Multiscale Model Parallelization and Run-to-Run Control of Batch Protein Crystallization

Joseph Kwon*[1] and Panagiotis D. Christofides[2]

[1]Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77845 USA
[2]Department of Chemical and Biomolecular Engineering and Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA

Abstract

In this work, we initially develop a parallelized multiscale, multidomain modeling scheme that directly reduces computation time requirements without compromising the accuracy of established chemical models for a batch protein crystallization process. Then, a double exponentially weighted moving average (dEWMA)-based model predictive controller (MPC) is applied to a multiscale model of a batch crystallization process used to produce hen-egg-while (HEW) lysozyme crystals to compensate for batch-to-batch drift in process parameters. The average crystal shape distribution of crystals produced from the closed-loop simulation of the batch crystallizer under the dEWMA-based MPC is much closer to a desired set-point value compared to that of MPC based on the nominal process model.

Keywords

Multiscale modeling, parallel computation, crystallization, run-to-run control, model predictive control.

Introduction

Multiscale process modeling has made possible to improve fundamental understanding and quantitative prediction of complex process behavior and product characteristics and has tremendous potential for significant contributions to the chemical, pharmaceutical and microelectronics industries Kevrekidis et al. (2004); Vlachos (2005); Christofides et al. (2008). Motivated by the growing high-performance computing power, an increasing interest for multiscale, multidomain modeling has been triggered.

More specifically, kinetic Monte Carlo (kMC) modeling has received a growing attention to carry out dynamic simulations of microscopic/mesoscopic process behavior. For many problems of practical interest, however, one needs to simulate systems with larger temporal and spatial scales than the ones that can be accomplished using a serial algorithm and available computing power. For these problems, motivated by the recent efforts to develop parallel computation frameworks for the simulation of multisacle process models Cheimarios et al. (2013); Mosby and Matous (2015), it would be desirable to develop efficient parallel kMC algorithms so that many processors can be used simultaneously in order to accomplish realistic computations over extended temporal and spatial scales. However, there has been surprisingly little work done on parallel algorithms for kMC simulation.

Motivated by this, we have attempted to directly deal with the problem of reducing computational requirements without compromising the accuracy of established chemical models via a parallelized kMC implementation. We show that choosing appropriate decomposition strategy is the key to reducing communication among processors and is ideally suitable for parallel implementation without compromising precision. Specifically, the message passing interface (MPI) settings that use the information passing between the cores are selected as the method of choice following a "manager-worker" scheme and are applied to batch configuration which is one of the most widely used reactor and crystallization config-

*To whom all correspondence should be addressed *kwonx075@tamu.edu*

urations in the specialty chemicals and pharmaceutical industries.

Additionally, from the practical standpoint of operating batch crystallization processes, unknown systematic trends or drifts in the process parameter values, for example, in initial pH level, operating conditions, and impurity concentrations in raw materials (e.g., Flores-Cerrillo and MacGregor (2004)) may be challenging because even a small change in the pH level may have a significant influence on the size and shape distribution of crystal products, and thereby, on the bioavailability of crystals produced from a batch crystallization process. The best known method for handling batch-to-batch drift is the double exponentially weighted moving average (dEWMA) formula, which can capture the changes in the rate of the process drift, and thus, forecast the process drift in the next batch run Su and Hsu (2004); Tseng and Hsu (2005). Within this context, a dEWMA-based model predictive controller is proposed in order to deal with the batch-to-batch dynamics of the process drift, utilizing both in-batch and post-batch measurements.

The manuscript is structured as follows: we initially discuss the multiscale model of our case study, a batch crystallization process used to produce tetragonal hen-egg-white (HEW) lysozyme crystals. Then, a general parallel computational framework suitable for multiscale models is discussed. Additionally, we develop a dEWMA-based model predictive controller (MPC) in order to achieve the production of crystals with a desired shape distribution by handling batch-to-batch parametric drifts in the batch system. A series of results demonstrating the computation efficiency of the parallel computation framework and the control performance of a dEWMA-based MPC will be presented.

