Fluidized bed spray granulation: Analysis of the system behaviour by means of dynamic flowsheet simulation. Powder Technol 2010;204:71–82. http://dx.doi.org/10.1016/j.powtec.2010.07.018.

[8] Skorych V, Dosta M, Hartge E-U, Heinrich S. Novel system for dynamic flowsheet simulation of solids processes. Powder Technol 2017;314:665–79. http://dx.doi.org/10.1016/j.powtec.2017.01.061.

[9] Skorych V, Das N, Dosta M, Kumar J, Heinrich S. Application of transformation matrices to the solution of population balance equations. Processes 2019;7(8). http://dx.doi.org/10.3390/pr7080535.

[10] DFG-Priority programme SPP 1679. 2020, http://dynsim-fp.de/en/home/, [Accessed 16 April 2020].

[11] Sargent RWH, Westerberg AW. SPEED-UP In chemical engineering design. Trans Inst Chem Eng 1964;42:190. http://dx.doi.org/10.13140/2.1.2585.6001.

[12] Hillestad M, Hertzberg T. Dynamic simulation of chemical engineering systems by the sequential modular approach. Comput Chem Eng 1986;10:377–88. http://dx.doi.org/10.1016/0098-1354(86)87008-9.

[13] Dosta M. Modular-based simulation of single process units. Chem Eng Technol 2019;42(3). http://dx.doi.org/10.1002/ceat.201800671.

[14] Wegstein JH. Accelerating convergence of iterative processes. Commun ACM 1958;1(6):9–13. http://dx.doi.org/10.1145/368861.368871.

[15] Mathews JH, Fink KD. Numerical methods using MATLAB. 3rd ed. Prentice Hall; 1999, p. 90–100.

[16] Skorych V, Buchholz M, Dosta M, Heinrich S. Dyssol – a framework for dynamic simulation of interconnected solids processes. In: Heinrich S, editor. Dynamic flowsheet simulation of solids processes. Springer International Publishing; 2020, http://dx.doi.org/10.1007/978-3-030-45168-4.

[17] Qt – Cross-platform software development for embedded & desktop. 2020, https://www.qt.io/, [Accessed 11 May 2020].

[18] The HDF5® library & file format. 2020, http://hdfgroup.org/solutions/hdf5/, [Accessed 16 April 2020].

[19] Hindmarsh AC, Serban R, Collier A. User Documentation for IDA V3.1.0 (SUNDIALS V3.1.0). Center for Applied Scientic Computing Lawrence Livermore National Laboratory; 2017.

[20] Collier AM, Hindmarsh AC, Serban R, Woodward CS. User Documentation for KINSOL V3.1.0 (SUNDAILS V3.1.0). Center for Applied Scientific Computing Lawrence Livermore National Laboratory; 2017.

[21] SUNDIALS: SUite of Nonlinear and DIfferential/ALgebraic Equation Solvers. 2020, http://computing.llnl.gov/projects/sundials, [Accessed 16 April 2020].

[22] A Fast Fourier Transform (FFT) library KISS FFT. 2020, http://github.com/mborgerding/kissfft, [Accessed 16 April 2020].

[23] Inno Setup. 2020, http://jrsoftware.org/isinfo.php, [Accessed 16 April 2020].

[24] Skorych V, Dosta M, Hartge E-U, Heinrich S, Ahrens R, Le Borne S. Investigation of an FFT based solver applied to dynamic flowsheet simulation of agglomeration processes. Adv Powder Technol 2019;30(3):555–64. http://dx.doi.org/10.1016/j.apt.2018.12.007.

[25] Bond FC. Crushing and grinding calculations. Part I. Br Chem Eng 1961;6(6):378–85.

[26] Muschelknautz E. Die Berechnung von Zyklonabscheidern für Gase. Chem Ing. 1972;44(1/2):63–71. http://dx.doi.org/10.1002/cite.330440112.

[27] Heinrich S, Peglow M, Ihlow M, Henneberg M, Mörl L. Analysis of the start up process in continuous fluidized bed spray granulation by population balance modelling. Chem Eng Sci 2002;57:4369–90. http://dx.doi.org/10.1016/S0009-2509(02)00352-4.

[28] Molerus O, Hoffmann H. Darstellung von windsichtertrennkurven durch ein stochastisches modell. Chem Ing Tech 1969;41(5+6):340–4. http://dx.doi.org/10.1002/cite.330410523.

[29] Plitt LR. The analysis of solid–solid separations in classifiers. CIM Bull 1971;64(708):42–7.

[30] Hennig M, Teipel U. Stationäre Siebklassierung. Chem Ing Tech 2016;88:911–8. http://dx.doi.org/10.1002/cite.201500161.

[31] Radichkov R, Müller T, Kienle A, Heinrich S, Peglow M, Mörl L. A numerical bifurcation analysis of continuous fluidized bed spray granulation with external product classification. Chem Eng Process 2006;45:826–37. http://dx.doi.org/10.1016/j.cep.2006.02.003.

[32] Kumar J, Peglow M, Warnecke G, Heinrich S. An efficient numerical technique for solving population balance equation involving aggregation, breakage, growth and nucleation. Powder Technol 2008;182(1):81–104. http://dx.doi.org/10.1016/j.powtec.2007.05.028.

[33] Kumar S, Ramkrishna D. On the solution of population balance equations by discretization – I. A fixed pivot technique. Chem Eng Sci 1996;51(8):1311–32. http://dx.doi.org/10.1016/0009-2509(96)88489-2.

[34] Austin LG, Luckie PT. The estimation of non-normalized breakage distribution parameters from batch grinding tests. Powder Technol 1971;5(72):267–71. http://dx.doi.org/10.1016/0032-5910(72)80030-5.

[35] Austin LG. A discussion of equations for the analysis of batch grinding data. Powder Technol 1999;106:71–7. http://dx.doi.org/10.1016/S0032-5910(99)00047-9.

[36] Haus J, Hartge E-U, Heinrich S, Werther J. Dynamic flowsheet simulation of gas and solids flows in a system of coupled fluidized bed reactors for chemical looping combustion. Powder Technol 2017;316:628–40. http://dx.doi.org/10.1016/j.powtec.2016.12.022.

[37] Haus J, Hartge E-U, Heinrich S, Werther J. Dynamic flowsheet simulation for chemical looping combustion of methane. Int J Greenhouse Gas Control 2018;72:26–37. http://dx.doi.org/10.1016/j.ijggc.2018.03.004.

[38] Metzger L. Process simulation of technical precipitation process – The influence of mixing. Karlsruhe: KIT Scientific Publishing; 2018, http://dx.doi.org/10.5445/KSP/1000076187.

[39] PHREEQC Version 3. 2020, http://usgs.gov/software/phreeqc-version-3, [Accessed 27 April 2020].

[40] Rehage H, Scherer S, Kind M. A steady-state precipitation model for flowsheet simulation and its application. Comput Chem Eng 2019;128:524–37. http://dx.doi.org/10.1016/j.compchemeng.2019.06.030.