Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-1 (2015) 011–016

Automatic Model Reduction of Linear Population Balance Models by Proper Orthogonal Decomposition

Dmytro Khlopov * **Michael Mangold** *

* *Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany (e-mail: khlopov@mpi-magdeburg.mpg.de, mangold@mpi-magdeburg.mpg.de)*

Abstract: This paper discusses the use of Proper Orthogonal Decomposition (POD) for the model reduction of particle processes in fluid flow described by Population Balance Equations (PBEs). This class of processes is very important for chemical engineering. As detailed models of such processes turn out to be very complicated, POD is an attractive way to obtain reduced models of low order. This paper reports on the automatization of the mentioned method. An automatic procedure for the reduction of PBE models is presented, which is implemented in the modeling and simulation environment ProMoT/Diana. Currently, the procedure is applicable to linear models, but is extendable to nonlinear models, as well.

Keywords: crystallization, population balance equations, linear model reduction, proper orthogonal decomposition, computer-aided modeling, differential algebraic systems, dynamic simulation

1. INTRODUCTION

Particle processes are very important for chemical and pharmaceutical industries. The vast majority of products in these industries is produced in particulate form (Wintermantel (1999)). Main examples of such processes are crystallization, granulation, and polymerization.

Particle processes typically consist of the actual dispersed phase and a fluid phase, which stands in exchange with the dispersed phase. Essential processes within the dispersed phase are formation of particles (nucleation), their division into smaller particles (breakage) and union of particles into larger units (aggregation). The dispersed phase with a large number of particles can be described by population balance equations (Ramkrishna (2000)). Particulate systems in general form can be expressed by the following set of model equations (see John et al. (2009), Jürgen Koch (2008), Öncül et al. (2006)):

- population balance of the dispersed phase

$$\frac{\partial n}{\partial t} + \underbrace{\frac{\partial}{\partial L_j}(G_j n)}_{growth} + \underbrace{\frac{\partial}{\partial x_k}(\nu_k n + j_k^n)}_{convection/diffusion} = \underbrace{\int_{\Omega_L} h_{br} dV_L}_{breakage} + \underbrace{\int_{\Omega_L} h_{agg} dV_L}_{agglomeration} \quad (1)$$

In Eq. (1) n is the number density function of the particle population. This function depends on time, and in addition on external x_k and internal L_j coordinates which usually describe a geometry configuration and some properties of a system, respectively. The function G_k denotes the growth rate of particles in the direction of internal coordinate L_j. ν_k is the fluid velocity in the direction of internal coordinate L_j and J_k^f is the diffusion flow in the direction of external coordinate x_k. The nonlinear functions h_{br} and h_{agg} denote breakage and agglomeration of particles respectively.

- mass and energy balances of the continuous phase

$$\frac{\partial(\rho s_l)}{\partial t} + \underbrace{\frac{\partial}{\partial x_k}(\rho \nu_k s_l + j_{lk}^s)}_{convection/diffusion} = \underbrace{\int_{\Omega_L} J_l^s dV_L}_{exchange\ with\ dispersed\ phase} + \underbrace{\sigma_L(s)}_{chemical\ reaction} \quad (2)$$

In Eq. (2) ρ denotes a mass density of the fluid phase. s_l describes a specific state (internal energy, mass fraction). j_{lk}^s denotes a diffusion flow and J_l^s is an exchange with the dispersed phase.

- momentum balances of the continuous phase

$$\frac{\partial(\rho \nu_m)}{\partial t} + \underbrace{\frac{\partial}{\partial x_k}(\rho \nu_k \nu_m + \pi_{mk})}_{convection/diffusion} = \underbrace{\int_{\Omega_L} J_m^\nu dV_L}_{exchange\ with\ dispersed\ phase} + \rho g_m \quad (3)$$

where π_{mk} is a stress tensor.

The numerical solution of such equation system (1)-(3) with several internal and external coordinates is hardly possible in real time. To enable model-based process design and process control of systems with particle populations in fluid flow, there is a need for reduced models. The

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.05.019

reduced models should have much lower system order than the reference models. Their numerical solutions should be much easier and faster. Also the reduced models should be able to reproduce the system behavior with sufficient accuracy in the relevant window of operation conditions and in a relevant range of system parameters.

In this paper Proper Orthogonal Decomposition is used for the development of an automatic procedure for model reduction and is applied to the linear model of a fluidized crystallizer. This method has been successfully used during the last years for the nonlinear model reduction of population balance systems like crystallizers (Krasnyk and Mangold (2010), Mangold et al. (2014a)), granulators (Mangold (2012)).