1 Multiscale batch crystallization process model and parallelization

In a crystallization process, there is large disparity of time and length scales of phenomena occurring in continuous phase and crystal surface. As a result, the assumption of continuum may not be valid on the crystal surface, and furthermore, it is computationally impossible to model the whole batch system from a molecular point of view.

Therefore, we present an integrated multiscale modeling and parallel computation framework for crystallization processes that elucidates the relationship between molecular-level processes like crystal nucleation, growth, and aggregation and macroscopically-observable process behavior and allows computing optimal design and operation conditions. The multiscale framework encompasses: a) kMC modeling for simulating crystal growth and aggregation and predicting the evolution of crystal shape distribution, and b) integrated multiscale computation simultaneously linking molecular-level models (e.g., kMC simulation) and continuous-phase macroscopic equations (e.g., mass and energy balance equations), covering entire batch crystallization systems.

1.1 Microscopic model

The solid-on-solid model is employed in this work to model the growth of lysozyme crystals. Each event of our kMC simulation is chosen randomly based on the rates of the three surface microscopic phenomena described in Ke et al. (1998); Durbin and Feher (1991).

The adsorption rate is defined as

$$r_a = K_0^+ \exp\left(\frac{\Delta\mu}{k_B T}\right), \tag{1}$$

where K_0^+ is the attachment coefficient, k_B is the Boltzmann constant, T is the temperature in Kelvin, and $\Delta\mu = k_B T \ln(C/S)$ is the crystal growth driving force, where C is the protein solute concentration and S is its solubility. The desorption rate is given by

$$r_d(i) = K_0^+ \exp\left(\frac{\phi}{k_B T} - i\frac{E_{pb}}{k_B T}\right), \tag{2}$$

where i is the number of bonds, ϕ is the binding energy per molecule of a fully occupied lattice, and E_{pb} is the average binding energy per bond. Similar to the desorption rate, the migration rate is shown below

$$r_m(i) = K_0^+ \exp\left(\frac{\phi}{k_B T} - i\frac{E_{pb}}{k_B T} + \frac{E_{pb}}{2k_B T}\right). \tag{3}$$

In this work, we have determined a set of model parameters as follows: $E_{pb}/k_B = 1077.26$ K and $\phi/k_B = 227.10$ K for the (110) face, and $E_{pb}/k_B = 800.66$ K and $\phi/k_B = 241.65$ K for the (101) face, and $K_0^+ = 0.211$ s^{-1}.

1.2 Macroscopic model

The following mass and energy balance equations are employed to compute the dynamic evolution of the protein solute concentration and temperature in the batch crystallization process with time:

$$\frac{dC}{dt} = -\frac{\rho_c}{V_{batch}}\frac{dV_{crystal}}{dt} \tag{4}$$

$$\frac{dT}{dt} = -\frac{\rho_c \Delta H_c}{\rho C_p V_{batch}} \frac{dV_{crystal}}{dt} - \frac{U_c A_c}{\rho C_p V_{batch}} (T - T_j) \quad (5)$$

where $V_{crystal}$ is the total volume of crystals growing in the crystallizer, T_j is the jacket temperature (i.e., manipulated input), $A_c = 0.25$ m^2 is the contact area of the crystallizer wall and jacket, $\Delta H_c = -4.5$ kJ/kg is the enthalpy of crystallization, $\rho_c = 1400$ mg/cm^3 is the crystal density, $\rho(t) = 1000 + C(t)$ mg/cm^3 is the density of the continuous phase, $C_p = 4.13$ kJ/K $\cdot$ kg is the specific heat capacity, $U_c = 1800$ kJ/m$^2\cdot$h $\cdot$ K is the overall heat transfer coefficient of crystallizer wall, and $V_{batch} = 1$ L is the volume of the crystallizer. The initial conditions are $C(0) = 48$ mg/mL and $T(0) = 15$ °C

Additionally, aggregation of protein crystals is considered. An expression (Eq. 6 below) can be used to calculate the number of aggregation event taking place during the time Δ by taking into account the aggregation kernel $\beta(V_i, V_j)$, batch reactor volume V_{batch}, collision efficiency $\alpha(V_i, V_j)$, and concentrations of particles of volume V_i and V_j as follows Kwon et al. (2013):

$$N_{ij} = \alpha(V_i, V_j)\, \beta(V_i, V_j)\, m_i m_j V \Delta \quad (6)$$

where m_i is the number concentration (i.e., the number of particles of class i per unit volume) and $1 \leq i, j \leq C_{total}$ where C_{total} indicates the number of classes). The rate of formation of aggregates of volume V_k from the collision of particles of volume V_i and V_j is calculated as $\frac{1}{2}\sum_{V_i+V_j=V_k} N_{ij}$.

1.3 Parallel computation of multiscale model

The simulation of the crystal growth process of crystals formed via nucleation is executed in parallel by using MPI through which we are able to divide the crystals to multiple cores by achieving the distribution of the computational cost and memory requirements. More detailed discussion on the step-by-step parallelization will be discussed in the following section.

1.3.1 Decomposition

We can decompose the nucleation and crystal growth processes in batch crystallization system into the collection of tasks where each task is the crystal growth of a nucleated crystal.

1.3.2 Assignment

As soon as a crystal is nucleated, it will be assigned to one of the available cores, and it will grow to a larger crystal via the kMC simulation. Since crystals are continuously nucleated, the total size of tasks assigned to each core grows with time. More specifically, nucleated crystals are assigned following the order described in Table 1 (i.e., crystal number modules is equal to the number of cores available). We note that this is a number-based allocation assuming that all cores have identical processor speed and memory.

core	crystal number	crystal number
worker 1	1	$n+1$
worker 2	2	$n+2$
$\vdots$	$\vdots$	$\vdots$
worker n	n	$2n$

Table 1. The order that nucleated crystals are assigned to each core. Suppose there are $2n$ crystals.

The collision probability of whether an aggregation event takes place between two crystals with volumes V_i and V_j during a time period can be calculated via Eq. 6. Next, the aggregation event is executed when a random number generated from $[0, 1)$ is less than the collision probability. If an aggregation event occurs, we pick a crystal from each class i and j, respectively, and the smaller crystal between those two crystals will be removed from the kMC simulation while its volume will be added to the larger one, making it an aggregate where its volume is equal to the total volume of the two crystals before the aggregation event. This process applies to all possible pairwise combinations of crystal volumes over the course of the entire batch crystallization simulation.

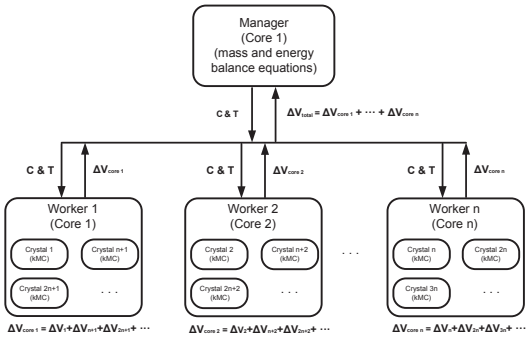

Figure 1. Manager-worker parallel computation scheme for multiscale model of batch crystallization process.

1.3.3 Orchestration

Fig. 1 illustrates schematically how the information passing between the cores is managed with the MPI settings in order to link the macroscopic model (i.e., mass and energy balance equations for the continuous phase) to microscopic model (i.e., kMC model). The coupled simulation follows the "manager-worker" MPI computational scheme: there is a core (i.e., manager) that is responsible for collecting the change in the total volume of crystals assigned to each core (i.e., worker) at each time step, which corresponds to the amount of solute transported from the continuous phase to the crystal surface. Then, the manager core computes the change in the total volume of the crystals in the crystallizer at each time step and computes the protein solute concentration C and the temperature T for the continuous phase in the crystallizer using mass and energy balance equations. The updated C and T will be sent back to the worker cores, and those values at each core will remain identical until they are updated again after a time step. Then, the crystals assigned to each core will grow with an updated condition via kMC simulations.