The basic idea of this method is to approximate the model solution by a linear combination of time independent basis functions weighted by time dependent coefficients. The basis functions are constructed from numerical simulation results of a detailed reference model. The reduced model consists of ordinary differential equations in time. The reference model will be presented in the next section. The model reduction method is discussed in Section 3. Section 4 describes technical details of the developed tool for automatic model reduction which is based on the modeling and simulation environment ProMoT/Diana.

2. LINEAR MODEL OF A FLUIDIZED BED CRYSTALLIZER

2.1 Reference Model

Fluidized bed crystallizers are an innovative method to continuously produce crystals with high purity (Tung et al. (2009)). In the following, the model of the fluidized crystallizer sketched in Fig. 1 will be considered. The crystallizer has the shape of a cylinder whose diameter narrows towards the crystallizer's bottom. An input volume flow of the fluid comes from outside the model and enters the bottom of the crystallizer. In the crystallizer, the fluid flow goes from bottom to top. Small particles are dragged upwards with the fluid. Larger particles sink to the bottom due to gravity.

The reference model for this process is a variant of the models described in Palis et al. (2013), Mangold et al. (2014b). The main model assumptions are:

- Spatial gradients perpendicular to the external coordinate x in the direction of the fluid flow are negligible.
- The fluid flow is a plug flow with flow velocity v_f.
- The number of particles is sufficiently high that the particle phase may be described by a particle population with a number size density $n(x, L, t)$ denoting the number of particles with size L per volume at a point x in space.
- The gravity force, the buoyancy force, and the drag force acting on a partcile are in equilibrium.

The population balance equation of the system reads:

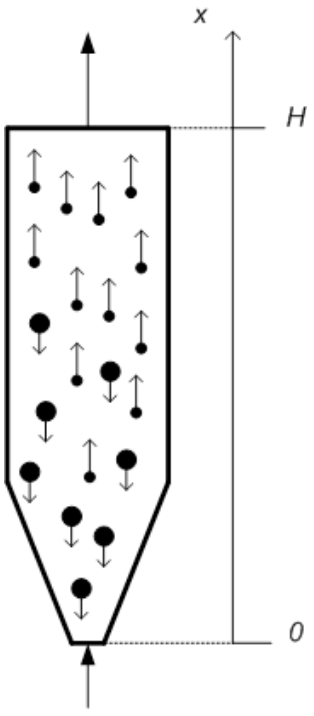

Fig. 1. Scheme of a fluidized crystallizer

$$A(x)\frac{\partial n}{\partial t}\bigg|_{x,L,t} = -\frac{\partial}{\partial x}(A(x)v_p(x,L)n(x,L,t)) + D\frac{\partial}{\partial x}(A(x)\frac{\partial n}{\partial x}\bigg|_{x,L,t}) \quad (0 < x < H, 0 < L, t > 0) \tag{4}$$

with boundary conditions

$$v_p(0,L)n(0,L,t) - D\frac{\partial n}{\partial x}\bigg|_{0,L,t} = 0 \tag{5}$$

$$\frac{\partial n}{\partial x}\bigg|_{H,L,t} = 0 \tag{6}$$

and initial conditions

$$n(x,L,0) = n_0(x,L) \tag{7}$$

The first term on the right-hand side of (4) is the advective transport of particles with velocity v_p; $A(x)$ denotes the cross-sectional area of the crystallizer. An expression for v_p follows from the assumption of an equilibrium of forces

$$0 = F_G + F_B + F_D \tag{8}$$

where $F_G = -\frac{\pi}{6}L^3\rho_P g$ is the gravity force, $F_B = \frac{\pi}{6}L^3\rho_F g$ is the buoyancy force, and $F_D = \frac{(v_P - v_F)^2}{2}\rho_F c_W \frac{\pi}{4}L^2$ is the drag force. One obtains from (8)

$$v_p = v_f - \sqrt{\frac{4}{3}\frac{Lg}{c_W}\frac{\rho_P - \rho_F}{\rho_F}} \tag{9}$$

The fluid flow velocity v_f is calculated from

$$v_f = \frac{\dot{V}}{A(x)} \tag{10}$$

where $\dot{V}$ is the volume flow of the fluid.