1.3.4 Pseudo-code

Algorithm 1 Parallel computation of the multiscale batch crystallization process model

for $i = 1 \rightarrow n$ **do**
  if $i == 1$ **then** ▷ Manager core
    1. assign nucleated crystals over Δt to each core according to Table 1
    2. compute $\Delta V_{\text{total}}(t) = \sum_{i=1}^{n} \Delta V_i(t)$
    3. update $C(t)$ and $T(t)$ through Eqs. 4–5
    4. compute the total number of collisions between crystals via Eq. 6
  else ▷ Worker core
    1. have crystals assigned to each core grow via Eqs. 1–3
    2. compute $\Delta V_i(t)$ and send it to the manager core
  end if
end for

Please note that $\Delta V_{\text{total}}(t)$ is the change in the total volume of crystals in the crystallizer from $t - \Delta t$ to t seconds, and $\Delta V_i(t)$ is the change in the total volume of crystals particularly assigned to the core i from $t - \Delta t$ to t seconds. The reader may refer to Kwon et al. (2014a,b) for the use of the parallelization scheme to different applications including the plug flow crystallizer and the continuous stirred tank crystallizer with a fines trap and a product classification unit.

1.4 Results

In this work, the Hoffman2 cluster at UCLA which consists of 1200 nodes with a total of 13,340 cores and over 50TB of memory, is used along with MPI settings for the simulations of all the batch crystallization runs.

It is presented in Table 2 that the simulation times required to complete a batch simulation decrease as the number of cores is increased. Also, the corresponding speedup presents that as the number of cores is doubled, the speedup achieved in comparison to the theoretical maximum speedup (i.e., the theoretical maximum speedup should be n times when n processors are used) decreases, because of overhead costs generated by communication taking place between multiple cores. Overall, it is clear that the batch crystallization process is greatly benefiting from the use of MPI for the kMC simulations.

n_{cores}	time (h)	speedup	ideal speedup
1	34.97	1.00	1
2	17.63	1.98	2
4	8.98	3.89	4
8	4.71	7.44	8
16	2.47	14.18	16
32	1.38	25.28	32
64	0.92	38.15	64

Table 2. The time required to run a batch simulation and the speedup achieved by using different numbers of cores. Please note that n_{cores} is the number of cores and the speedup is defined as $\frac{t_1}{t_n}$, where t_1 is the time the process takes on 1 core and t_n is the time the process takes on n cores.

2 R2R-based MPC

2.1 MPC

We assume that a model predictive controller (MPC) is available for in-batch control. Specifically, the dominant dynamic behavior of the evolution of crystal shape distribution in the batch crystallization process is modeled through a moments model and the manipulated input is the jacket temperature. The detailed MPC formulation can be found in Kwon et al. (2015).

2.2 dEWMA-based model predictive control

Because of its ability to handle batch-to-batch parametric drifts Bulter and Stefani (1994); Simith et al. (1998), a dEWMA scheme is integrated with the MPC to updated the MPC model parameters after each batch and the closed-loop performance is evaluated along with that of the MPC with the nominal process model. In the dEWMA scheme, the predicted value of average crystal shape for the k^{th} batch run can be formulated as follows:

$$\langle \widetilde{\alpha(t_f)} \rangle_k = \langle \widehat{\alpha(t_f)} \rangle_k + \hat{e}_k + \Delta \hat{e}_k \tag{7}$$

where $\langle \widetilde{\alpha(t_f)} \rangle_k$ is the predicted average crystal shape at the end of the k^{th} batch, $\langle \widehat{\alpha(t_f)} \rangle_k$ is the predicted average crystal shape using only the nominal process model that consists of Eqs. (1)-(3) and (7)-(14) in Kwon et al. (2015), $\hat{e}_k$ is the estimated model prediction error, and $\Delta \hat{e}_k$ is used to compensate for the error in the parameter estimation caused by the change in the rate of the process drift. For a dEWMA-based MPC, an offset from the nominal model is approximated by $\hat{e}_k + \Delta \hat{e}_k$, which is described in detail as follows:

1. At the end of the k^{th} batch run, average crystal size and shape of crystals are measured (i.e., post-batch measurements).