For the drag coefficient c_W, the correlation

$$c_W = \frac{A_{cw}}{Re} + \frac{B_{cw}}{\sqrt{Re}} + C_{cw} \tag{11}$$

with $Re = \frac{v_f d}{\nu}$ is used [Palis et al. (2013)], where A_{c_W}, B_{c_W}, C_{c_W} are empirical constant parameters.

The second term on the right-hand side of (4) stands for particle transport by dispersion.

2.2 Discretization

In order to apply a numerical method to solve the reference system, it is transformed into a spatially discretized form using the method of lines. For the numerical solution of the reference system, a finite volume scheme is applied. The external coordinate x and the internal coordinate L are discretized. An equidistant grid is used in x direction, and a logarithmic distribution of grid points in L direction. Gradients in x direction are approximated by central differences because the particle velocity v_p may change its sign along the x coordinate and a discretization scheme is needed that provides numerical stability under these circumstances.

After discretization a set of ordinary differential equations is obtained for approximation of the population balance equations (4), which may be written as

$$\frac{dn(t)}{dt} = L_n n(t), \tag{12}$$

where $n(t)$ is the discretized state vector, and L_n denotes a system matrix of the reference model.

3. MODEL REDUCTION

3.1 POD method

The Proper Orthogonal Decomposition Method (Kunisch and Volkwein (2003), H. M. Park (1996), Sirovich (1987)) can be subdivided roughly into two steps. The first step, the offline phase, includes preliminary calculations, like the solution of the reference model and the generation of the reduced model equations. Depending on the complexity of the reference model the offline phase can be extremely computationally intensive. But this investment pays off in the second fast and cheap step, the online phase. In the online phase only a system of ordinary differential equations of very low order has to be solved.

As a starting point, the detailed reference model has to be solved numerically. Snapshots for each state of the model $n(t_1), n(t_2), ...$ are collected and arranged as a matrix $N = (n(t_1), n(t_2), ...)$.

A reduced basis for the snapshot vectors is constructed from the singular value decomposition

$$N = U\Sigma V^T \tag{13}$$

The new basis vectors are taken as

$$\psi_i^n = U_i, \qquad i = 1, ..., N^n, \tag{14}$$

where U_i denotes the i th column of U, and N^n is the dimension of the reduced basis and correspondingly the order of the resulting reduced model.

The state vector $n(t)$ can be approximated by the following expression:

$$n(t) \approx \psi^n \phi^n(t) \tag{15}$$

where $\psi^n = (\psi_1^n, ..., \psi_{N^n}^n)$, and $\phi^n(t)$ is the coefficient vector of the reduced basis; $\phi^n(t)$ will become the state vector of the reduced model.

In order to obtain equations for $\phi^n(t)$, the approximation for the state vector (15) is inserted into the discretized population balance equation (12). To make the projection of the residuals on the reduced basis vanish, Galerkin's method of weighted residuals is applied, which leads to

$$\frac{\partial \phi^n(t)}{\partial t} = \underbrace{\psi^{nT} L_n \psi^n}_{=:L'} \phi^n(t) \tag{16}$$

The matrix L' on the right-hand side of (16) has to be evaluated only once for a fixed reduced basis, because it is not depending on the reduced state vector $\phi^n(t)$. During the online phase of POD, L' is just a constant matrix.

3.2 Simulation Results

Simulation of the reference model was performed to obtain snapshots for model reduction. Following discretization grid was applied: 160 points in the direction of external coordinate (position along the crystallizer) and 80 point in the direction of internal coordinate (size of particles). In total, we obtain a system with 12800 ordinary differential equations. The snapshots are solutions of the detailed model on a equidistant time grid for $t = 0..1000$ seconds with interval of 10 seconds.

Figures 2 and 3 show a dynamic behavior of the reference model in the upper and in the lower parts of the crystallizer respectively. Initially, the particle population is located in the upper part of the crystallizer. Large particles sink faster to the ground than small particles. Therefore, the large particles in the population in Fig. 2 vanish first, while small particles are present until the end of the simulation. The opposite situation is found at the point shown in Fig. 3, which is close to, but not exactly at the bottom. Initially, this part of the crystallizer is free of particles. After some time, large particles appear, and disappear again, because they sink further to the ground. Smaller particles arrive at a later time, as their sinking velocity is smaller.

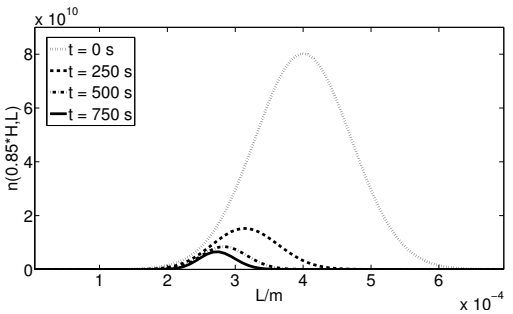

Fig. 2. Dynamic behavior of particles at the top of the crystallizer

The reduced model consist of 24 ordinary differential equations, compared to 12800 differential equations of the

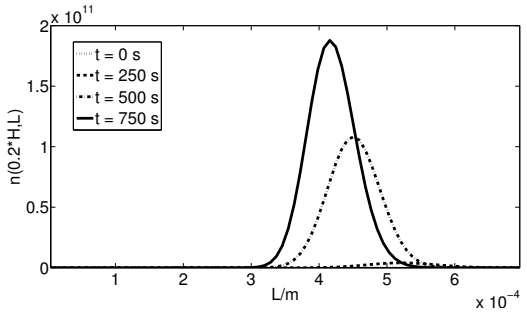

Fig. 3. Dynamic behavior of particles at the bottom of the crystallizer

reference model. The reduced model and the reference model agree very well, as is illustrated by Figure 4. The Figure shows the relative total error $||n(t) - \hat{n}||/||n(t)||$, where $\hat{n}(t)$ is the approximation of the reduced model.

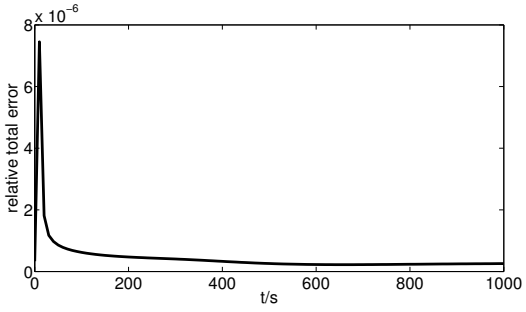

Fig. 4. Relative total error of approximation

4. AUTOMATIC PROCEDURE FOR MODEL REDUCTION

An automatic procedure for the reduction of linear models is implemented in the modeling and simulation environment ProMoT/Diana.

ProMoT is a modeling tool written in Common Lisp with a graphical user interface written in Java (Ginkel et al. (2003), Mirschel et al. (2009)). ProMoT supports the structured implementation of dynamic models described by systems of nonlinear implicit differential algebraic equations. ProMoT itself is a purely symbolic modeling tool and hence has no restriction with respect to numerical properties of the models. On the ProMoT level the idea is to keep the model formulation separate from numerical requirements. ProMoT provides a general modeling language and outputs to different numerical solution tools.

Diana (Krasnyk (2009)) is a simulation tool for the solution and nonlinear analysis of differential algebraic systems, as they typically result from first principle modeling of chemical engineering systems and biochemical systems. The numerical core of Diana is written in `C++` in order to ensure fast and efficient numerical solutions. Model equations also have to be implemented in `C++` as an equation set object (ESO) using CAPE-OPEN standard interfaces. Usually, the model implementation is done automatically by ProMoT. For the numerical analysis, the modeler accesses Diana via scripts written in the scripting language Python. The advantage is that Python is more user friendly than `C++` code.

The developed automatic tool for model reduction is a part of the ProMoT project and hence is written in Common Lisp. One uses Diana simulation tool only as an intermediate step for the numerical solution of the reference model. The main parts of the tool so far are the snapshots reader, the generator of basis functions, the extractor of the system matrix of the reference model and the generator of reduced model. The structure of the automatic tool for model reduction is sketched in Fig. 5.