2. Then, the average crystal shape measured from Step 1, $\langle \alpha(t_f) \rangle_k$, is used to compute the estimated model prediction error, $\hat{e}_k$, and the estimated change in the rate of the process drift, $\Delta \hat{e}_k$, through the following equation:

$$\hat{e}_{k+1} = \omega_1 \left[\langle \alpha(t_f) \rangle_k - \langle \widehat{\alpha(t_f)} \rangle_k \right] + (1 - \omega_1) \hat{e}_k \tag{8}$$

$$\begin{aligned} \Delta \hat{e}_{k+1} =& \omega_2 \left[\langle \alpha(t_f) \rangle_k - \langle \widehat{\alpha(t_f)} \rangle_k - \hat{e}_k \right] \\ &+ (1 - \omega_2) \Delta \hat{e}_k \end{aligned} \tag{9}$$

   where $0 < \omega_1 \leq 1$ and $0 < \omega_2 \leq 1$ are the learning factors.

3. Then, the predicted average crystal shape for the $k+1$ batch run, $\langle \widetilde{\alpha(t_f)} \rangle_{k+1}$, that accounts for the change in the rate of the process drift is obtained by the following equation:

$$\langle \widetilde{\alpha(t_f)} \rangle_{k+1} = \langle \widehat{\alpha(t_f)} \rangle_{k+1} + \hat{e}_{k+1} + \Delta \hat{e}_{k+1} \tag{10}$$

   and is used in the model employed in the MPC to compute a set of optimal jacket temperatures T_j which will drive the temperature T in the crystallizer to a desired value.

4. Increase k by 1 and repeat Step 1 to Step 4.

We note that the first equation, Eq. (8), is used to estimate the offset in the average crystal shape (i.e., output) and the second equation, Eq. (9), is used to capture an additional offset in the average crystal shape due to the change in the rate of the process drift.

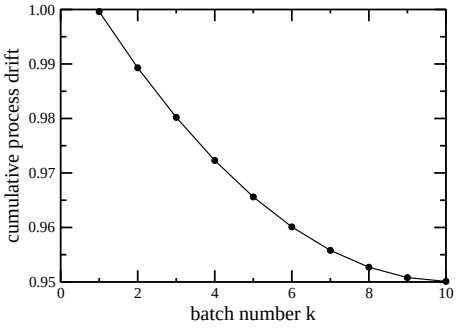

Figure 2. The evolution of the cumulative process drift with an exponentially decaying rate from batch-to-batch.

2.3 Application of dEWMA-based MPC to Batch Crystallization

One of the reasons that the control of the size and shape distributions of crystals produced from a batch process may be difficult is minor contaminations in the feedstock container which 1) may lead to a significant drift of key process parameters from batch-to-batch and 2) cannot be identified immediately, and thus, their undesired effect on the product quality continues to the next batch runs. To tackle this problem, the proposed dEWMA-based MPC where a dEWMA scheme is used to simultaneously approximate the batch-to-batch dynamics of the drift and adjust the jacket temperature in order to suppress the effect of the process drift in the next batch.

The controller performance of the dEWMA-based MPC is evaluated in response to an exponentially decaying process drift whose rate decays exponentially from 1 (i.e., nominal system) to 0.95 over 10 batch runs (see, e.g., Fig. 2). While the closed-loop performance of the MPC with the nominal process model becomes progressively worse (Fig. 3) due to the increasing mismatch between the process model and the actual batch crystallization process, it is shown in Fig. 3 that the dEWMA-based MPC is able to produce crystals with a desired shape distribution.

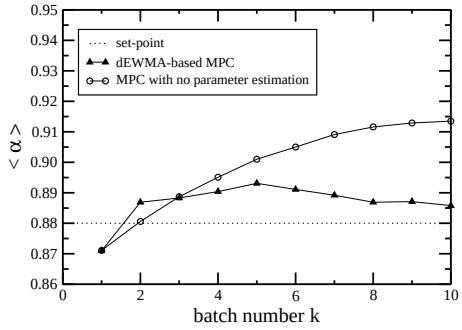

Figure 3. The evolution of the average crystal shape at $t = 20000$ seconds obtained from the kMC simulations from batch-to-batch under the MPC with the nominal process model and the dEWMA-based MPC with (w1,w2)=(0.5,0.5), with the desired set-point $\langle \alpha_{\text{set}} \rangle = 0.88$.