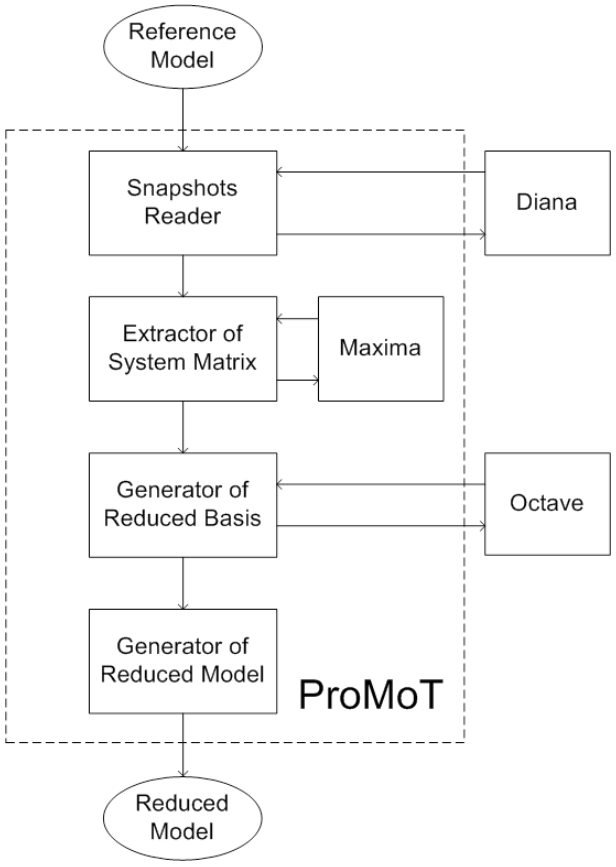

Fig. 5. Structure of automatic tool for model reduction

4.1 Snapshots Reader

The automatic procedure for model reduction begins with the creation of the snapshots reader's instance. The user has to provide the name of a Python script which contains all information about simulation conditions. Via this script, the user has to define model parameter values, a time range, and an output time interval for collecting snapshots to which the POD method is to be applied.

After reading of all necessary information, the snapshots reader runs the `C++` code generator for Diana and translates the reference model for which we want to produce a reduced model. Then the snapshots reader runs Diana to generate snapshots and writes them into an output file. The last step of this part is to read the snapshots from the output file into the ProMoT environment and to arrange them into a matrix $N = (n(t_1), n(t_2), ...)$, where $n(t_1), n(t_2), ...$ are the states of the reference model.

4.2 Extractor of System Matrix of Reference Model

When the work of the snapshots reader is finished, the automatic tool starts building the system matrix of the detailed reference model. For these purposes the instance of system matrix extractor has been developed. The main idea in extracting the system matrix is a search of coefficients in the ordinary differential equations which correspond to the desired state variables of the reference model. The general form of an element of the system matrix of a linear autonomous system can be expressed by the following equation:

$$L_{n\ i,j} = \frac{\partial ODE_i}{\partial s_j} \tag{17}$$

In Eq. (17) $L_{n\ i,j}$ is an element of the system matrix at i th row and at j th column. ODE_i is the right-hand side of the ordinary differential equation corresponding to i th state variable, s_j is an symbolic name of j th state variable.

In ProMoT all modeling data is stored in symbolic form. To preserve generality and to preserve the dependency of the reduced model on the model parameters, the system matrix has to keep the elements in a symbolic form. For this purpose the computer algebra system Maxima is used. Since it is written in Common Lisp and can be called directly from Lisp code, Maxima is embedded into the ProMoT core. For convenience of use of the computer algebra system a program interface between ProMoT and Maxima has been developed. The interface allows to convert internal data structures of ProMoT into corresponding Maxima representation and vice versa.

4.3 Generator of Basis Functions

In order to generate the equations of reduced model, it is necessary to calculate the reduced basis of model. A complicated mathematical apparatus is needed to accomplish this, in particular to calculate singular value decomposition of snapshots matrix. For these purposes it was decided to use a specialized software as an external tool. GNU Octave is a high-level interpreted language, primarily intended for numerical computations. To use this numerical tool externally a program interface between ProMoT and Octave has been developed. The interface allows to run Octave in interactive mode, send and receive data, and to apply all the built-in mathematical functions of Octave.

4.4 Generator of Reduced Model

After completion of the above parts it is possible to calculate the system matrix L' of the reduced model from Eq. (16). The generator of the reduced model creates a new modeling file into which it writes the set of ordinary differential equations, which can be expressed in the following form:

$$\frac{d\phi_i^n(t)}{dt} = \sum_{j=1}^{N^n} L'_{i,j}\phi_j^n(t) \tag{18}$$

In Eq. (18) ϕ_j^n is a coefficient of the reduced basis and N^n denotes the number of basis functions.

For reconstruction of actual states of the given model $n(t_1), n(t_2), ...$ one has to evaluate Eq. (15).

5. OUTLOOK

The automatic tool for reduction of linear models has been developed by using proper orthogonal decomposition (POD). The implemented approach has to be extended for nonlinear systems. For these purposes, automatic splitting of the ride-hand sides of differential equations into linear and nonlinear terms has to be done.

There are various suggestions in literature on how to deal with nonlinear terms in the context of POD model reduction, whose key idea to approximate also the nonliniarities by basis vectors constructed from snapshots. The most populare ones are the empirical interpolation method (Grepl et al. (2007)) and best-point interpolation method (Nguyen et al. (2008)). These methods have to be integrated into the existing environment for automatic model reduction.

REFERENCES

Ginkel, M., Kremling, A., Nutsch, T., Rehner, R., and Gilles, E.D. (2003). Modular modeling of cellular systems with ProMoT/diva. *Bioinformatics*, 19(9), 1169–1176. doi:10.1093/bioinformatics/btg128.

Grepl, M.A., Maday, Y., Nguyen, N.C., and Patera, A.T. (2007). Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(3), 575–605. doi:10.1051/m2an:2007031.

H. M. Park, D.H.C. (1996). The use of the karhunen-love decomposition for the modeling of distributed parameter systems. *Chemical Engineering Science*, 81–98. doi: 10.1016/0009-2509(95)00230-8.

John, V., Mitkova, T., Roland, M., Sundmacher, K., Tobiska, L., and Voigt, A. (2009). Simulations of population balance systems with one internal coordinate using finite element methods. *Chemical Engineering Science*, 64(4), 733–741. doi:10.1016/j.ces.2008.05.004.

Jürgen Koch, W.H. (2008). H-matrix methods for quadratic integral operators appearing in population balances. *Computers & Chemical Engineering*, 32, 1789–1809. doi:10.1016/j.compchemeng.2007.09.003.

Krasnyk, M. (2009). *Ph.D. Thesis*. Otto-von-Guericke-University Magdeburg.

Krasnyk, M. and Mangold, M. (2010). Reduction of a urea crystallizer model by proper orthogonal decomposition and best point interpolation. *Industrial & Engineering Chemistry Research*, 49, 9887–9898.

Kunisch, K. and Volkwein, S. (2003). Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2), 492–515.

Mangold, M. (2012). Model reduction of a batch drum granulator by proper orthogonal decomposition. In *8th IFAC Symposium on Advanced Control of Chemical Processes, 10 - 13 July 2012, Singapore*, 856–861.

Mangold, M., Khlopov, D., Palis, S., Feng, L., Benner, P., Binev, D., and Seidel-Morgenstern, A. (2014a). Non-

linear model reduction of a continuous fluidized bed crystallizer. *Journal of Computational and Applied Mathematics*.

Mangold, M., Khlopov, D., Danker, G., Palis, S., Svjatnyj, V., and Kienle, A. (2014b). Development and nonlinear analysis of dynamic plant models in ProMoT /diana. *Chemie Ingenieur Technik*, 86(7), 1107–1116. doi:10.1002/cite.201400003.

Mirschel, S., Steinmetz, K., Rempel, M., Ginkel, M., and Gilles, E.D. (2009). ProMoT: modular modeling for systems biology. *Bioinformatics*, 25(5), 687–689. doi: 10.1093/bioinformatics/btp029.

Nguyen, N.C., Patera, A.T., and Peraire, J. (2008). A best points interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering*, 73(4), 521–543. doi: 10.1002/nme.2086.

Öncül, A.A., Sundmacher, K., Seidel-Morgenstern, A., and Thvenin, D. (2006). Numerical and analytical investigation of barium sulphate crystallization. *Chemical Engineering Science*, 61(2), 652–664. doi: 10.1016/j.ces.2005.07.037.

Palis, S., Binev, D., Lorenz, H., Seidel-Morgenstern, A., and Kienle, A. (2013). Population balance modeling of crystallization in a fluidized bed. *BIWIC 2013 20th International Workshop on Industrial Crystallization*, 252–259.

Ramkrishna, D. (2000). *Population balances: theory and application to particulate systems in engineering.* Academic Press.

Sirovich, L. (1987). Turbulence and the dynamics of coherent structures part i: coherent structures. *Quarterly of Applied Mathematics*, 257–267.

Tung, H.H., Paul, E.L., Midler, M., and McCauley, J.A. (2009). *Crystallization of Organic Compounds: An Industrial Perspective*. John Wiley & Sons.

Wintermantel, K. (1999). Process and product engineeringachievements, present and future challenges. *Chemical Engineering Science*, 54, 1601–1620. doi: 10.1016/S0009-2509(98)00411-4.