3 Conclusions

In this work, we proposed a parallelized multiscale, multidomain modeling scheme to directly reduce the computation time and memory requirements without compromising the accuracy of simulation results for a batch protein crystallization process. The parallelized multiscale modeling strategy that consists of the three steps of decomposition, assignment, and orchestration was applied to a batch crystallization process multiscale model. Then, a dEWMA-based MPC is designed and applied to a multiscale model of a batch crystallization process used to produce HEW lysozyme crystals. Lastly, for comparison purposes, the performance of the dEWMA-based MPC was favorably compared with that of the MPC based on the nominal process model.

References

Bulter, S. W. and Stefani, J. (1994). Supervisory run-to-run control of polysilicon gate etch using in situ ellipsometry. *IEEE Trans. Semiconductor Manufacturing*, 7:193–201.

Cheimarios, N., Kokkoris, G., and Boudouvis, A. (2013). An efficient parallel iteration method for multiscale analysis of chemical vapor deposition process. *Appl. Numerical Math.*, 67:78–88.

Christofides, P. D., Armaou, A., Lou, Y., and Varshney, A. (2008). *Control and Optimization of Multiscale Process Systems*. Birkhäuser, Boston.

Durbin, S. D. and Feher, G. (1991). Simulation of lysozyme crystal growth by the Monte Carlo method. *Journal of Crystal Growth*, 110:41–51.

Flores-Cerrillo, J. and MacGregor, J. (2004). Multivariate monitoring of batch processes using batch-to-batch information. *AIChE J.*, 50:1219–1228.

Ke, S. C., DeLucas, L. J., and Harrison, J. G. (1998). Computer simulation of protein crystal growth using aggregates as the growth unit. *J. Phys. D: Appl. Phys.*, 31:1064–1070.

Kevrekidis, I. G., Gear, C. W., and Hummer, G. (2004). Equation-free: the computer-aided analysis of complex multiscale systems. *AIChE J.*, 50:1346–1355.

Kwon, J. S., Nayhouse, M., Christofides, P. D., and Orkoulas, G. (2013). Modeling and control of shape distribution of protein crystal aggregates. *Chem. Eng. Sci.*, 104:484–497.

Kwon, J. S., Nayhouse, M., Christofides, P. D., and Orkoulas, G. (2014a). Modeling and control of crystal shape in continuous protein crystallization. *Chem. Eng. Sci.*, 107:47–57.

Kwon, J. S., Nayhouse, M., Orkoulas, G., and Christofides, P. D. (2014b). Crystal shape and size control using a plug flow crystallization configuration. *Chem. Eng. Sci.*, 119:30–39.

Kwon, J. S., Nayhouse, M., Orkoulas, G., Ni, D., and Christofides, P. D. (2015). A method for handling batch-to-batch parametric drift using moving horizon estimation: application to run-to-run MPC of batch crystallization. *Chem. Eng. Sci.*, 127:210–219.

Mosby, M. and Matous, K. (2015). Hierarchically parallel coupled finite strain multiscale solver for modeling heterogeneous layers. *Int. J. Numerical Methods in Eng.*, 102:748–765.

Simith, T., Boning, D., Stefani, J., and Bulter, S. (1998). Run by run advanced process control of metal sputter deposition. *IEEE Trans. Semiconductor Manufacturing*, 11:276–284.

Su, C. and Hsu, C. (2004). A time-varying weights tuning method of the double EWMA controller. *Omega Int. J. of Mgmt. Sci.*, 32:473–480.

Tseng, S. and Hsu, N. (2005). Sample-size determination for achieving asymptotic stability of a double EWMA control scheme. *IEEE Trans. Semiconductor Manufacturing*, 18:104–111.

Vlachos, D. G. (2005). A review of multiscale analysis: examples from systems biology, materials engineering, and other fluid-surface interacting systems. *Advances in Chem. Eng.*, 30:1–61.